Arbres de décision optimaux

SOD322 - RO et données massives

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

Créé le 22/02/2022
Modifié le 16/02/2023 (v2)

Sommaire

Définition - Classification supervisée

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

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

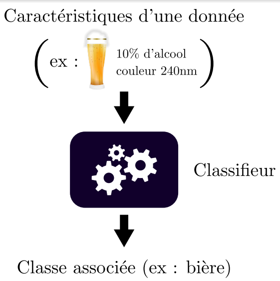
Classifieur

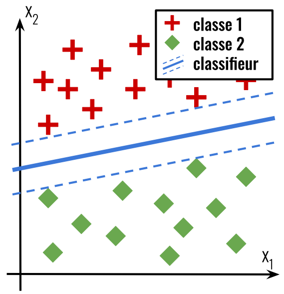
Exemple - Classe

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

Exemple - Caractéristiques

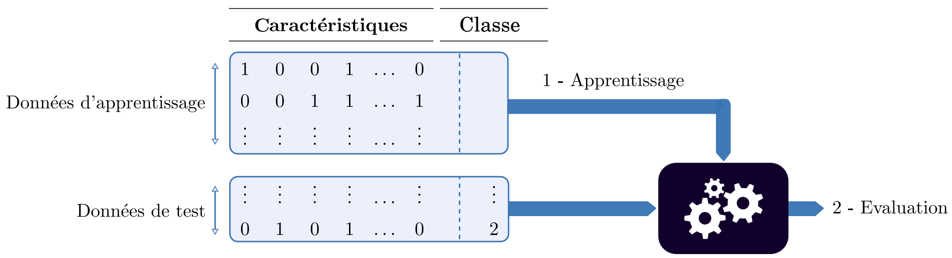
- Taux d'alcool, couleur
- Ratio hauteur/longueur, forme, ...





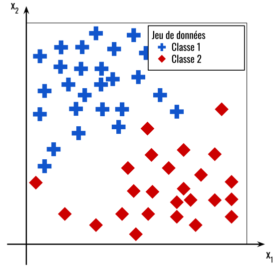
Partage des données

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

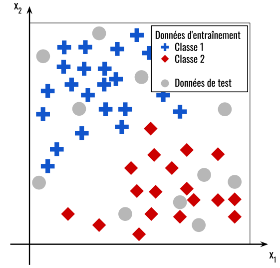


La classification supervisée en 5 minutes | Construction d'arbres de décision optimaux | Projet | Annexes

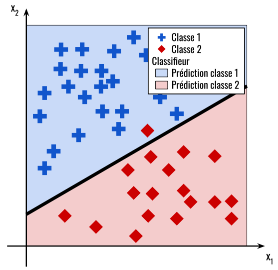
Évaluation des classifieurs



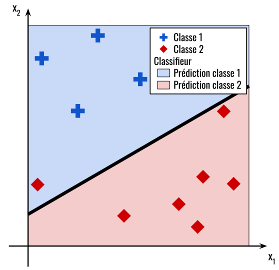
Données initiales



Partition des données en 2 ensembles



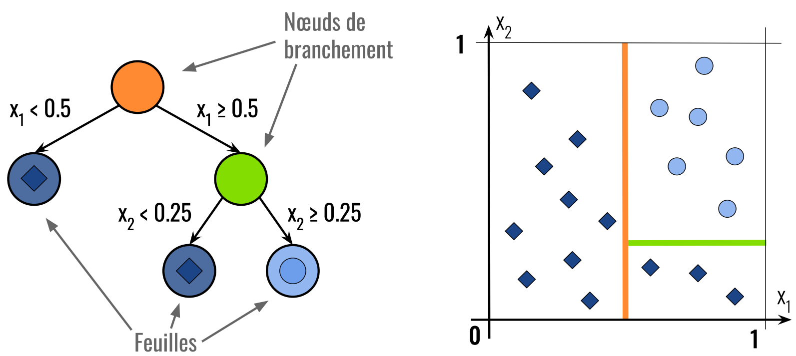
Apprentissage du classifieur sur les données d'entraînement



