BI = Business Intelligence
Master Data-Science
Cours 7 - Data Mining

Ludovic DENOYER - ludovic.denoyer@lip6.fr

UPMC

30 mars 2015

Typologie des méthodes de Data Mining

Différents types de méthodes :

Méthodes descriptives :

- Permettent de décrire un ensemble de données pour une compréhension "humaine"
- Peuvent servir de socle au développement de modèles prédictifs
- Ex : Règles d'association (voir cours précédent)

Méthodes prédictives :

- Permettent de faire de la prédiction
- En support à la prise de décision
- Ex : Réseaux de neurones

Typologies des méthodes

	Supervisées	Non spervisées
Segmentation		▪ k moyennes (K-means) ▪ k plus proches voisins (PPV-raisonnement à partir de cas) ▪ Réseaux de neurones avec cartes de Kohonen ▪ …
Classification Prédiction	▪ Arbres de décision ▪ Réseaux de neurones avec Perceptron ▪ (raisonnement à partir de cas) ▪ Modèles / réseaux bayésiens ▪ Machines à vecteur supports (SVM) ▪ programmation logique inductive ▪ …	▪ k plus proches voisins (PPV-raisonnement à partir de cas) ▪ *Règles temporelles* ▪ *Recherche de séquences* ▪ *Reconnaissance de formes* ▪ …
Prédiction	▪ Arbres de décision ▪ Réseaux de neurones avec Perceptron ▪ …	▪ k plus proches voisins (PPV-raisonnement à partir de cas) ▪ …
Association		Règles d'association

Source : Bernard Espinasse

Règles d'association (rappel)

Principe :

Extraire à partir de données transactionnelles des règles de type $A \Rightarrow B$ où A et B sont des sous-ensembles des items en présence

Algorithme APriori :

- Création des itemsets fréquents
- Création des règles à haute confiance à partir de ces itemsets
- Utilisation d'algorithme "greedy" qui utilisent la propriété de monotonie

Arbres de décision

Plan du cours :

- Arbres de décision (définition)
- Algorithme d'apprentissage
- Forêts Aléatoires
- Forêts Extrêmement aléatoires

Arbres de décision

Les arbres de décisions sont :

- Des modèles (d'apprentissage/de fouille de données) supervisés utilisés pour la régression et/ou la classification
- Ce sont des modèles **descriptifs** mais aussi des modèles **prédictifs**
- Ils produisent un décision **interprétable**, sous forme d'arbres

Arbres de décision

C'est une méthode appréciée en entreprise car :

- Elle obtient de **bons résultats** en terme de performance pure
- Le modèle appris est **lisible** par un expert. Cette caractéristique est très importante, car le travail de l'analyste consiste aussi à faire comprendre ses résultats afin d'emporter l'adhésion des décideurs.
- Elle est capable de **sélectionner** les variables à utiliser. En ce sens, elle permet un travail exploratoire sur les grands volumes de données

Arbres de décision

According to research in cognitive psychology (Miller 1956 ; Kahneman, Slovic, and Tversky 1982) the ability to conceptually grasp and manipulate multiple chunks of knowledge is limited by the physical and cognitive processing limitations of the shortterm memory portion of the brain. This places a premium on the utilization of dimensional manipulation and presentation techniques that are capable of preserving and reflecting high-dimensionality relationships in a readily comprehensible form so that the relationships can be more easily consumed and applied by humans.

Retour sur les règles d'association

Règles d'association

Les règles d'association expriment une notion de lien entre objets de même type (ex : les produits vendus par un magasin).
Attention : Une règle d'association n'exprime pas une corrélation

Règles de production

Les règles de production expliquent le lien entre une classe particulière et la valeur des caractéristiques de plusieurs objets

