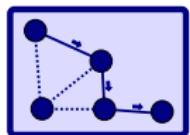


# Une approche d'optimisation discrète pour la classification associative

**SOD322 - RO et données massives**

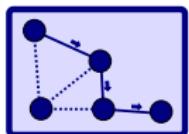
Zacharie ALES  
(zacharie.ales@ensta-paris.fr)





## Introduction à la RO

E. Soutil



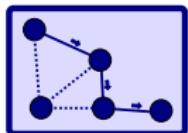
### Introduction à la RO

E. Soutil



### Introduction au ML

P. Bianchi



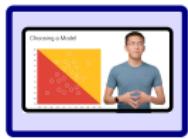
Introduction à la RO

E. Soutil

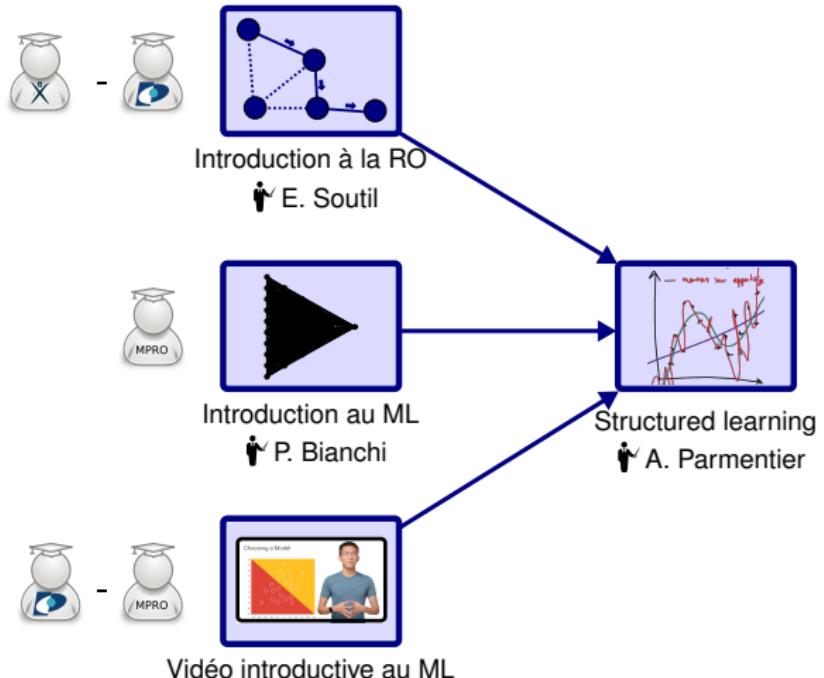


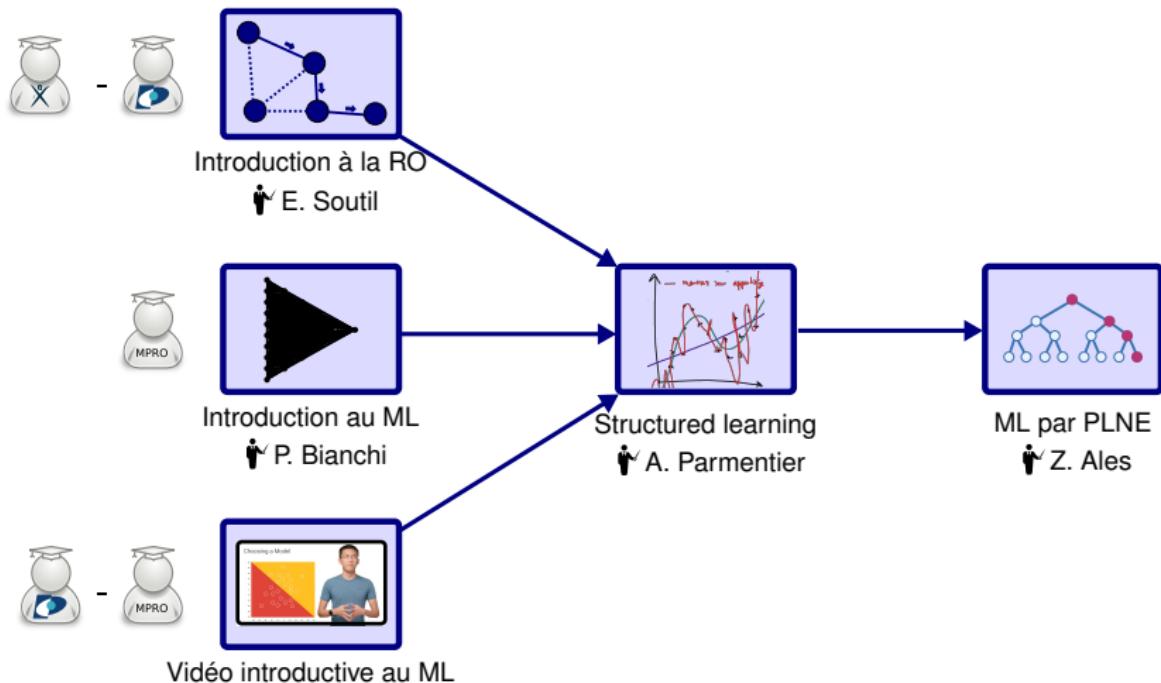
Introduction au ML

P. Bianchi



Vidéo introductive au ML





## 1 Introduction à la classification supervisée

## 2 Construction d'arbres de décision optimaux

## 3 Projet

- Sujet
- Julia

# Sommaire

- 1 Introduction à la classification supervisée
- 2 Construction d'arbres de décision optimaux
- 3 Projet

## Définition - Classification supervisée

**Données :** Ensemble de couples  $\{(x^i, y^i)\}_{i \in \mathcal{I}}$

Caractéristiques de la donnée  $i \in \mathbb{R}^{|\mathcal{J}|}$   $\uparrow$   $\uparrow$  Classe de la donnée  $i \in \mathcal{K}$

**But :** Déterminer  $f(x) = y$  qui prédit au mieux la classe de données à partir de leurs caractéristiques  
 $\downarrow$  Classifieur

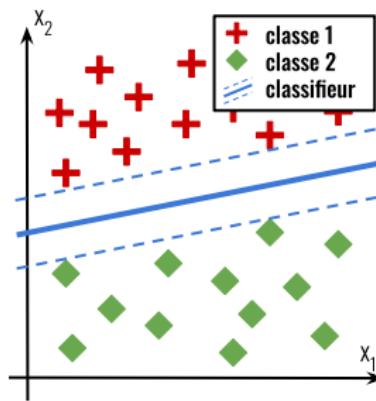
Caractéristiques d'une donnée

(ex :  10% d'alcool couleur 240nm)



Classifieur

Classe associée (ex : bière)



## Définition - Classification supervisée

**Données :** Ensemble de couples  $\{(x^i, y^i)\}_{i \in \mathcal{I}}$

Caractéristiques de la donnée  $i \in \mathbb{R}^{|\mathcal{J}|}$   $\uparrow$  Classe de la donnée  $i \in \mathcal{K}$

**But :** Déterminer  $f(x) = y$  qui prédit au mieux la classe de données à partir de leurs caractéristiques  
 $\uparrow$   
 Classifieur

### Exemple - Classe

- Vin, bière
- Chat, chien, oiseaux, ...
- Chiffres manuscrits
- ...

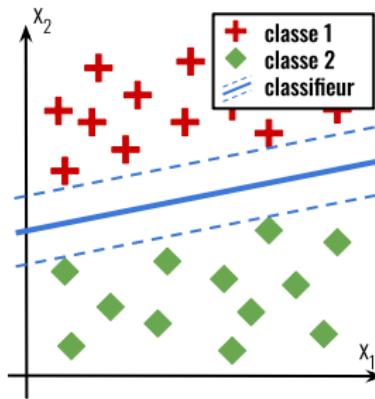
Caractéristiques d'une donnée

(ex :  10% d'alcool couleur 240nm)



Classifieur

Classe associée (ex : bière)



## Définition - Classification supervisée

**Données :** Ensemble de couples  $\{(x^i, y^i)\}_{i \in \mathcal{I}}$

Caractéristiques de la donnée  $i \in \mathbb{R}^{|\mathcal{J}|}$   $\uparrow$  Classe de la donnée  $i \in \mathcal{K}$

**But :** Déterminer  $f(x) = y$  qui prédit au mieux la classe de données à partir de leurs caractéristiques  
 $\uparrow$   
 Classifieur

### Exemple - Classe

- Vin, bière
- Chat, chien, oiseaux, ...
- Chiffres manuscrits
- ...

Caractéristiques d'une donnée

(ex :  10% d'alcool  
couleur 240nm)

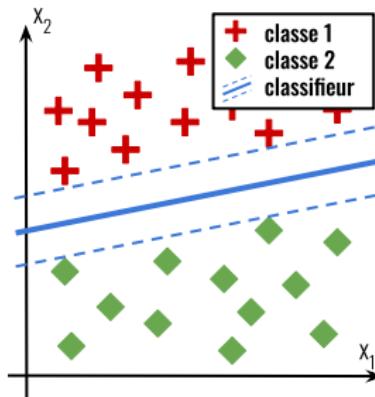


Classifieur

Classe associée (ex : bière)

### Exemple - Caractéristiques

- Taux d'alcool, couleur
- Ratio hauteur/longueur, forme, ...



## Partage des données

- ① **Apprentissage** ("train") : données utilisées pour définir le classifieur
- ② **Test** : données utilisées pour évaluer les performances du classifieur

Caractéristiques						Classe
1	0	0	1	...	0	
0	0	1	1	...	1	
:	:	:	:	...	:	
:	:	:	:	...	:	:
0	1	0	1	...	0	2

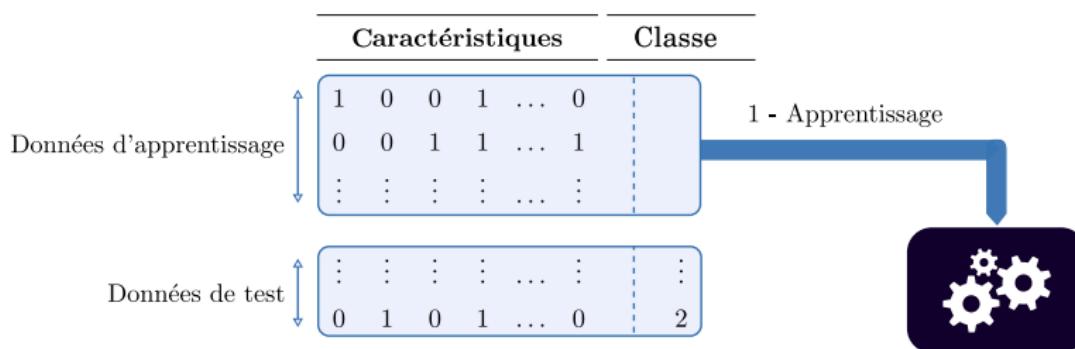
## Partage des données

- ① **Apprentissage** ("train") : données utilisées pour définir le classifieur
- ② **Test** : données utilisées pour évaluer les performances du classifieur

	Caractéristiques						Classe
Données d'apprentissage	1	0	0	1	...	0	
	0	0	1	1	...	1	
	:	:	:	:	...	:	
Données de test	:	:	:	:	...	:	:
	0	1	0	1	...	0	2