Prédictions du classifieur sur les données de test

La classification supervisée en 5 minutes | Construction d'arbres de décision optimaux | Projet | Annexes

Exemple de classifieur : Arbres de décision



La classification supervisée en 5 minutes | Construction d'arbres de décision optimaux | Projet | Annexes

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)

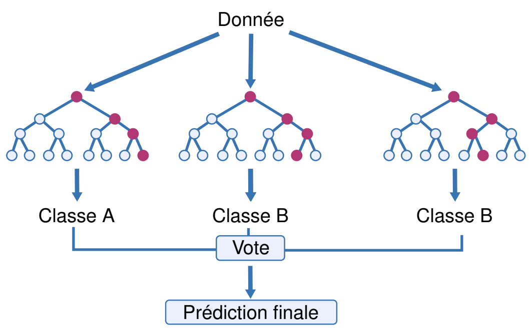
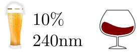


Image inspirée de kdnuggets.com

Exemple de classifieurs : Liste de décision



Règle n°1 : Taux d'alcool $\geq$ 20% → Vin ✗ ✓ Je prédis vin

Règle n°2 : Couleur $\in [200, 300]$nm → Bière ✓ Je prédis bière

⋮

Règle n°L : Taux d'alcool $\leq$ 10% → Bière

Méthode - Ordered Rules for Classification (ORC)

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

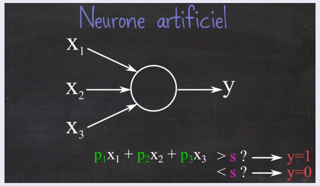
Allison Chang, Dimitris Bertsimas, and Cynthia Rudin.
An integer optimization approach to associative classification.
In Advances in neural information processing systems, pages 269–277, 2012.

Désavantage

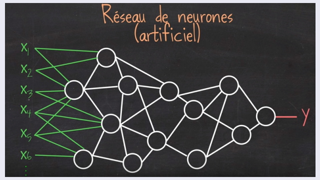
- Résolution exacte lente
- Données sous forme de vecteurs binaires
  One-hot encoding

Exemple de classifieurs : Réseaux de neurones

Classifieur - Réseaux de neurones

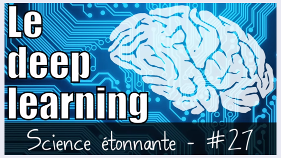


Neurone artificiel



Réseau de neurones

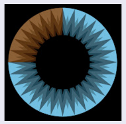
Images issues de



- 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

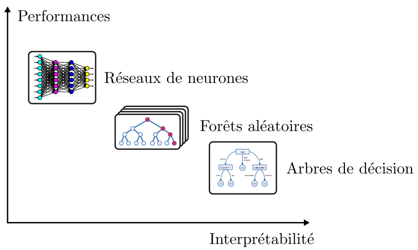
Interprétabilité des classifieurs

Définition - Interprétabilité

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

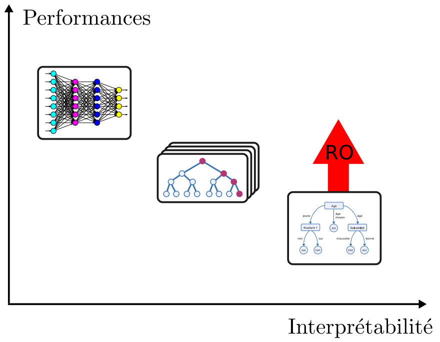
Pourquoi s'intéresser à l'interprétabilité ?