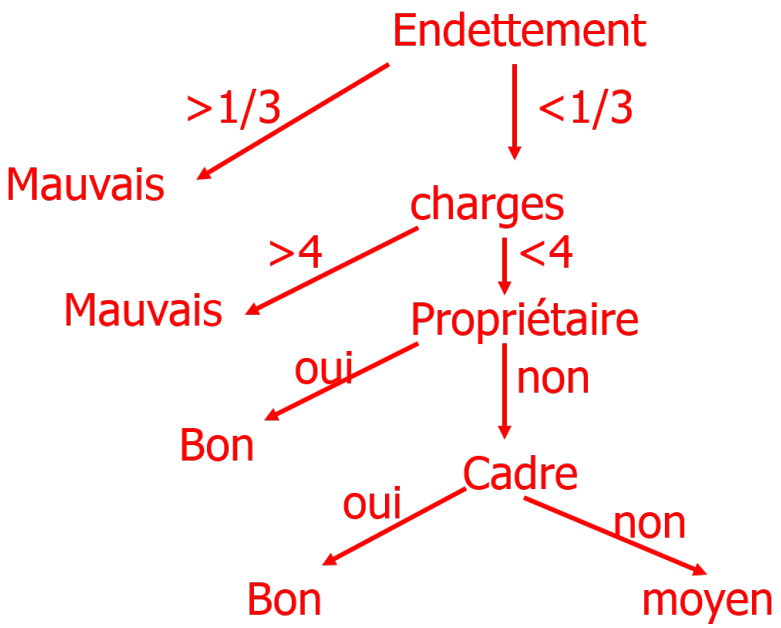
Les deux types de règles ne véhiculent pas le même type d'information.

Règles de production : Exemples

- Si crédit $> 1/3$ salaire $\Rightarrow$ mauvais
- Si crédit $< 1/3$ salaire ET charges $> 4 \Rightarrow$ mauvais
- Si crédit $< 1/3$ salaire ET charges < 4 ET propriétaire $=$ oui $\Rightarrow$ bon
- Si crédit $< 1/3$ salaire ET charges < 4 ET propriétaire=non ET cadre=oui $\Rightarrow$ bon
- Si crédit $< 1/3$ salaire ET charges < 4 ET propriétaire=non ET cadre $=$ non $\Rightarrow$ moyen