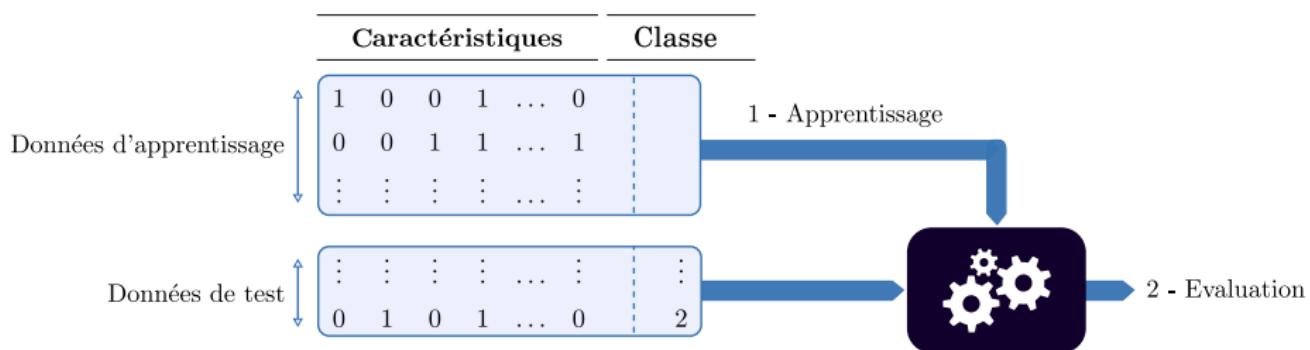
## Partage des données

- ① **Apprentissage** ("train") : données utilisées pour définir le classifieur
- ② **Test** : données utilisées pour évaluer les performances du classifieur

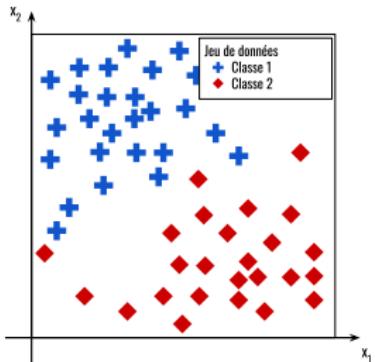


## Partage des données

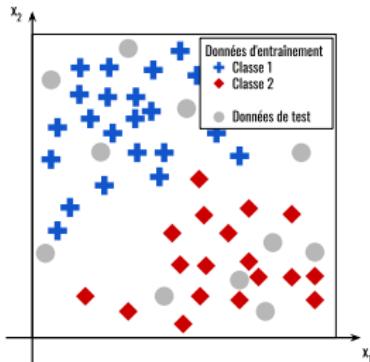
- ➊ **Apprentissage** ("train") : données utilisées pour définir le classifieur
  - ➋ **Test** : données utilisées pour évaluer les performances du classifieur



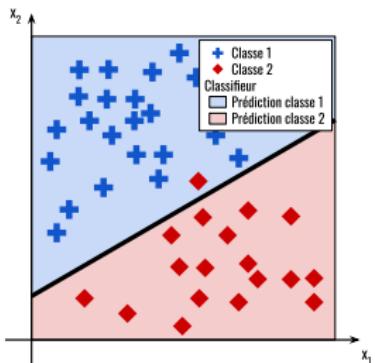
# Évaluation des classificateurs



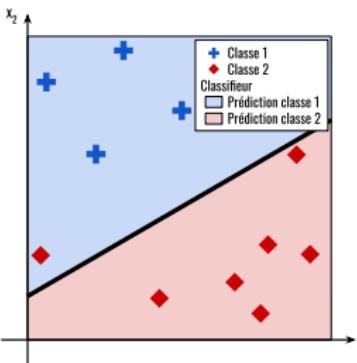
Données initiales



Partition des données en 2 ensembles

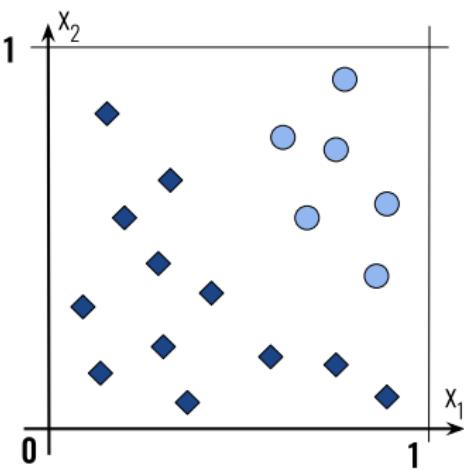


Apprentissage du classifieur sur les données d'entraînement

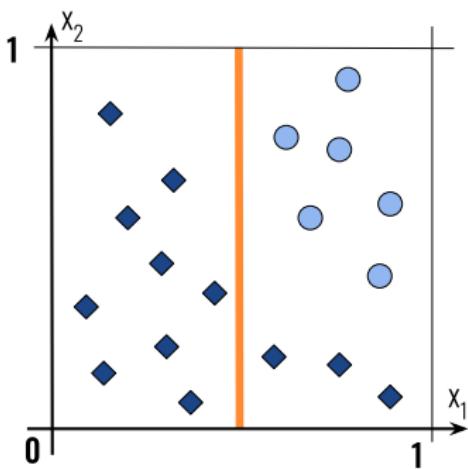
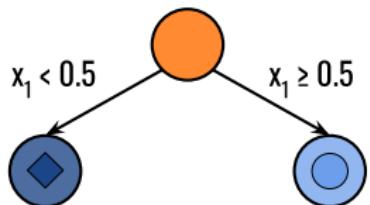


Prédictions du classifieur sur les données de test

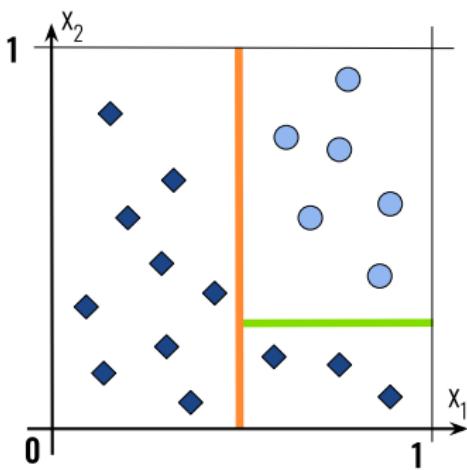
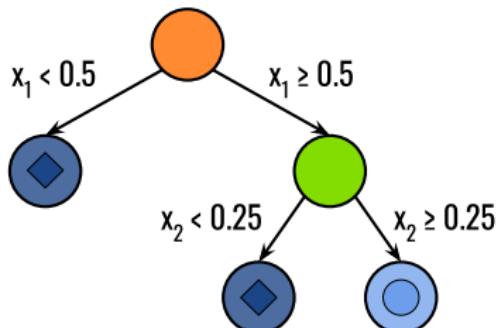
# Exemple de classifieur : Arbres de décision



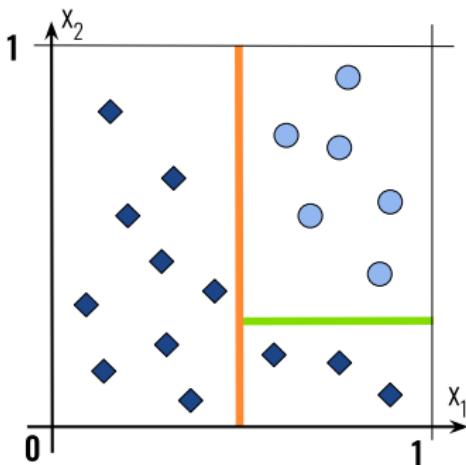
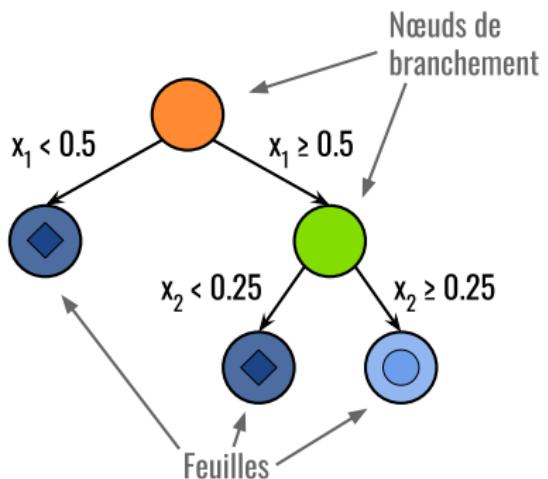
# Exemple de classifieur : Arbres de décision



# Exemple de classifieur : Arbres de décision



# Exemple de classifieur : Arbres de décision



# Exemple de classifieurs

## Classifieur - Forêts d'arbres décisionnels

- Apprentissage de multiples arbres aléatoires sur des sous-ensembles de données légèrement différents
- Prédiction : vote majoritaire des arbres

Aussi appelées forêts aléatoires ("random forest " en anglais)

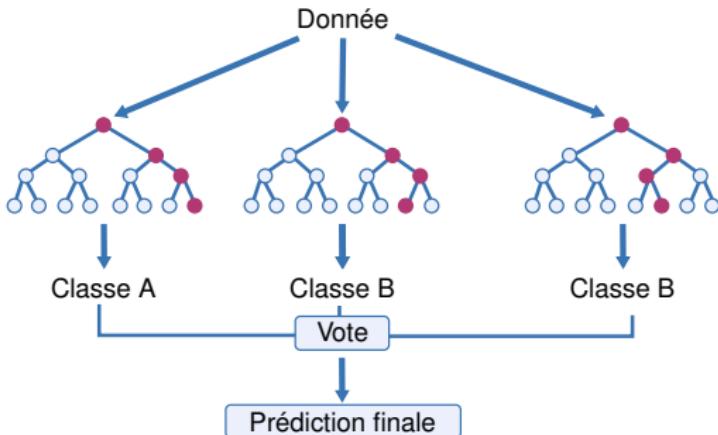
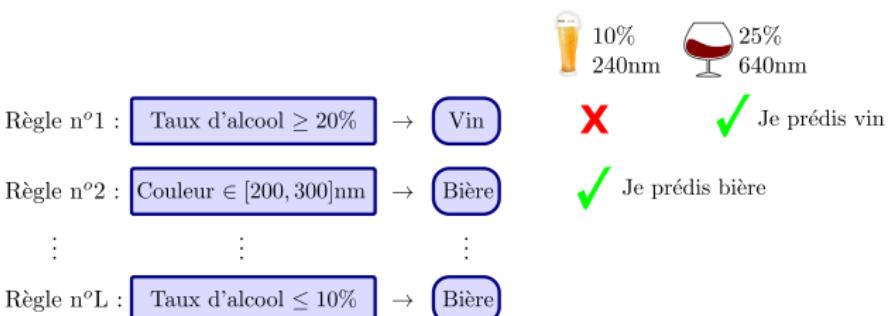


Image inspirée de [kdnuggets.com](http://kdnuggets.com)

# Exemple de classificateurs : Liste de décision



## Méthode - Ordered Rules for Classification (ORC)

Un classifieur associatif de type liste de décision obtenu par résolution de PLNE



Allison Chang, Dimitris Bertsimas, and Cynthia Rudin.

An integer optimization approach to associative classification.

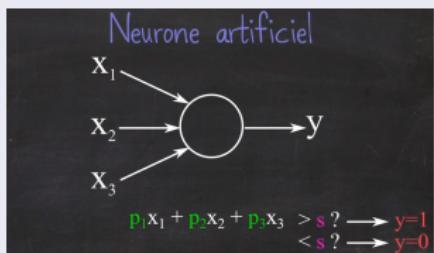
In Advances in neural information processing systems, pages 269–277, 2012.