- Important pour la prise de décisions sensibles
  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 classifieurs par des méthodes exactes
- Améliore le pouvoir prédictif de modèles de classifieurs interprétables



Inconvénients

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

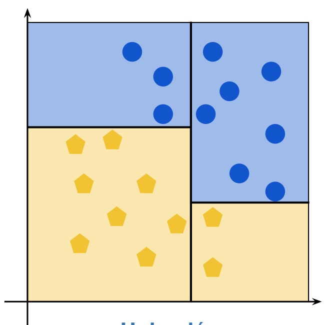
Sommaire

Deux types de séparations

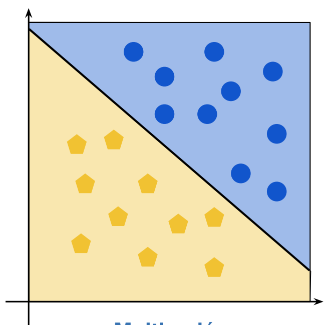
Valeur de l'attribut j de la donnée i

$$\sum_{j \in \mathcal{J}} a_j x_{i,j} < b$$

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

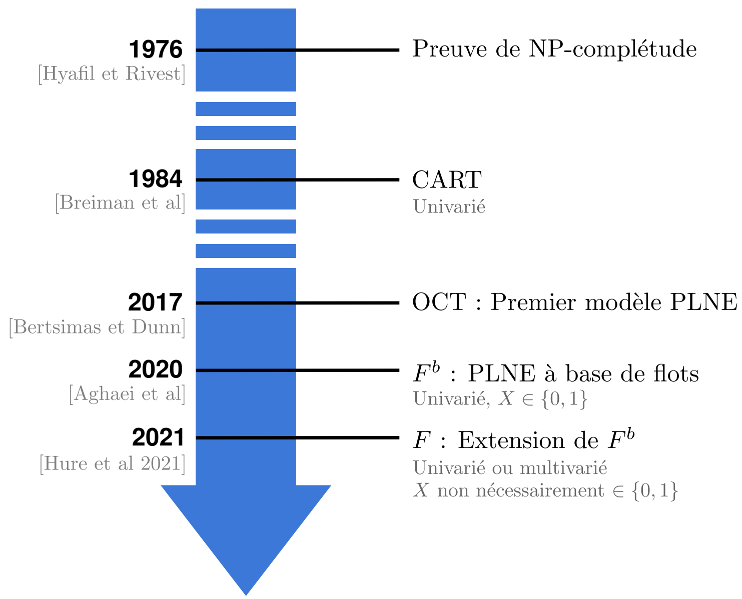


Univarié



Multivarié

État de l'art



Objectif

Présentation de F dans le cas **univarié**

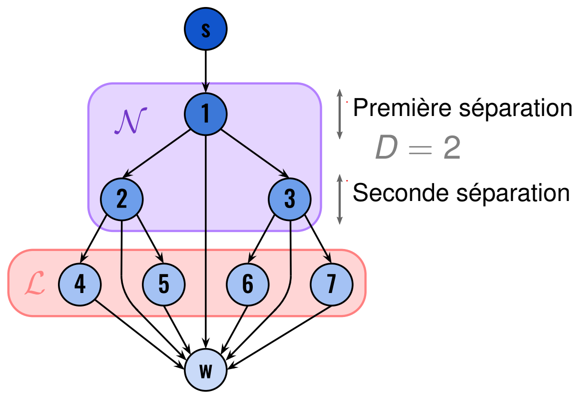
Remarque : les caractéristiques $\{x_{i,j}\}_{i\in\mathcal{I}\ j\in\mathcal{J}}$ sont ici ramenées sur $[0,1]$

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 séparations 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
- $A = T \cup \{(s,1)\} \cup \{(v,w) \mid v \in V \backslash \{s,w\}\}$



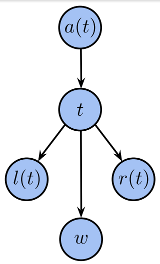
Variables de flots

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 $i = \begin{cases} 1 & \textbf{si } \boldsymbol{i} \textbf{ est correctement classifiée} \text{ par l'arbre} \\ 0 & \text{sinon} \end{cases}$
- Conservation du flot :

$$\underbrace{u^i_{a(t),t}}_{\text{Ancêtre de } t} = \underbrace{u^i_{t,l(t)}}_{\text{Fils gauche de } t} + \underbrace{u^i_{t,r(t)}}_{\text{Fils droit de } t} + u^i_{t,w} \quad \forall t \in \mathcal{N}$$