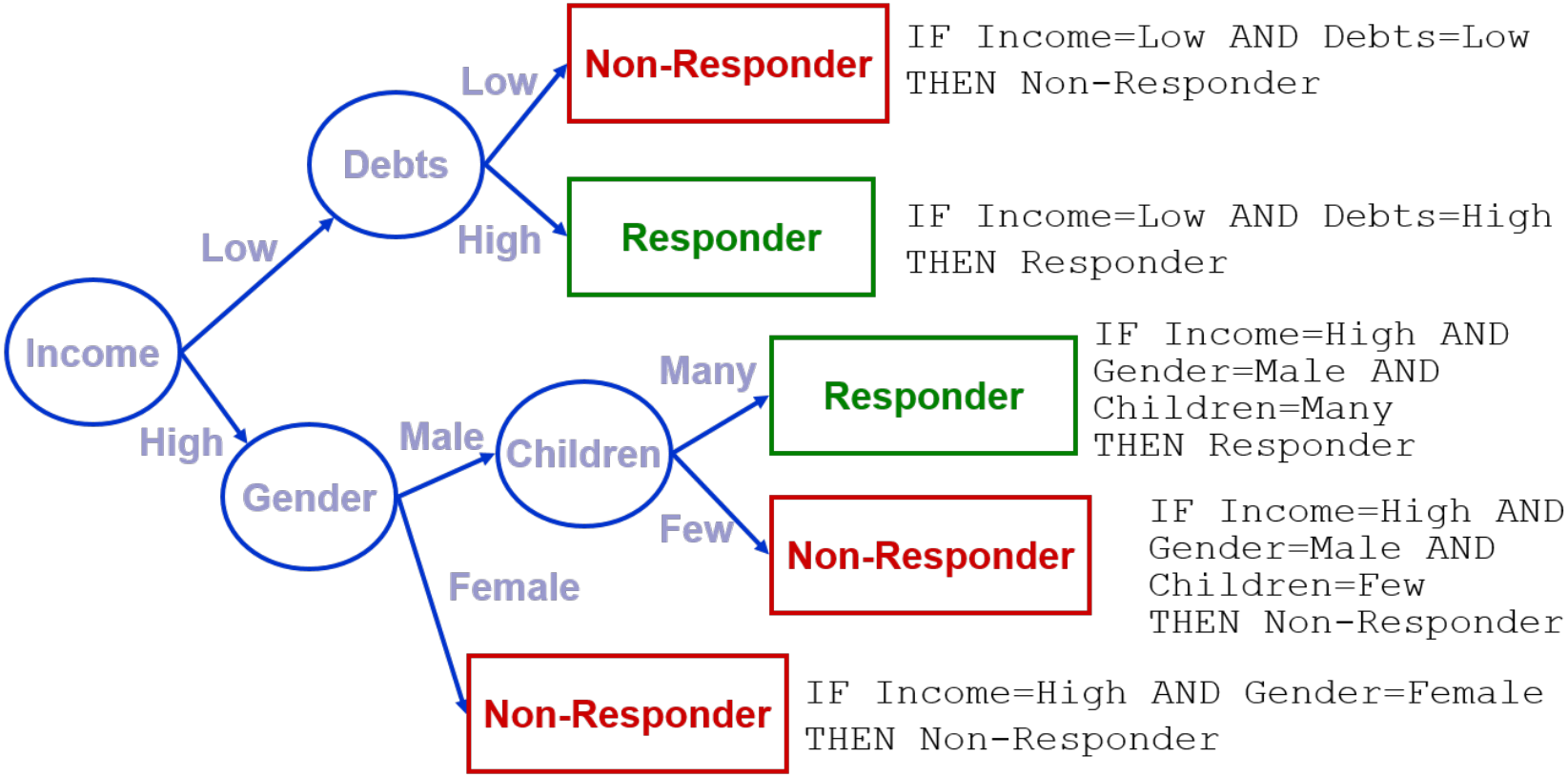
Arbres de décision

Les règles de production peuvent être regroupées sous la forme d'un **arbre de décision**

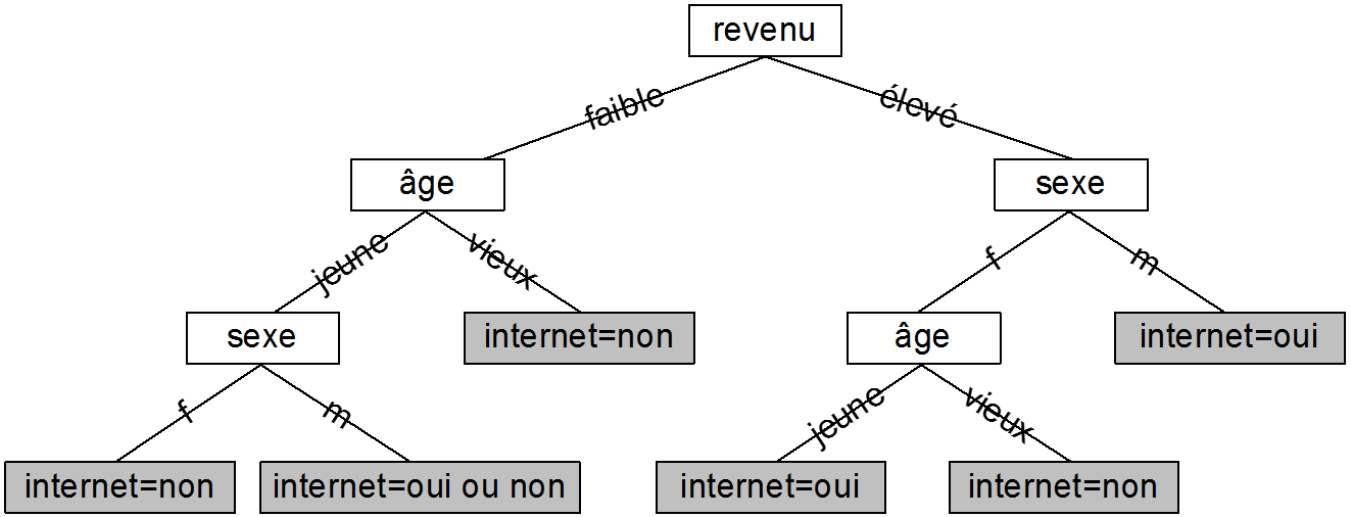


En pratique, les systèmes construisent d'abord les arbres d'où ils dérivent les règles