## Désavantage

- Résolution exacte lente
- Données sous forme de vecteurs binaires  
*One-hot encoding*

# Exemple de classificateurs : Réseaux de neurones

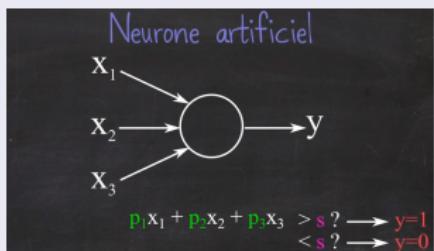
## Classifieur - Réseaux de neurones



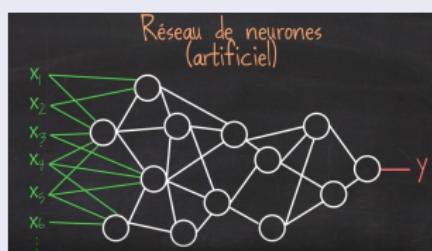
Neurone artificiel

# Exemple de classificateurs : Réseaux de neurones

## Classifieur - Réseaux de neurones



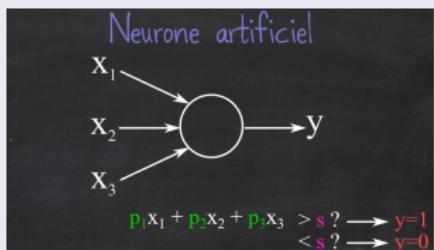
Neurone artificiel



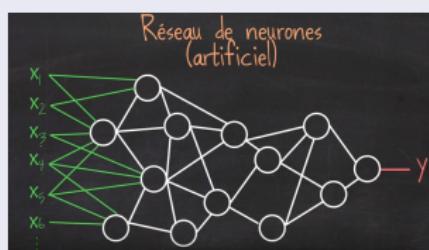
Réseau de neurones

# Exemple de classifieurs : Réseaux de neurones

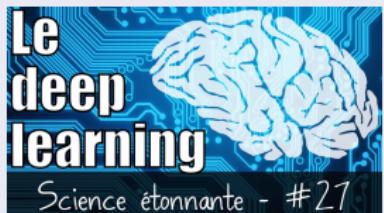
## Classifieur - Réseaux de neurones



Neurone artificiel



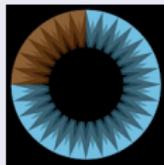
Images issues de



- Lien youtube
- Chaîne : ScienceEtonnante
- Intervenant : David Louapre

# Exemple de classifieurs

Plus de détails sur le fonctionnement des réseaux de neurones



- Série de 4 vidéos youtube
- Chaîne : 3Blue1Brown

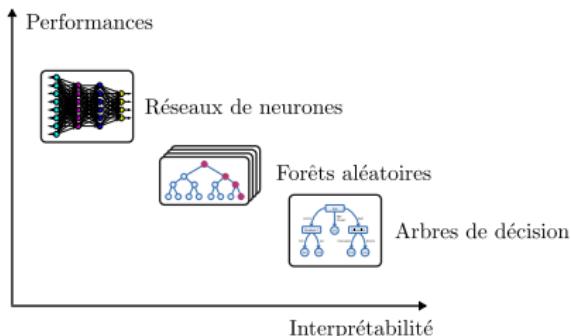
# Interprétabilité des classifieurs

## Définition - Interprétabilité

Quantifie dans quelle mesure un humain serait capable de prédire de manière cohérente le résultat d'un modèle [Miller 19]

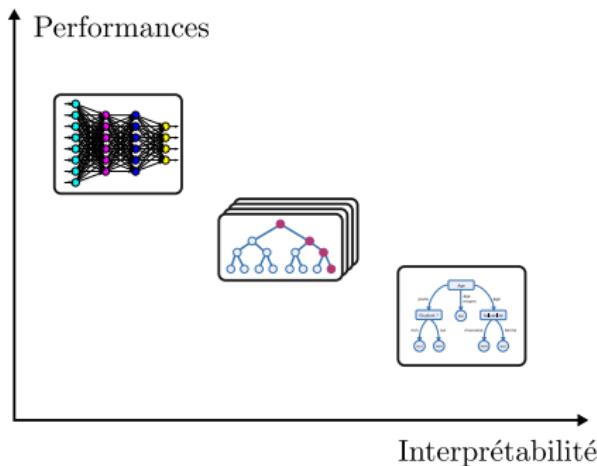
## Pourquoi s'intéresser à l'interprétabilité ?

- Important pour la prise de décision sensible  
[Peines de prison, trajectoire de voiture, identification de tumeurs, ...](#)
- Peut augmenter la confiance de l'utilisateur dans les résultats  
[Ex : médecins, patients, décideurs, ...](#)
- Parfois imposé par la loi  
[RGPD, Right to explanation, credit scoring, ...](#)
- Permet d'extraire des connaissances



## Intérêt de la RO

- Permet l'obtention de classificateurs par des méthodes exactes
- Améliore le pouvoir prédictif de modèles de classificateurs interprétables

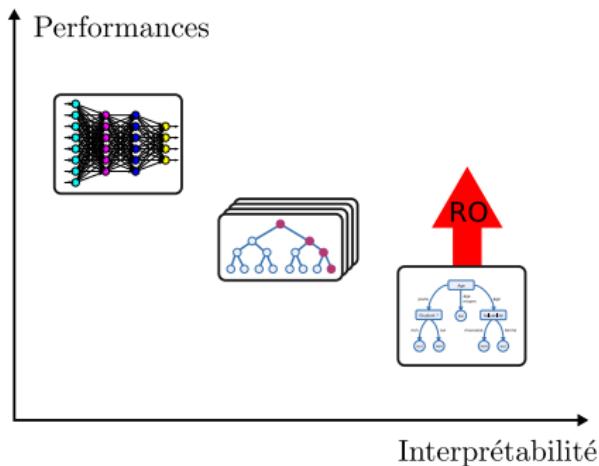


## Inconvénients

- Méthodes exactes souvent lentes
- Nécessite des données d'apprentissages de faible taille  
Ou des pré-traitements (voir projet)
- Peut entraîner du surapprentissage

## Intérêt de la RO

- Permet l'obtention de classificateurs par des méthodes exactes
- Améliore le pouvoir prédictif de modèles de classificateurs interprétables



## Inconvénients

- Méthodes exactes souvent lentes
- Nécessite des données d'apprentissages de faible taille  
Ou des pré-traitements (voir projet)
- Peut entraîner du surapprentissage

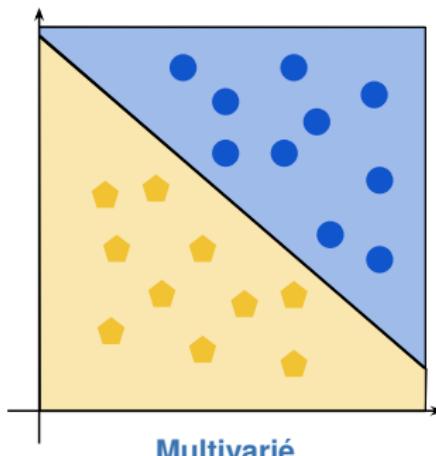
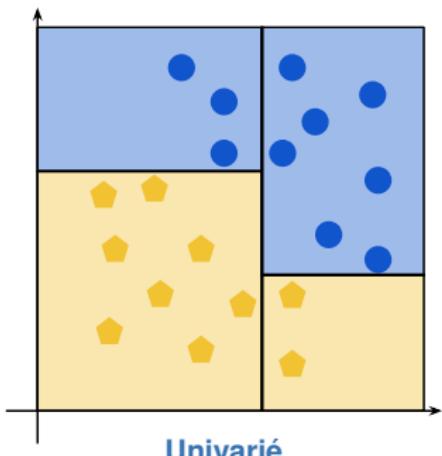
# Sommaire

- 1 Introduction à la classification supervisée
- 2 Construction d'arbres de décision optimaux
- 3 Projet

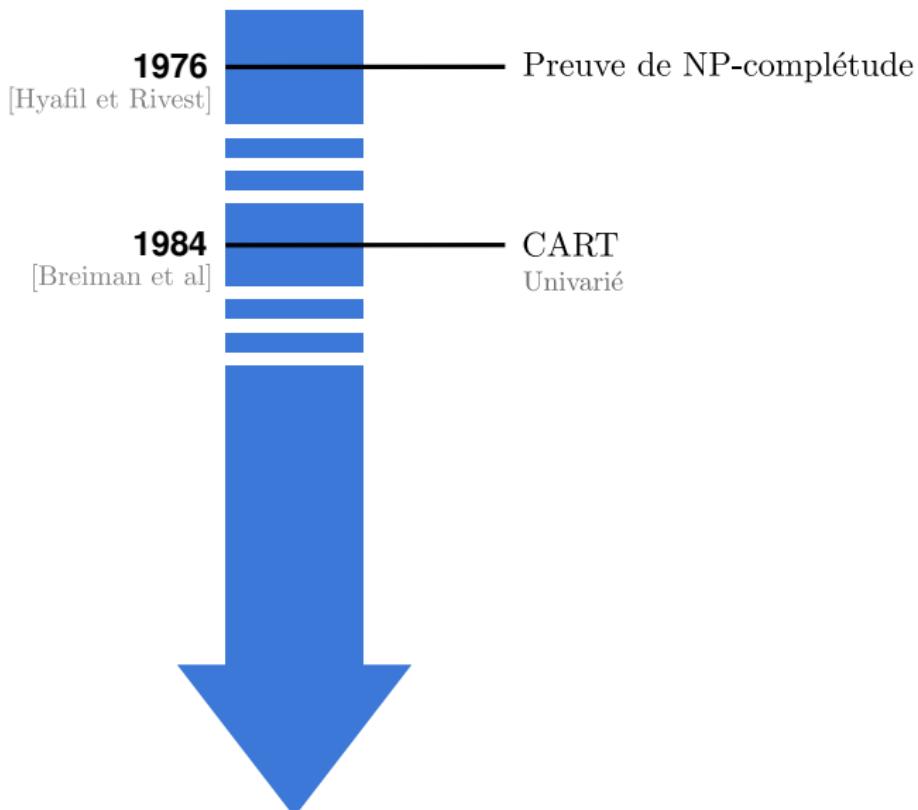
## Deux types de séparations

$$\sum_{j \in \mathcal{J}} a_j x_{i,j} < b \quad \text{Valeur de l'attribut } j \text{ de la donnée } i$$

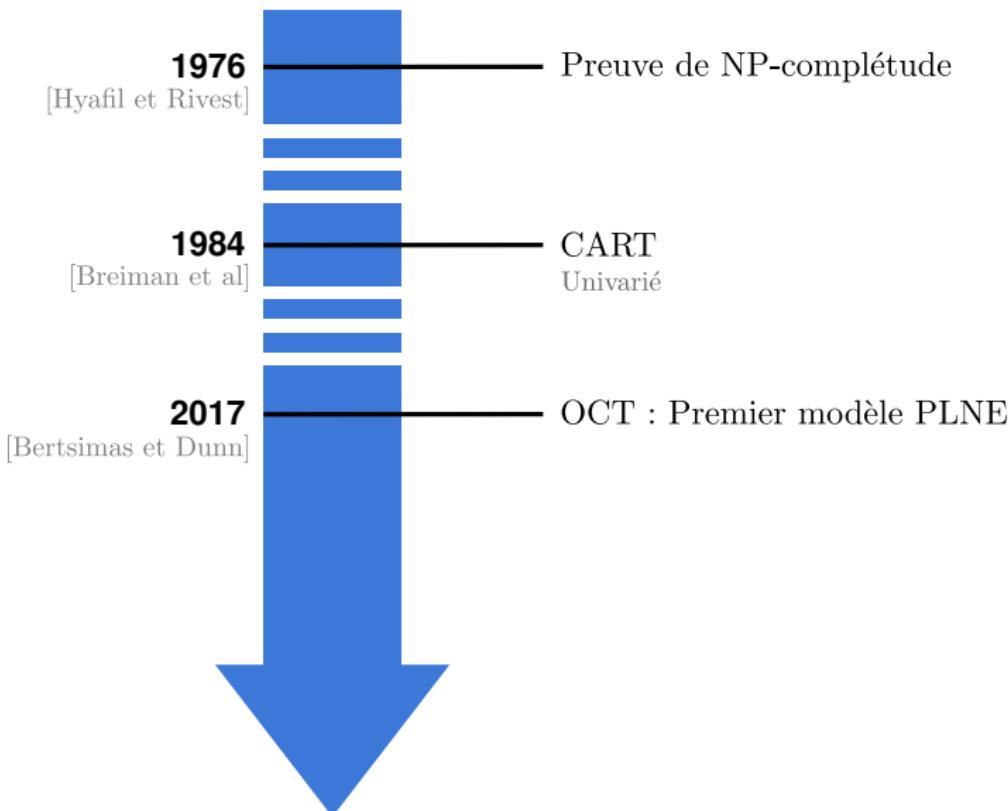
- **Univarié** : 1 seul attribut par séparation  
 $a$  : vecteur unitaire
- **Multivarié** : pas de limite du nombre d'attributs par séparation  
 $a$  non contraint