$$u^i_{a(t),t} = u^i_{t,w} \qquad \forall t \in \mathcal{L}$$



Fonction objectif

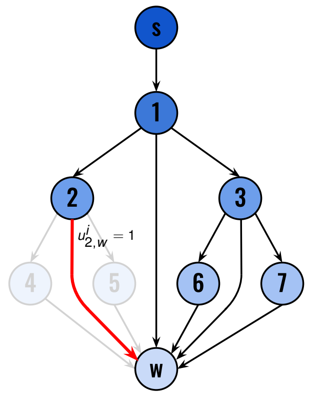
$$\max \sum_{i\in\mathcal{I}} u^i_{s,1}$$

Maximiser les données correctement classées

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

On associe des variables à chaque sommet $t \in \mathcal{N}$
mais s'il existe un flot utilisant l'arc (tw)...
i.e., $\exists i \in \mathcal{I}$ tel que $u^i_{t,w} = 1$
... alors on impose que

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



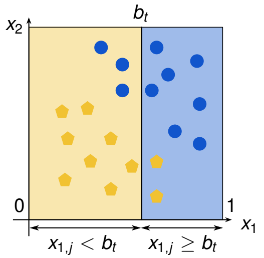
Variables de séparation $a^T x \leq b$

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

    Si i va à gauche ($u^i_{t,l(t)} = 1$) alors $a^T x_i < b_t$
  - $\sum_{j \in \mathcal{J}} a_{j,t} x_{i,j} \geq b_t - M_2(1 - u^i_{t,r(t)}) \quad \forall i \in \mathcal{I}$

    Si i va à droite ($u^i_{t,r(t)} = 1$) 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 $\mu \in \mathbb{R}^+$

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

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

Améliorations

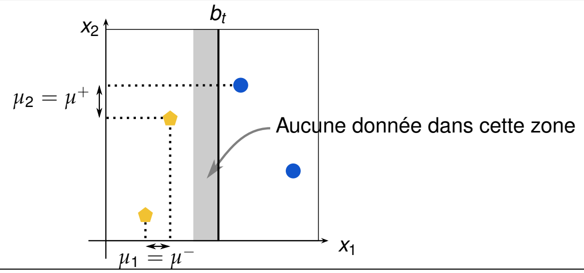
1. 1 μ_j par caractéristique $j \in \mathcal{J}$

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

   $$\sum_{j \in \mathcal{J}} a_{j,t}(x_{i,j} + \mu_j) \leq b_t + M_1(1 - u^i_{t,l(t)}) \qquad \forall i \in \mathcal{I}$$
2. Renforcement de la contrainte

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

   $$\sum_{j \in \mathcal{J}} a_{j,t}(x_{i,j} + \mu_j - \mu^-) + \mu^- \leq b_t + M_1(1 - u^i_{t,l(t)}) \qquad \forall i \in \mathcal{I} \tag{1}$$