Arbres de décision



Arbres de décision



Exercice

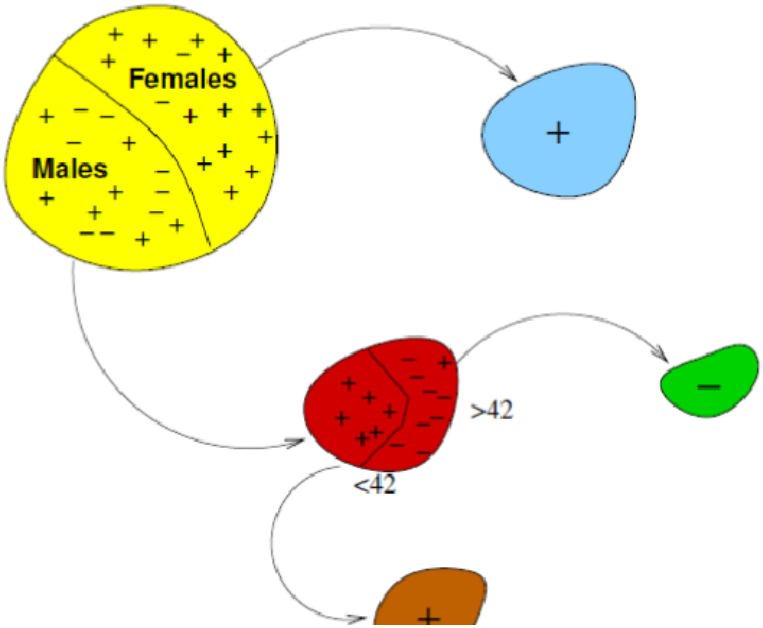
D

onner les arbres de décisions qui expriment les fonctions booléennes suivantes :

- A et B
- A ou (B et C)
- A XOR B
- (A et B) ou (C et D)

Algorithme

- L'algorithme d'inférence des arbres de décision est un algorithme de type *Divide and Conquer*
- Au début, tous les exemples sont à la racine de l'arbre
- Partitionnement récursif des exemples

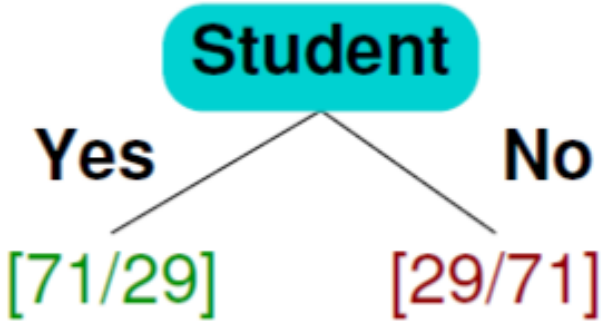


Algorithme

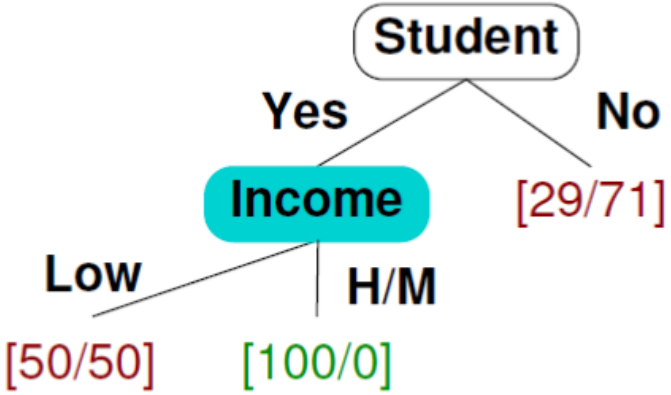
Quelles règles trouvez vous pour savoir qui achète un ordinateur ?