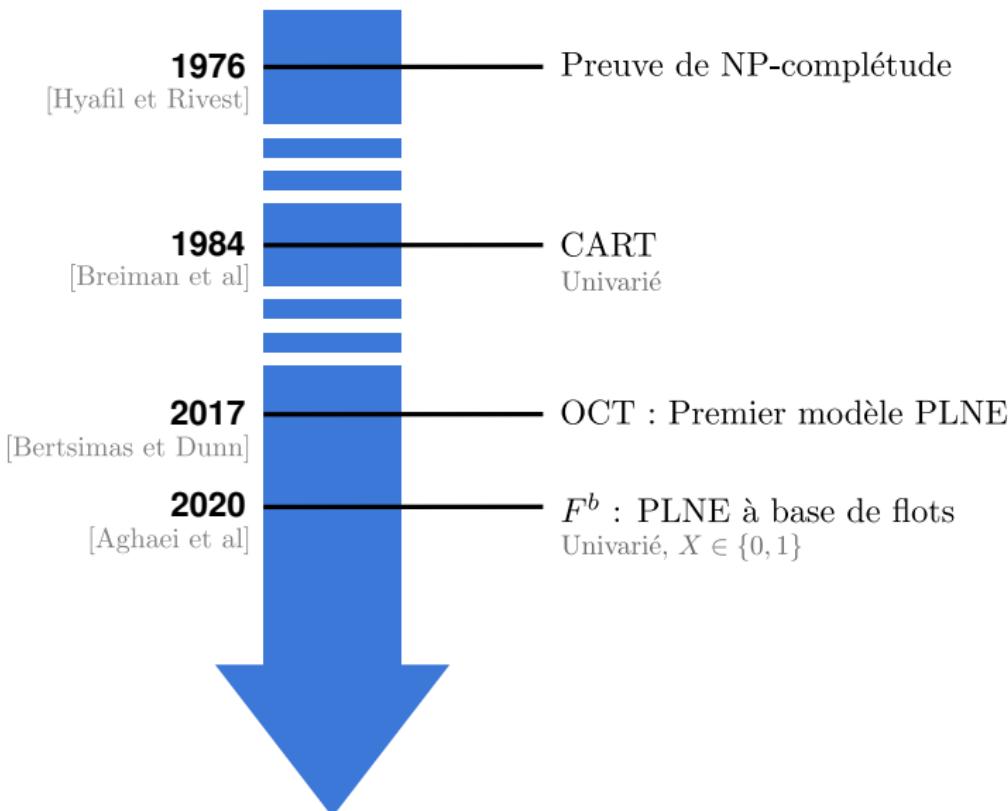
# État de l'art



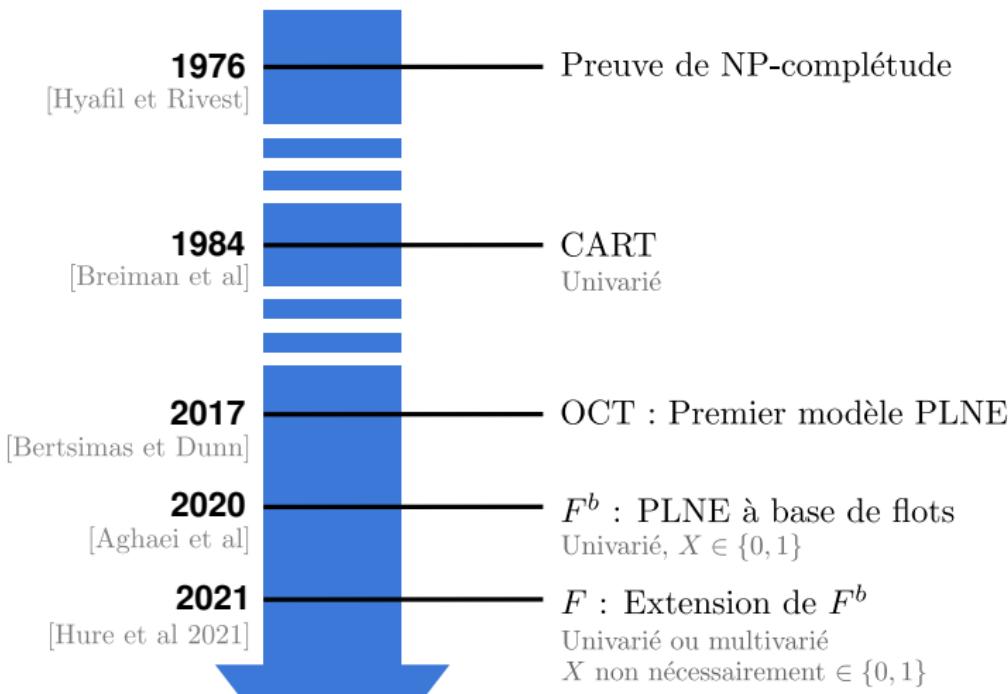
# État de l'art



# État de l'art



# État de l'art



## Objectif

Présentation de  $F$  dans le cas univarié

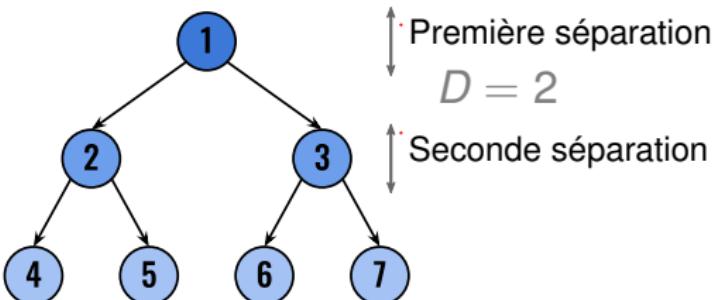
Remarque : les caractéristiques  $\{x_{i,j}\}_{i \in \mathcal{I}, j \in \mathcal{J}}$  sont ici ramenées sur  $[0, 1]$

## Notations

- $\mathcal{I}$  : indice des données
- $\mathcal{J}$  : indice des caractéristiques
- $\mathcal{K}$  : indice des classes

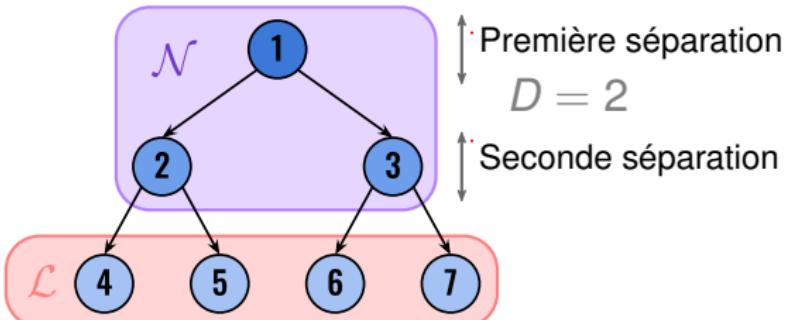
## Modélisation par un graphe $G = (V, A)$

- $D$  : Nombre de séparation d'une branche  
= Profondeur de l'arbre – 1



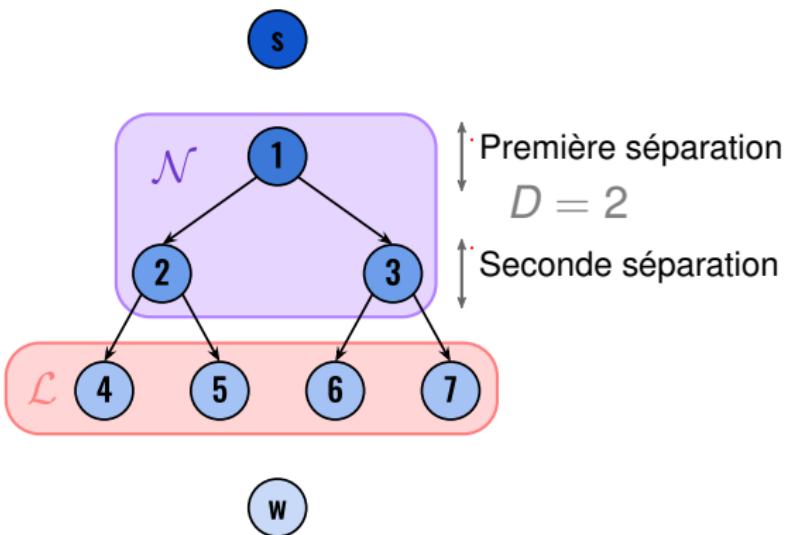
## Modélisation par un graphe $G = (V, A)$

- $D$  : Nombre de séparation d'une branche  
= Profondeur de l'arbre – 1
  - └ Noeuds internes de l'arbre
- $V = \mathcal{N} \cup \mathcal{L}$ 
  - └ Feuilles de l'arbre



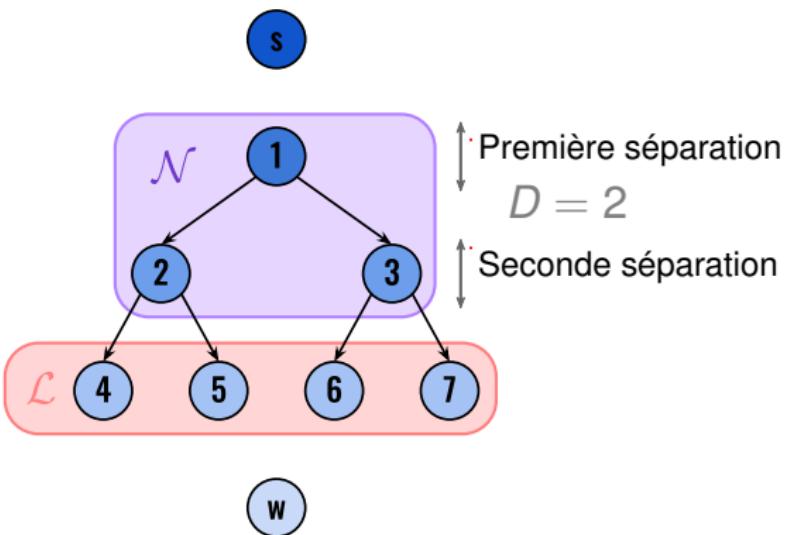
## Modélisation par un graphe $G = (V, A)$

- $D$  : Nombre de séparation d'une branche  
= Profondeur de l'arbre – 1
  - └ Noeuds internes de l'arbre
- $V = \mathcal{N} \cup \mathcal{L} \cup \{s, w\}$ 
  - ↑ Feuilles de l'arbre