Fixation de M_1

- Si $u^i_{t,l(t)} = 0$

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

μ^+ : $\max_{j \in \mathcal{J}} \mu_j$

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

Car contrainte la plus serrée possible

Fixation de M_2

Si $u^i_{t,r(t)} = 0$

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

On fixe $M_2 = 1$

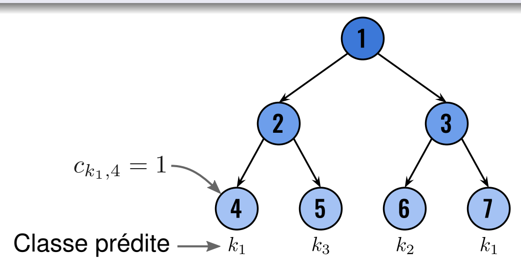
Variables de classe

Classe prédite en t

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

  Un sommet effectue une séparation ou prédit une classe

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



Autres contraintes de restriction du flot

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

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

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

Objectif limitant le surapprentissage

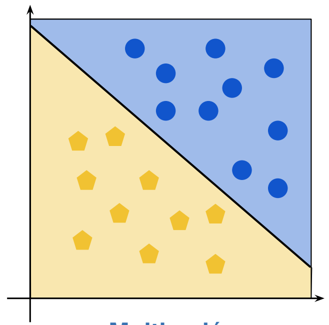
$$\max \overbrace{\sum_{i\in\mathcal{I}} u^i_{s,1}}^{\text{Bonnes classifications}} - \overset{\text{Poids du second objectif}}{\lambda} \underbrace{\sum_{t\in\mathcal{N}} \sum_{j\in\mathcal{J}} a_{j,t}}_{\text{Nombre de séparations de l'arbre}}$$

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 $d_t = \mathbb{1}_{\exists j\in\mathcal{J}\ a_{j,t}\neq 0}$
- μ devient un paramètre d'entrée :

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



Multivarié

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

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

Résultats théoriques

Proposition

- Cas univarié

$$\overset{\text{Valeure optimale de l'objectif pour la formulation } F}{v(F)} = val(\underset{\text{Formulation de [Bertsimas et Dunn 17]}}{OCT})$$

- Cas multivarié

$$val(F\text{-}H) \leq val(OCT\text{-}H)$$

Proposition

- Cas univarié

$$val(\overset{\text{Relaxation linéaire de } F}{\overline{F}}) > val(\overline{OCT})$$

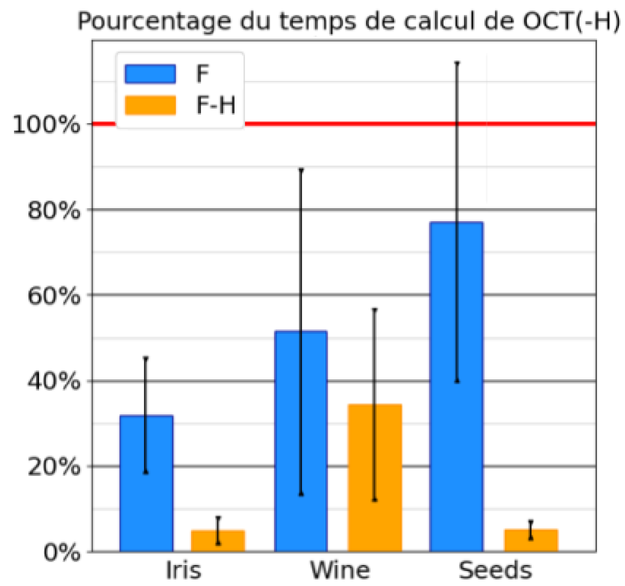
- Cas multivarié

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

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

La classification supervisée en 5 minutes | Construction d'arbres de décision optimaux | Projet | Annexes



La classification supervisée en 5 minutes | Construction d'arbres de décision optimaux | Projet | Annexes

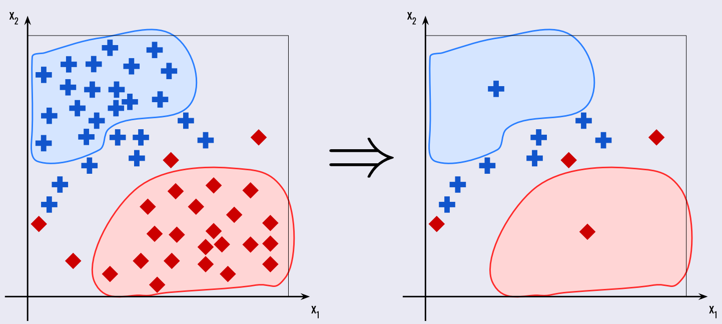
Limitation de cette approche

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** : $2^D << |\mathcal{I}| \Rightarrow$ On souhaite réduire $|\mathcal{I}|$

Regroupement de données



La classification supervisée en 5 minutes | Construction d'arbres de décision optimaux | Projet | Annexes