Age	Income	Student	Credit	Buys
< 30	High	No	Poor	No
< 30	High	No	Good	Yes
$30 - 40$	High	No	Poor	Yes
> 40	Medium	No	Poor	Yes
> 40	Low	Yes	Poor	Yes
> 40	Low	Yes	Good	No
$30 - 40$	Low	Yes	Good	Yes
< 30	Medium	No	Poor	No
< 30	Low	Yes	Poor	No
> 40	Medium	Yes	Poor	Yes
< 30	Medium	Yes	Good	Yes
$30 - 40$	Medium	No	Good	Yes
$30 - 40$	High	Yes	Poor	Yes
> 40	Medium	No	Good	No

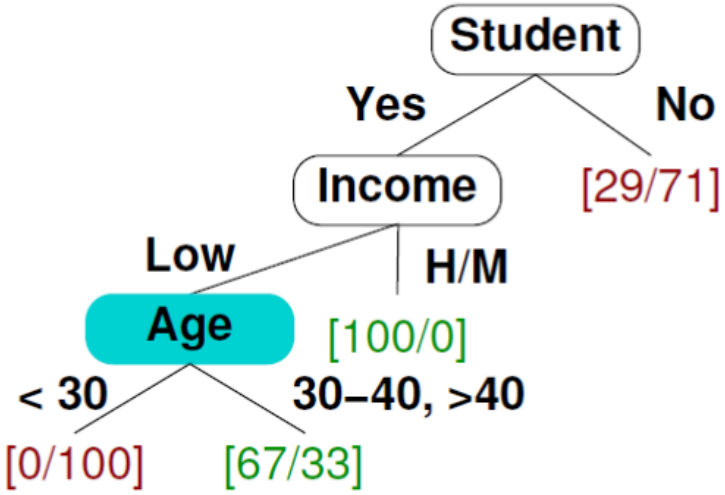
Algorithme



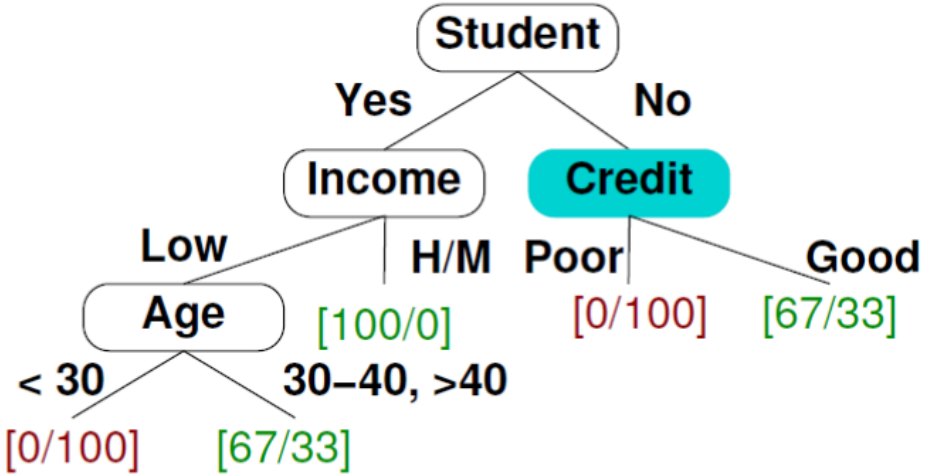
Algorithme



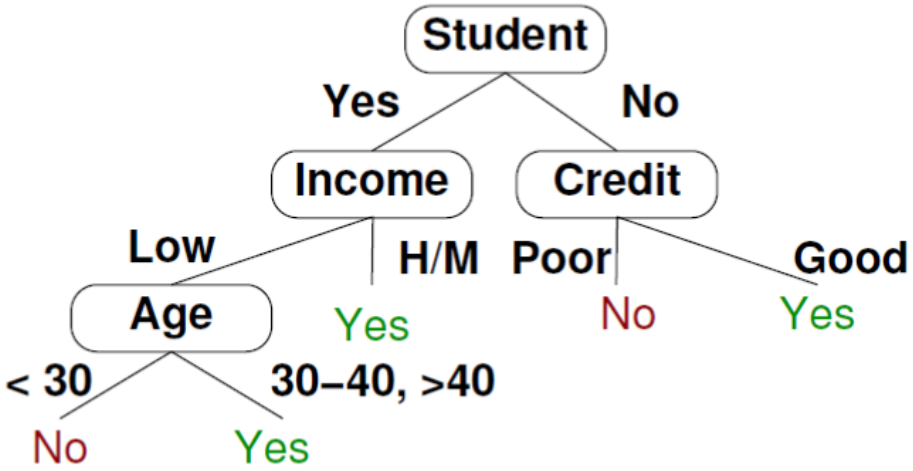
Algorithme


Student
Yes
No
Income
[29/71]
Low
H/M
Age
[100/0]
< 30
30–40, >40