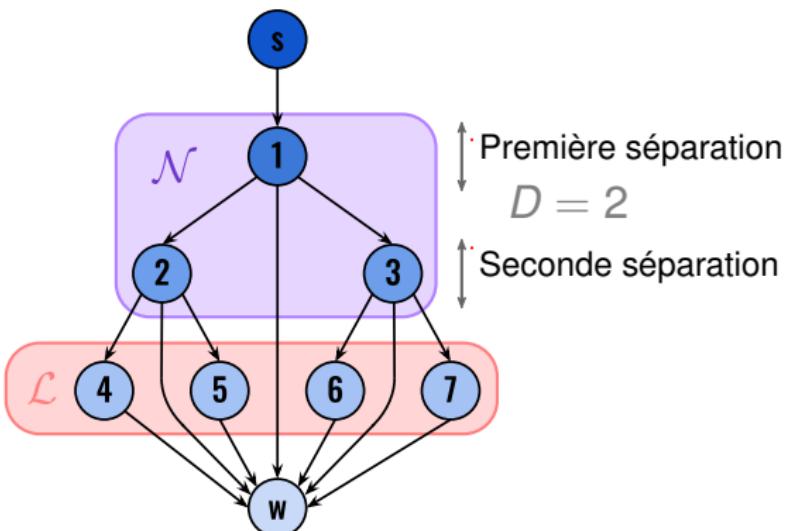
## Modélisation par un graphe $G = (V, A)$

- $D$  : Nombre de séparation d'une branche  
= Profondeur de l'arbre – 1
  - └ Noeuds internes de l'arbre
- $V = \mathcal{N} \cup \mathcal{L} \cup \{s, w\}$ 
  - ↑ Feuilles de l'arbre
- $A = T$



## Modélisation par un graphe $G = (V, A)$

- $D$  : Nombre de séparation d'une branche  
= Profondeur de l'arbre – 1
  - └ Noeuds internes de l'arbre
- $V = \mathcal{N} \cup \mathcal{L} \cup \{s, w\}$ 
  - ↑ Feuilles de l'arbre
- $A = T \cup \{(s, 1)\} \cup \{(v, w) \mid v \in V \setminus \{s, w\}\}$



# Variables de flots

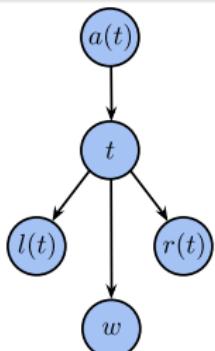
1 flot de  $s$  à  $w$  est associé à chaque donnée  $i \in \mathcal{I}$

- $u_a^i$  : variable de flot de la donnée  $i$  sur l'arc  $a \in A$
- Valeur du flot de la donnée  $i = \begin{cases} 1 & \text{si } i \text{ est correctement classifiée par l'arbre} \\ 0 & \text{sinon} \end{cases}$
- Conservation du flot :

$$u_{a(t),t}^i = u_{t,l(t)}^i + u_{t,r(t)}^i + u_{t,w}^i \quad \forall t \in \mathcal{N}$$

↑                           ↑                           ↑  
Ancêtre de  $t$               Fils gauche de  $t$               Fils droit de  $t$

$$u_{a(t),t}^i = u_{t,w}^i \quad \forall t \in \mathcal{L}$$



Fonction objectif

$$\max \sum_{i \in \mathcal{I}} u_{s,1}^i$$

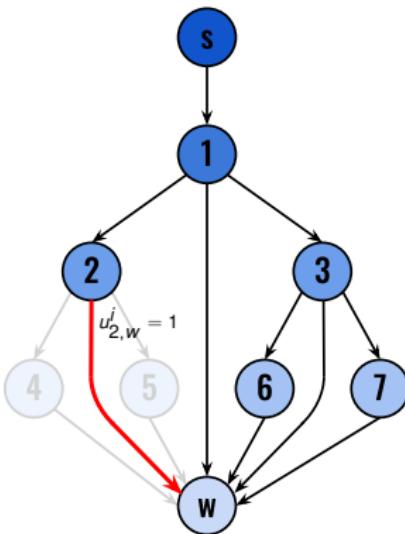
Maximiser les données correctement classées

## Une solution ne contient pas nécessairement tous les sommets de l'arbre

On associe des variables à chaque sommet

mais si  $\exists i \in \mathcal{I}$  tel que  $u_{t,w}^i = 1$ , alors on impose que

- $t$  soit une feuille
- les descendants de  $t$  ne fassent pas partie de la solution



# Variables de séparation $a^T x \leq b$

## Séparation en $t \in \mathcal{N}$

- Variables

- $a_{j,t} = \begin{cases} 1 & \text{si séparation en } t \text{ sur la caractéristique } j \\ 0 & \text{sinon} \end{cases} \quad \forall j \in \mathcal{J}$
- $b_t \in [0, 1]$  : second membre de la séparation

- Contraintes

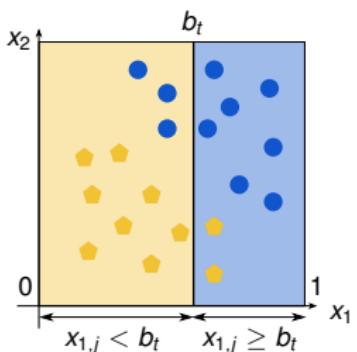
↓ Coefficient suffisamment grand

- $\sum_{j \in \mathcal{J}} a_{j,t} x_{i,j} < b_t + M_1(1 - u_{t,l(t)}^i) \quad \forall i \in \mathcal{I}$

Si  $i$  va à gauche ( $u_{t,l(t)}^i = 1$ ) alors  $a^T x_i < b_t$

- $\sum_{j \in \mathcal{J}} a_{j,t} x_{i,j} \geq b_t - M_2(1 - u_{t,r(t)}^i) \quad \forall i \in \mathcal{I}$

Si  $i$  va à droite ( $u_{t,r(t)}^i = 1$ ) alors  $a^T x_i \geq b_t$



# Variables de séparation $a^T x \leq b$

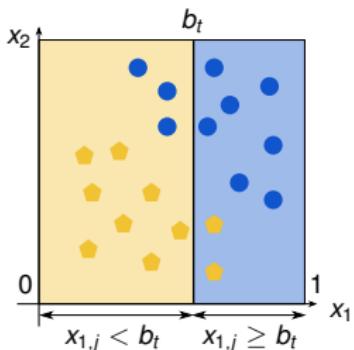
## Séparation en $t \in \mathcal{N}$

- Variables

- $a_{j,t} = \begin{cases} 1 & \text{si séparation en } t \text{ sur la caractéristique } j \\ 0 & \text{sinon} \end{cases} \quad \forall j \in \mathcal{J}$
- $b_t \in [0, 1]$  : second membre de la séparation

- Contraintes

- $\downarrow$  Coefficient suffisamment grand
- $\sum_{j \in \mathcal{J}} a_{j,t} x_{i,j} < b_t + M_1(1 - u_{t,l(t)}^i) \quad \forall i \in \mathcal{I}$   
Si  $i$  va à gauche ( $u_{t,l(t)}^i = 1$ ) alors  $a^T x_i < b_t$
  - $\sum_{j \in \mathcal{J}} a_{j,t} x_{i,j} \geq b_t - M_2(1 - u_{t,r(t)}^i) \quad \forall i \in \mathcal{I}$   
Si  $i$  va à droite ( $u_{t,r(t)}^i = 1$ ) alors  $a^T x_i \geq b_t$



Difficulté de modélisation

Prise en compte d'une inégalité stricte

## Passage en inégalité non-stricte

Ajout d'un paramètre  $\mu \in \mathbb{R}^+$

$$\sum_{j \in \mathcal{J}} a_{j,t} x_{i,j} + \mu \leq b_t + M_1(1 - u_{t,I(t)}^i) \forall i \in \mathcal{I}$$

Doit être  $\leq \min_{j \in \mathcal{J}} \min_{i_1, i_2 \in \mathcal{I}, x_{i_1,j} \neq x_{i_2,j}} |x_{i_1,j} - x_{i_2,j}|$

## Améliorations

① 1  $\mu_j$  par caractéristique  $j \in \mathcal{J}$

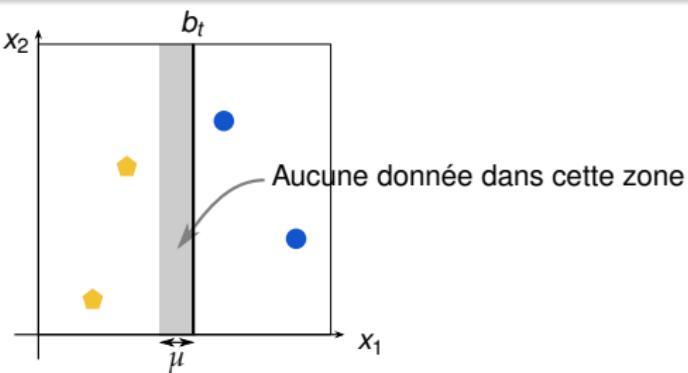
$$\sum_{j \in \mathcal{J}} a_{j,t} (x_{i,j} + \mu_j) \leq b_t + M_1(1 - u_{t,I(t)}^i) \forall i \in \mathcal{I}$$

$\mu_j = \min_{i_1, i_2 \in \mathcal{I}, x_{i_1,j} \neq x_{i_2,j}} |x_{i_1,j} - x_{i_2,j}|$

② Renforcement de la contrainte

$$\sum_{j \in \mathcal{J}} a_{j,t} (x_{i,j} + \mu_j - \mu^-) + \mu^- \leq b_t + M_1(1 - u_{t,I(t)}^i) \forall i \in \mathcal{I} \quad (1)$$

$\mu^- = \min_{j \in \mathcal{J}} \mu_j$



## Passage en inégalité non-stricte

Ajout d'un paramètre  $\mu \in \mathbb{R}^+$

$$\sum_{j \in \mathcal{J}} a_{j,t} x_{i,j} + \mu \leq b_t + M_1(1 - u_{t,I(t)}^i) \forall i \in \mathcal{I}$$

Doit être  $\leq \min_{j \in \mathcal{J}} \min_{i_1, i_2 \in \mathcal{I}, x_{i_1,j} \neq x_{i_2,j}} |x_{i_1,j} - x_{i_2,j}|$

## Améliorations

① 1  $\mu_j$  par caractéristique  $j \in \mathcal{J}$

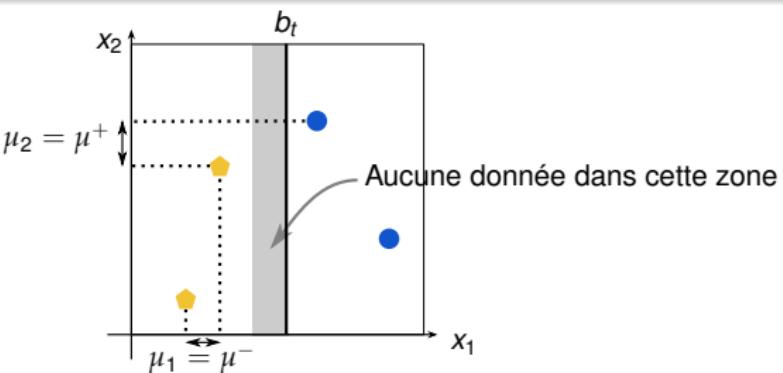
$$\sum_{j \in \mathcal{J}} a_{j,t} (x_{i,j} + \mu_j) \leq b_t + M_1(1 - u_{t,I(t)}^i) \forall i \in \mathcal{I}$$

$\downarrow = \min_{i_1, i_2 \in \mathcal{I}, x_{i_1,j} \neq x_{i_2,j}} |x_{i_1,j} - x_{i_2,j}|$