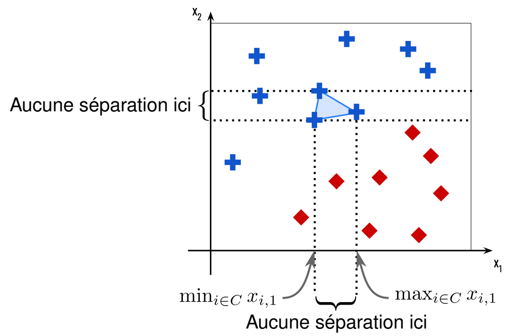
Regroupement de données

Adaptation des contraintes

Soit $C = \{i_1, ..., i_{|C|}\}$ 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

$$\sum_{j \in \mathcal{J}} a_{j,t} \max_{i \in C} x_{i,j} < b_t + M_1(1 - u^C_{t,l(t)})$$
$$\sum_{j \in \mathcal{J}} a_{j,t} \min_{i \in C} x_{i,j} \geq b_t - M_2(1 - u^C_{t,r(t)})$$



Adaptation de l'objectif

$$\max \underbrace{\sum_{C \in \mathcal{C}}}_{\text{Ensemble des clusters}} |C| u^C_{s,1}$$

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

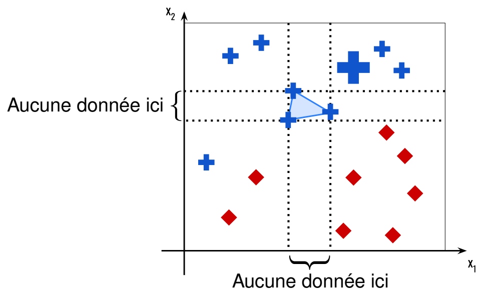
Hypothèse H_1

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

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

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

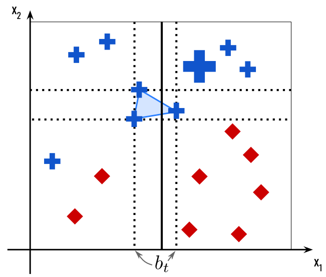


Propriété 1

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 $\mathcal{I} \backslash C$ dans l'arbre

Démonstration

Chaque séparation $x_{i,j} < b_t$ $\overbrace{\text{séparant } C}^{\text{i.e., telle que } b_t \in [\min_{i_c \in C} x_{i,j}, \max_{i_c \in C} x_{i,j}[}$ 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 - $\nexists c \in C$ 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 $\rightarrow$ 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.
$\Rightarrow$ toutes les données de C sont correctement classifiées
Le nombre de bonnes prédiction des données de $\mathcal{I} \backslash C$ 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 :

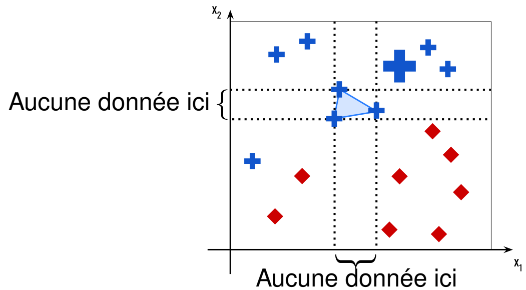
- n'était pas majoritaire dans F
- était majoritaire et
  - reste majoritaire
  - ne reste pas majoritaire

Dans ces trois cas, le nombre de bonnes prédictions de $\mathcal{I} \backslash C$ 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

Sommaire

Informations générales

Groupe

- Seul ou en binôme

Langage

- Libre
  Code Julia fourni

Calendrier

- 22/02 : ∼1h30 de TP
- 08/03 : 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) similaires à la Propriété 2
   3. Identification et utilisation d'inégalités valides intéressantes
   4. 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

$\mathcal{C} \leftarrow \{C_i = \{i\}\}_{i \in \mathcal{I}}$ **tant que** $\underline{|\mathcal{C}| \geq \gamma |\mathcal{I}|}$ **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 satisfaisant une des conditions suivantes :

- Permettre d'obtenir des clusters pouvant contenir des données de classes différentes ; ou
- Ne pas traiter indépendamment chaque classe
  Exemple : ne pas simplement appliquer un k-means pour chaque classe sans prendre en compte les données des autres classes