② Renforcement de la contrainte

$$\sum_{j \in \mathcal{J}} a_{j,t} (x_{i,j} + \mu_j - \mu^-) + \mu^- \leq b_t + M_1(1 - u_{t,I(t)}^i) \forall i \in \mathcal{I} \quad (1)$$

$\downarrow = \min_{j \in \mathcal{J}} \mu_j$



## Fixation de $M_1$

- Si  $u_{t,I(t)}^i = 0$

$$\underbrace{\sum_{j \in \mathcal{J}} a_{j,t} (x_{i,j} + \mu_j - \mu^-) + \mu^- - b_t}_{\leq 1 + \mu^+} \leq M_1$$

$\uparrow$   
 $\max_{j \in \mathcal{J}} \mu_j$

On fixe  $M_1 = 1 + \mu^+$

Car contrainte la plus serrée possible

## Fixation de $M_2$

Si  $u_{t,r(t)}^i = 0$

$$M_2 \geq \underbrace{b_t - \sum_{j \in \mathcal{J}} a_{j,t} x_{i,j}}_{\leq 1}$$

On fixe  $M_2 = 1$

# Variables de classe

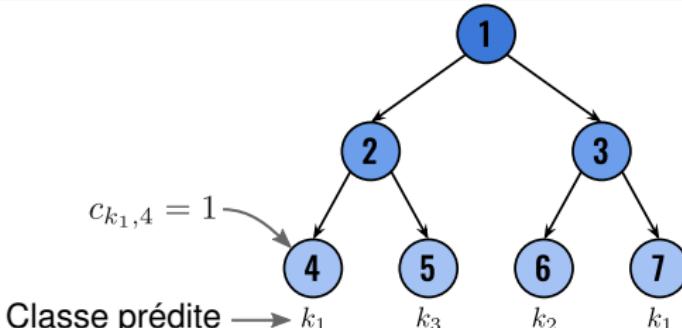
## Classe prédictive en $t$

- Variables  $c_{k,t} = \begin{cases} 1 & \text{si } t \text{ prédict la classe } k \\ 0 & \text{sinon} \end{cases}$
- Contraintes

Un sommet effectue une séparation ou prédict une classe

$$\sum_{j \in \mathcal{J}} a_{j,t} + \sum_{k \in \mathcal{K}} c_{k,t} = 1 \quad \forall t \in \mathcal{N}$$

$$\sum_{k \in \mathcal{K}} c_{k,t} = 1 \quad \forall t \in \mathcal{L}$$



## Autres contraintes de restriction du flot

- $u_{t,r(t)}^i \leq \sum_{j \in \mathcal{J}} a_{j,t} \quad \forall i \in \mathcal{I}, t \in \mathcal{N}$

S'il n'y a pas de séparation en  $t$ , le flot ne va pas à droite

- $b_t \leq \sum_{j \in \mathcal{J}} a_{j,t} \quad \forall t \in \mathcal{N}$

S'il n'y a pas de séparation en  $t$ ,  $b_t = 0$  et grâce à (1), le flot ne va pas à gauche

- $u_{t,w}^i \leq c_{k,t} \quad \forall i \in \mathcal{I} : y_i = k, t \in \mathcal{N} \cup \mathcal{L}$

Une donnée  $i$  de classe  $k$  ne peut emprunter  $(t, w)$  que si  $t$  prédit la classe  $k$

## Variante d'objectif limitant le surapprentissage

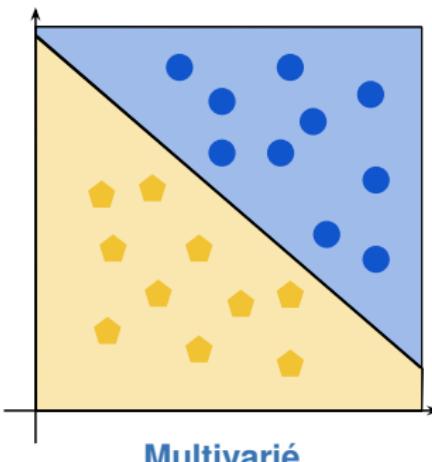
$$\max \underbrace{\sum_{i \in \mathcal{I}} u_{s,1}^i}_{\text{Bonne classifications}} - \lambda \underbrace{\sum_{t \in \mathcal{N}} \sum_{j \in \mathcal{J}} a_{j,t}}_{\substack{\text{Poids du second objectif} \\ \text{Nombre de séparations de l'arbre}}}$$

# Variante multivariée

## Différences

- $a_{j,t} \in [-1, 1]$  au lieu de  $\{0, 1\}$
- Ajout de variables  $\hat{a}_{j,t} = |a_{j,t}|$ ,  $s_{j,t} = \mathbb{1}_{a_{j,t} \neq 0}$  et  $d_t = \mathbb{1}_{\exists j \in \mathcal{J} a_{j,t} \neq 0}$
- $\mu$  devient un paramètre d'entrée :

$$\sum_{j \in \mathcal{J}} a_{j,t} x_{i,j} + \mu \leq b_t + (2 + \mu)(1 - u_{t,I(t)}^i) \quad \forall t \in \mathcal{N}, i \in \mathcal{I}$$



# Performances d'OCT multivarié (extrait de [Bertsimas et Dunn 17])

Dataset	$ \mathcal{I} $	$ \mathcal{J} $	$ \mathcal{K} $	Accuracy		Mean improvement
				CART	OCT-H	
Acute-inflammations-1	120	6	2	78.7	<b>100.0</b>	+21.33 ± 3.09
Acute-inflammations-2	120	6	2	92.0	<b>97.3</b>	+5.33 ± 1.70
Balance-scale	625	4	3	60.9	<b>87.6</b>	+26.75 ± 0.73
Banknote-authentication	1372	4	2	83.6	<b>89.8</b>	+6.18 ± 8.63
Blood-transfusion	748	4	2	75.9	<b>77.2</b>	+1.28 ± 0.69
Breast-cancer-diagnostic	569	30	2	88.5	<b>93.1</b>	+4.62 ± 1.39
Breast-cancer-prognostic	194	32	2	75.5	<b>75.5</b>	0.00 ± 0.00
Breast-cancer	683	9	2	92.2	<b>97.0</b>	+4.80 ± 0.73
Car-evaluation	1728	15	4	69.9	<b>87.5</b>	+17.55 ± 0.35
Chess-king	3196	37	2	66.8	<b>94.9</b>	+28.14 ± 1.41
Climate-model-crashes	540	18	2	91.9	<b>93.2</b>	+1.33 ± 0.82
Congressional-voting-records	232	16	2	<b>98.6</b>	<b>98.6</b>	0.00 ± 0.00
Connectionist-bench-sonar	208	60	2	<b>70.4</b>	<b>70.4</b>	0.00 ± 1.49
Connectionist-bench	990	10	11	<b>16.2</b>	<b>16.2</b>	0.00 ± 0.00
Contraceptive-method-choice	1473	11	3	42.8	<b>45.4</b>	+2.55 ± 1.66

# Résultats théoriques

## Proposition

- Cas univarié

$$\begin{array}{c} \text{Valeure optimale de l'objectif pour la formulation } F \\ \downarrow \\ v(F) = val(OCT) \\ \uparrow \\ \text{Formulation de [Bertsimas et Dunn 17]} \end{array}$$

- Cas multivarié

$$val(F - H) \leq val(OCT - H)$$

## Proposition

- Cas univarié

$$\begin{array}{c} \text{Relaxation linéaire de } F \\ \downarrow \\ val(\overline{F}) > val(\overline{OCT}) \end{array}$$

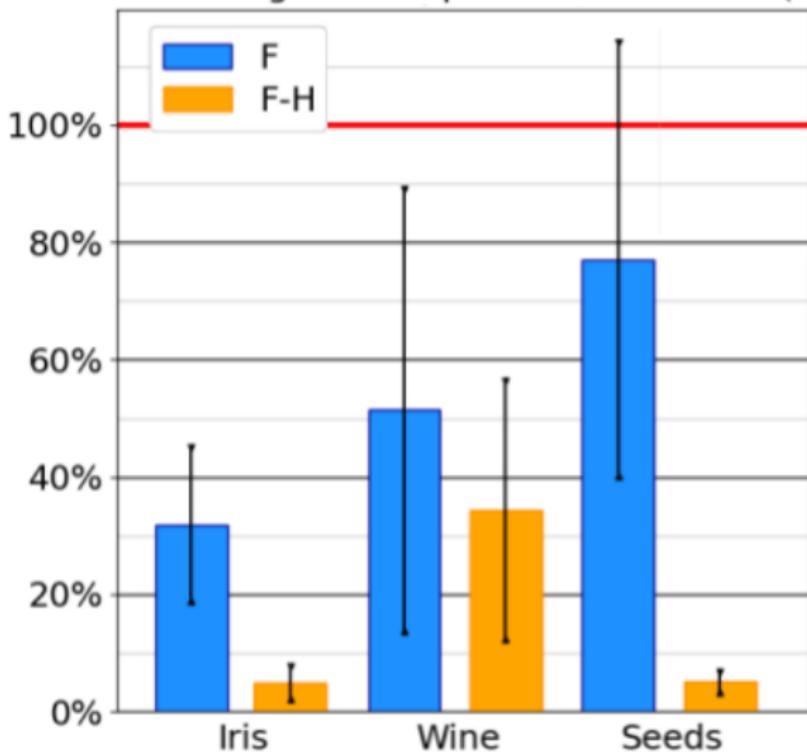
- Cas multivarié

$$val(\overline{F - H}) > val(\overline{OCT - H})$$

## Jeux de données considérés

Jeu de données	Caractéristiques du jeu de données			
	Train	Test	Attributs	Classes
Iris	120	30	4	3
Seeds	168	42	7	3
Wine	142	36	13	3

### Pourcentage du temps de calcul de OCT(-H)



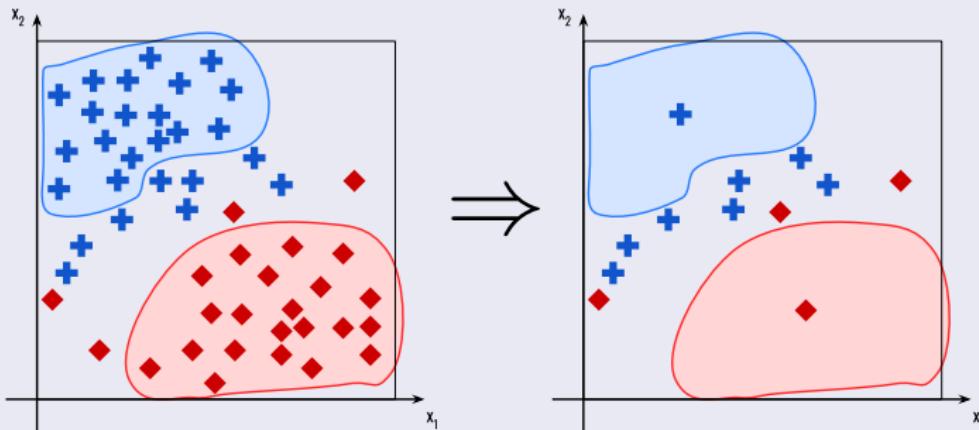
## Limitation de cette approche

Temps de calcul prohibitifs pour des jeux de données de plus grande taille

## Comment passer à l'échelle ?

- Nombre de contraintes de la formulation :  $\mathcal{O}(2^D \times |\mathcal{I}|)$
- **Remarque** :  $2^D \ll |\mathcal{I}| \Rightarrow$  On souhaite réduire  $|\mathcal{I}|$

## Regroupement de données



# Regroupement de données

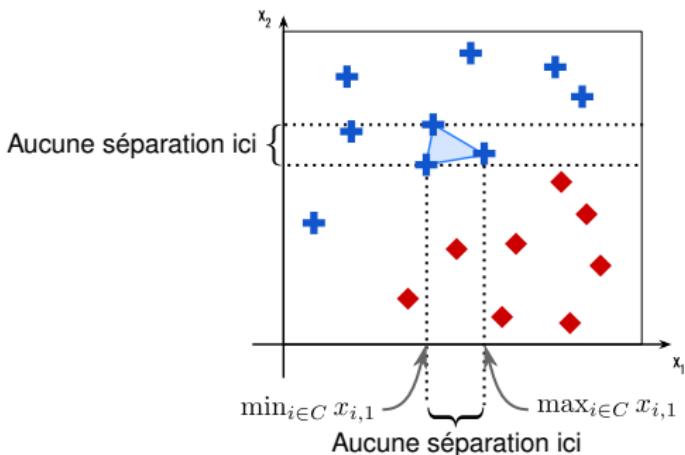
## Adaptation de la formulation

Soit  $C = \{i_1, \dots, i_{|C|}\}$  un cluster de données

- On associe 1 unique flot à  $C$   
Variable  $u_a^C \forall a \in A$
- Adaptation des contraintes de branchement pour  $C$

$$\sum_{j \in \mathcal{J}} a_{j,t} \max_{i \in C} x_{i,j} < b_t + M_1(1 - u_{t,l(t)}^C)$$

$$\sum_{j \in \mathcal{J}} a_{j,t} \min_{i \in C} x_{i,j} \geq b_t - M_2(1 - u_{t,r(t)}^C)$$



## Peut-on regrouper des données tout en garantissant l'optimalité ?

### Hypothèse $H_1$

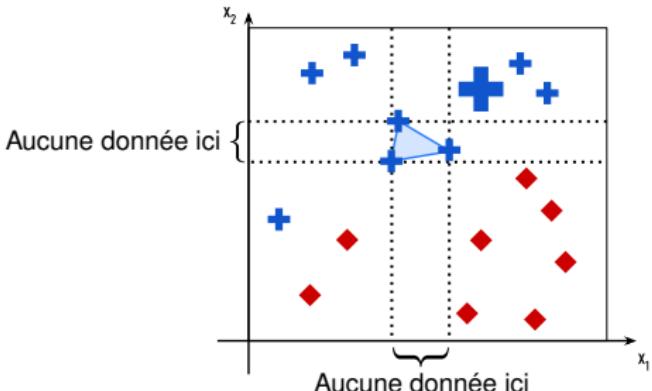
Un cluster de données  $C = \{i_1, \dots, i_{|C|}\}$  vérifie  $H_1$  si

- Toutes les données de  $C$  sont de même classe
- $\forall i \notin C, \forall j \in \mathcal{J}, x_{i,j} \notin [\min_{i_c \in C} x_{i_c}, \max_{i_c \in C} x_{i_c}]$

### Propriété 2

Si  $C$  vérifie  $H_1$ , il existe nécessairement un arbre de décision optimal ne séparant pas  $C$

i.e., dans lequel toutes les données de  $C$  atteignent la même feuille



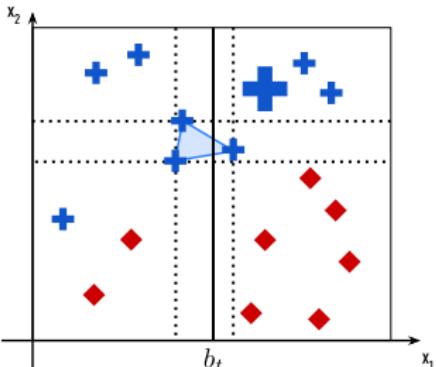
## Propriété 1

Si  $C$  vérifie  $H_1$ , alors pour tout arbre de décision univarié  $T$ , les données de  $C$  peuvent être regroupées dans une même feuille sans impacter le chemin des données  $\mathcal{I} \setminus C$  dans l'arbre

## Démonstration

i.e., telle que  $b_t \in [\min_{i_c \in C} x_{i,j}, \max_{i_c \in C} x_{i,j}]$

Chaque séparation  $x_{i,j} < b_t$  séparant  $C$  peut être modifiée pour que  $C$  ne soit plus séparé, sans que cela n'impacte le chemin des autres données  
e.g., en fixant  $b_t = \min_{i_c \in C} x_{i,j}$  ou  $\max_{i_c \in C} x_{i,j}$



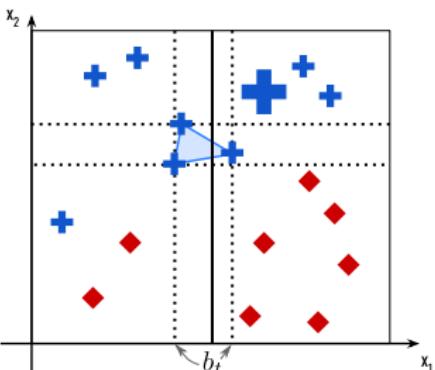
## Propriété 1

Si  $C$  vérifie  $H_1$ , alors pour tout arbre de décision univarié  $T$ , les données de  $C$  peuvent être regroupées dans une même feuille sans impacter le chemin des données  $\mathcal{I} \setminus C$  dans l'arbre

## Démonstration

i.e., telle que  $b_t \in [\min_{i_c \in C} x_{i,j}, \max_{i_c \in C} x_{i,j}]$

Chaque séparation  $x_{i,j} < b_t$  séparant  $C$  peut être modifiée pour que  $C$  ne soit plus séparé, sans que cela n'impacte le chemin des autres données  
e.g., en fixant  $b_t = \min_{i_c \in C} x_{i,j}$  ou  $\max_{i_c \in C} x_{i,j}$



## Propriété 2

Si  $C$  vérifie  $H_1$ , il existe nécessairement un arbre de décision optimal ne séparant pas  $C$  i.e., dans lequel toutes les données de  $C$  atteignent la même feuille

### Démonstration

Supposons que tout les arbres de décision optimaux séparent  $C$ .

Soit  $T_{opt}$  un de ces arbres.

#### Cas 1 - $\exists c \in C$ bien classifié par $T_{opt}$

D'après la Propriété 1, on peut regrouper toutes les données de  $C$  dans une même feuille sans impacter le chemin des autres données. Cet arbre est également optimal → contradiction

#### Cas 2 - $\exists c \in C$ bien classifié par $T_{opt}$

On regroupe toutes les données de  $C$  dans la feuille de  $c$  sans changer le chemin des autres données.

⇒ toutes les données de  $C$  sont correctement classifiées

Le nombre de bonnes prédictions des données de  $\mathcal{I} \setminus C$  n'est pas impacté. En effet, pour une feuille  $F$  perdant des données de  $C$ , trois cas sont possibles suivant que la classe de  $C$  :

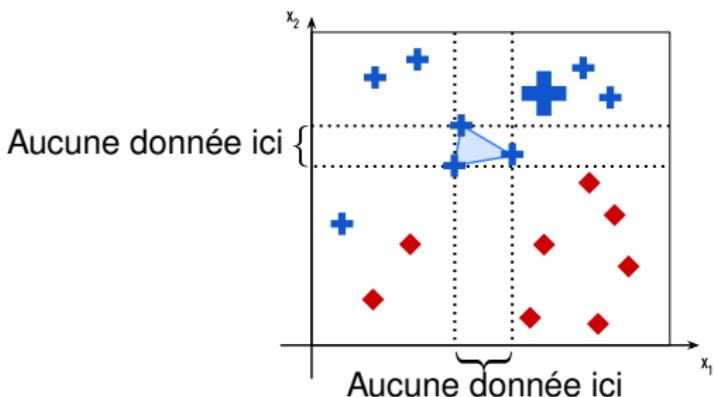
- n'était pas majoritaire dans  $F$
- était majoritaire et
  - reste majoritaire
  - ne reste pas majoritaire

Dans ces trois cas, le nombre de bonnes prédictions de  $\mathcal{I} \setminus C$  n'est pas modifié

## Remarque

- Il est donc possible de regrouper des données tout en garantissant l'optimalité
- Malheureusement la condition  $H_1$  est très restrictive  
Jamais vérifiée pour les jeux de données Iris, Wine et Seeds

## Approche non optimale testée en projet



# Sommaire

- 1 Introduction à la classification supervisée
- 2 Construction d'arbres de décision optimaux
- 3 Projet
  - Sujet
  - Julia

# Sommaire

1 Introduction à la classification supervisée

2 Construction d'arbres de décision optimaux

3 Projet

- Sujet

- Julia

## Informations générales

### Groupe

- Seul ou en binôme

### Langage

- Libre

Code Julia fourni

### Calendrier

- 02/03 : ~1h30 de TP
- 09/03 : 3h de TP (présentation de l'avancement)
- 31/03 : date limite de rendu

# Regroupement

## Travail demandé

- 1 Appliquer  $F$  à ces jeux de données

Code fourni

- 2 Appliquer  $F$  avec regroupement

Code fourni (méthode de regroupement naïve)

- 3 Appliquer  $F$  avec et sans regroupements à d'autres jeux de données (au moins 2)

Vous pouvez en trouver ici : <https://archive.ics.uci.edu/ml/datasets.php>

Ne pas oublier de ramener les caractéristiques  $\{x_{i,j}\}_{i \in \mathcal{I}, j \in \mathcal{J}}$  dans  $[0, 1]$

- 4 Traiter une question d'ouverture au choix :

- 1 Proposer et tester d'autre(s) méthode(s) de regroupement
- 2 Résultats théoriques de regroupement(s)
- 3 Implémentation de l'heuristique figurant dans [Dunn 2018]
- 4 Identification et utilisation d'inégalités valides intéressantes
- 5 Tout autre idée permettant d'améliorer les temps de calculs ou la qualité des prédictions

## Remarque

Dans le code, un recentrage des séparations est effectué en post-traitement pour avoir des séparations passant aussi loin que possible des données

Pour tenter de limiter le surapprentissage

# Ouverture 1 : Autres méthodes de regroupement

## Algorithme naïf fourni

### Data :

$\{(x_i, y_i)\}_{i \in \mathcal{I}}$  : jeu de données

$\gamma \in [0, 1]$  : pourcentage de regroupements

### Result :

$\mathcal{C}$  : partition des données

$\mathcal{C} \leftarrow \{C_i = \{i\}\}_{i \in \mathcal{I}} \leftarrow 1$  cluster par donnée  $i \in \mathcal{I}$

**tant que**  $|\mathcal{C}| \geq \gamma |\mathcal{I}|$  **faire**

Fusionner les deux clusters  $C$  et  $C'$  de même classe minimisant

$\min_{i \in C} \min_{i' \in C'} \|x_i - x_{i'}\|_2$

## Travail demandé

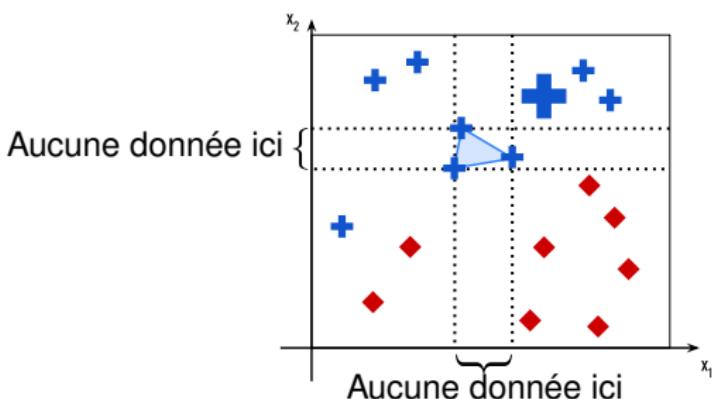
Proposer d'autre(s) méthode(s) de regroupement et comparer leurs performances à celles des méthodes fournies