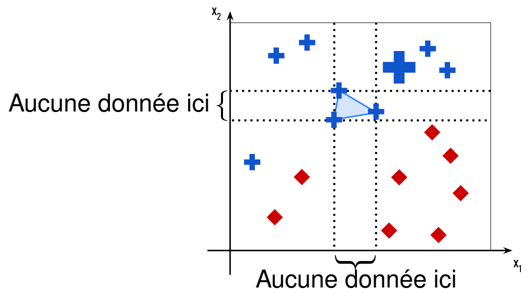
Comparer ensuite les performances à celles des méthodes fournies

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 ne soit pas trop détériorée
  Soit en le constatant expérimentalement,
  Soit en obtenant une borne sur la détérioration du nombre de bonnes prédictions



Ouverture 3 : Inégalités valides

Contexte

- La 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

  ou
- 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 4 : Autres idées

Travail demandé

Proposer et tester des idées permettant l'amélioration des performances

Sommaire

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
  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)
```


La classification supervisée en 5 minutes | Construction d'arbres de décision optimaux | Projet | Annexes

Définir des contraintes

$x_1 + x_2 = 1$

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

$y_i + x_1 \leq 1 \ \forall i\{1, \dots, n\}$

```
@constraint(m, [i = 1:n], y[i] + x1 <= 1)
```

$t_{ij} + x_1 \geq 1 \ \forall i\{1, \dots, n\} \ \forall j\{1, \dots, 4\}$

```
@constraint(m, [i = 1:n, j = 1:4], t[i, j] + x1 >= 1)
```

$\sum_{i=1}^{n} y_i \geq 3$

```
@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)
```


La classification supervisée en 5 minutes | Construction d'arbres de décision optimaux | Projet | Annexes

Obtenir la valeur d'une variable entière x1

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

Obtenir la valeur d'un tableau de variables entières `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)
```

Remarque

Le temps limite est fixé à 30 secondes dans le code fourni. N'hésitez pas à l'augmenter !

La classification supervisée en 5 minutes | Construction d'arbres de décision optimaux | Projet | Annexes

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]
```

Éxecuter ce fichier à l'ENSTA

1. Ouvrir une console : Alt + F2, puis entrer "xterm"
2. Fixer les chemins (pour les ordinateurs de l'ENSTA) : usediam ro
3. Ajouter les packages nécessaires (à ne faire qu'une fois) :
   julia
   using Pkg
   Pkg.add("JuMP")
   Pkg.add("CPLEX")
4. Éxecuter 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

```
           Lance le mode console
login@pc $ julia
julia> include("knapsack.jl")
           Exécute le fichier
```

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 l'inclusion d'un fichier

Références

Jack Dunn.
Optimal Trees for Prediction and Prescription.
PhD. Thesis, 2014.

Dimitris Bertsimas and Jack William Dunn.
Optimal classification treess.
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

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

Formulation F multivariée

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

$$
\begin{aligned}
& a_t^\intercal X_i + \mu \leq b_t + (2 + \mu)(1 - u^i_{t,l(t)}) && t \in \mathcal{N}, i \in \mathcal{I} \\
& a_t^\intercal X_i \geq b_t - 2(1 - u^i_{t,r(t)}) && t \in \mathcal{N}, i \in \mathcal{I} \\
& u^i_{t,r(t)} \leq d_t && i \in \mathcal{I}, t \in \mathcal{N} \\
& u^i_{t,w} \leq c_{k,t} && i \in \mathcal{I}^k, t \in \mathcal{N} \cup \mathcal{L} \\
& s_{j,t} \in \{0,1\} && t \in \mathcal{N}, j \in \mathcal{J} \\
& d_t \in \{0,1\} && t \in \mathcal{N} \\
& c_{k,t} \in \{0,1\} && t \in \mathcal{N} \cup \mathcal{L}, k \in \mathcal{K} \\
& u^i_e \in \{0,1\} && e \in \mathcal{E}, i \in \mathcal{I}
\end{aligned}
$$

Données de class k