## Ouverture 2 : Résultats théoriques de regroupement(s)

### Travail demandé

Trouver une ou plusieurs hypothèses similaires à  $H_1$  permettant de garantir :

- Que l'optimalité soit conservée ou
- Que la qualité de la prédiction n'est pas trop détériorée  
i.e., obtenir une borne sur la détérioration du nombre de bonnes prédictions



# Ouverture 3 : Heuristique de [Dunn 18]

## Contexte

- Dunn constate qu'OCT ne passe pas à l'échelle
- Il propose une heuristique de recherche locale ayant de bonnes performances

[Algorithme 2.1 en Section 2.3 de \[Dunn 18\]](#)

## Travail demandé

Implémenter cette heuristique et comparer ses performances à celles des méthodes fournies

## Ouverture 4 : Inégalités valides

### Contexte

- Le relaxation linéaire des PLNE de construction d'arbres de décision optimaux est très mauvaise  
*Gap élevé à la racine*
- Une meilleure gestion des contraintes pourrait améliorer les performances

### Travail demandé

- Identifier, puis ajouter des inégalités valides à la formulation
  - Statiquement (lors de la construction initiale du modèle) ou
  - Dynamiquement (lors de la résolution)  
*Dans un callback pour couper des points fractionnaires*
- Retirer des inégalités de la formulation et les ajouter au cours de la résolution  
*Dans un callback pour couper des points entiers*

Comparer les performances aux méthodes fournies

## Ouverture 5 : Autres idées

### Travail demandé

Proposer et tester des idées permettant l'amélioration des performances

# Sommaire

1 Introduction à la classification supervisée

2 Construction d'arbres de décision optimaux

3 Projet

• Sujet

• Julia

# Langage Julia

## Avantages

- Performant  
*Comparable au C++*
- Syntaxe simple et efficace
- De plus en plus répandu  
*Surtout dans la communauté académique*
- Facilité de développement et d'utilisation de packages

## Package JuMP

Package de Julia permettant de résoudre des problèmes d'optimisation

- Mêmes avantages que Julia  
*Performant, syntaxe aisée*
- Indépendant du solveur  
*Simple de passer de l'un à l'autre*

## Déclarer une variable

```
n = 10 # entier  
b = "Hello world" # chaîne de caractères  
v = [1 2 3 4] # vecteur  
m = [1 2; 3 4] # matrice 2x2
```

## Inclure un fichier contenant des variables

```
include("monFichier.dat")
```

## Affichage

```
println("Afficher du texte")  
println("Afficher une variable $a")  
println("ou ", a)
```

## Écrire dans un fichier

```
fout = open("monFichierDeSortie.dat", "a")  
print(fout, v)  
# Remarque :  
# Remplacer "a" par "w" pour écraser l'ancien contenu du fichier
```

## Conditionnelle

```
if v[1] == 1
    # contenu du if
else
    # contenu du else
end
```

## Boucle for

```
for i in 1:10 # ou i = 1:10
    print(i)
end
```

## Boucle while

```
while v[1] == 1
    # contenu de la boucle
end
```

## Déclarer un problème d'optimisation avec CPLEX

```
using JuMP  
using CPLEX  
m = Model(CPLEX.Optimizer)
```

## Déclarer des variables d'un problème d'optimisation

```
# Variable continue  
@variable(m, 0 <= x1 <= 1)  
  
# Variable binaire  
@variable(m, x2, Bin)  
  
# Tableau n*1  
@variable(m, 0 <= y[i in 1:n] <= 1)  
  
# Tableau n*4  
@variable(m, 0 <= t[i in 1:n, j in 1:4] <= 1)
```

## Définir des contraintes

```
# x1 + x2 = 1
@constraint(m, x1 + x2 == 1)

# y_i + x1 ≤ 1 ∀i{1,...,n}
@constraint(m, [i = 1:n], y[i] + x1 <= 1)

# t_{ij} + x1 ≥ 1 ∀i{1,...,n} ∀j{1,...,4}
@constraint(m, [i = 1:n, j = 1:4], t[i, j] + x1 >= 1)

# ∑_{i=1}^n y_i ≥ 3
@constraint(m, sum(y[i] >= 3 for i in 1:n))
```

## Définir l'objectif

```
@objective(m, Max, sum(y[i] for i = 1:n))

# objectif avec condition
@objective(m, Max, sum(y[i] for i = 1:n if v[i] == 2))
```

## Résoudre un problème

```
optimize!(m)
```

### Obtenir la valeur d'une variable x1

```
vx1 = JuMP.value(x1)  
vx1Int = round(Int, JuMP.value(x1))
```

### Obtenir la valeur d'un tableau de variables tx

```
vtx = JuMP.value.(tx)  
vtxInt = round.(Int, JuMP.value.(tx))
```

### Masquer les sorties de CPLEX

```
set_optimizer_attribute(m, "CPX_PARAM_SCRIND", 0)
```

### Limiter le temps d'exécution à 30 secondes

```
set_optimizer_attribute(m, "CPX_PARAM_TILIM", 30)
```

# Problème de sac à dos

## Fichier knapsack.jl

```
using JuMP
using CPLEX

include("donnees.dat")

m = Model(CPLEX.Optimizer)

@variable(m, x[i in 1:n], Bin)
@constraint(m, sum(x[i] * w[i] for i = 1:n) <= K)
@objective(m, Max, sum(x[i] * p[i] for i in 1:n))
optimize!(m)
```

## Fichier donnees.dat

```
n = 6
K = 23
w = [1 2 4 5 7 10]
p = [1 3 5 7 9 11]
```

## Éxécuter ce fichier à l'ENSTA

- ➊ Ouvrir une console : Alt + F2, puis entrer "xterm"
- ➋ Fixer les chemins (pour les ordinateurs de l'ENSTA) : `usediam ro`
- ➌ Ajouter les packages nécessaires (à ne faire qu'une fois) :

```
julia
using Pkg
Pkg.add("JuMP")
Pkg.add("CPLEX")
```

- ➍ Éxécuter le programme :
 

```
julia knapsack.jl # ou include("knapsack.jl") si vous êtes en mode console
```

# Deux types d'exécutions

## 1 - Commande julia

```
login@pc $ julia knapsack.jl
```

## 2 - Mode console

```
login@pc $ julia
julia> include("knapsack.jl")
```



### Avantages du mode console

- Plus pratique pour tester des commandes
- Librairies chargées une seule fois  
*Sinon prend plusieurs secondes à chaque exécutions*

### Désavantages du mode console

- Doit être relancé en cas de redéfinition d'une structure
- Potentiels effets indésirables si variables fixées avant d'inclure un fichier

## Références

-  [Jack Dunn.](#)  
Optimal Trees for Prediction and Prescription.  
PhD. Thesis, 2014.
-  [Dimitris Bertsimas and Jack William Dunn.](#)  
Optimal classification trees.  
In Machine Learning, 2017.
-  [Sina Aghaei, Andres Gomez and Phebe Vayanos.](#)  
Learning Optimal Classification Trees : Strong Max-Flow Formulations.  
In arXiv, 2020.

# Formulation $F$ univariée

$$\begin{aligned}
 \max \quad & \sum_{i \in \mathcal{I}} u_{s,1}^i - \lambda \sum_{t \in \mathcal{N}} \sum_{j \in \mathcal{J}} a_{j,t} \\
 \text{s.t.} \quad & \sum_{j \in \mathcal{J}} a_{j,t} + \sum_{k \in \mathcal{K}} c_{k,t} = 1 \quad \forall t \in \mathcal{N} \\
 & \sum_{k \in \mathcal{K}} c_{k,t} = 1 \quad \forall t \in \mathcal{L} \\
 & b_t \leq \sum_{j \in \mathcal{J}} a_{j,t} \quad \forall t \in \mathcal{N} \\
 & u_{a(t),t}^i = u_{t,l(t)}^i + u_{t,r(t)}^i + u_{t,w}^i \quad \forall t \in \mathcal{N}, i \in \mathcal{I} \\
 & u_{a(t),t}^i = u_{t,w}^i \quad \forall t \in \mathcal{L}, i \in \mathcal{I} \\
 & \sum_{j \in \mathcal{J}} a_{j,t} (x_{i,j} + \mu_j - \mu^-) + \mu^- \leq b_t + (1 + \mu^+) (1 - u_{t,l(t)}^i) \quad \forall t \in \mathcal{N}, i \in \mathcal{I} \\
 & a_t^\top X_i \geq b_t - (1 - u_{t,r(t)}^i) \quad \forall t \in \mathcal{N}, i \in \mathcal{I} \\
 & u_{t,r(t)}^i \leq \sum_{j \in \mathcal{J}} a_{j,t} \quad \forall i \in \mathcal{I}, t \in \mathcal{N} \\
 & u_{t,w}^i \leq c_{k,t} \quad \forall i \in \mathcal{I} : y_i = k, t \in \mathcal{N} \cup \mathcal{L} \\
 & a_{j,t} \in \{0, 1\} \quad \forall t \in \mathcal{N}, j \in \mathcal{J} \\
 & c_{k,t} \in \{0, 1\} \quad \forall t \in \mathcal{N} \cup \mathcal{L}, k \in \mathcal{K} \\
 & u_e^i \in \{0, 1\} \quad \forall e \in \mathcal{E}, i \in \mathcal{I}
 \end{aligned}$$

# Formulation $F$ multivariée

$$\begin{aligned}
 \max \quad & \sum_{i \in \mathcal{I}} u_{s,1}^i - \lambda \sum_{t \in \mathcal{N}} \sum_{j \in \mathcal{J}} s_{j,t} \\
 \text{s.t.} \quad & d_t + \sum_{k \in \mathcal{K}} c_{k,t} = 1 \quad t \in \mathcal{N} \\
 & \sum_{k \in \mathcal{K}} c_{k,t} = 1 \quad t \in \mathcal{L} \\
 & \sum_{j \in \mathcal{J}} a_{j,t} \leq d_t \quad t \in \mathcal{N} \\
 & -\hat{a}_{j,t} \leq a_{j,t} \leq \hat{a}_{j,t} \quad j \in \mathcal{J}, t \in \mathcal{N} \\
 & -s_{j,t} \leq a_{j,t} \leq s_{j,t} \quad j \in \mathcal{J}, t \in \mathcal{N} \\
 & s_{j,t} \leq d_t \quad j \in \mathcal{J}, t \in \mathcal{N} \\
 & \sum_{j \in \mathcal{J}} s_{j,t} \geq d_t \quad t \in \mathcal{N} \\
 & -d_t \leq b_t \leq d_t \quad t \in \mathcal{N} \\
 & u_{a(t),t}^i = u_{t,l(t)}^i + u_{t,r(t)}^i + u_{t,w}^i \quad t \in \mathcal{N}, i \in \mathcal{I} \\
 & u_{a(t),t}^i = u_{t,w}^i \quad t \in \mathcal{L}, i \in \mathcal{I}
 \end{aligned}$$

Données de class  $k$

$\downarrow$

$i \in \mathcal{I}, t \in \mathcal{N}$

$i \in \mathcal{I}^k, t \in \mathcal{N} \cup \mathcal{L}$

$t \in \mathcal{N}, j \in \mathcal{J}$

$t \in \mathcal{N}$

$t \in \mathcal{N} \cup \mathcal{L}, k \in \mathcal{K}$

$e \in \mathcal{E}, i \in \mathcal{I}$