[0/100]
[67/33]

Algorithme


Student
Yes
No
Income
Credit
Low
H/M
Poor
Good
Age
[100/0]
[0/100]
[67/33]
< 30
30–40, >40
[0/100]
[67/33]

Algorithme


Student
Yes
No
Income
Credit
Low
H/M
Poor
Good
Age
Yes
No
Yes
< 30
30–40, >40
No
Yes

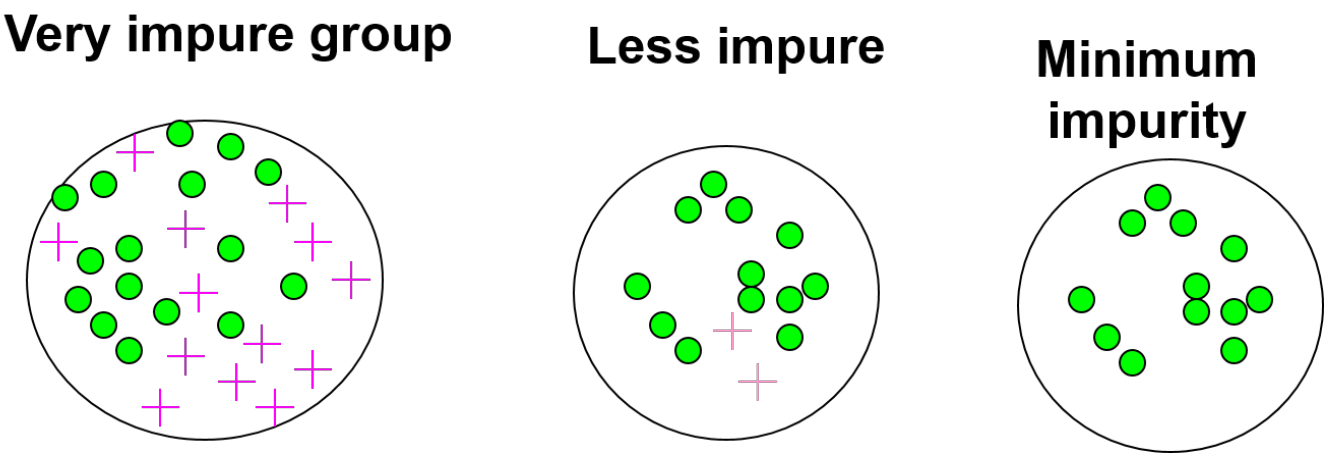
Algorithme

- Algorithme *greedy* qui prend la "meilleure" décision à chaque pas de temps
- L'arbre est construit "de haut en bas" (top-down) de manière récursive (divide and conquer)
- Tous les exemples d'apprentissage sont à la racine au début
- A chaque étape, les données sont partitionnées suivant la valeur **d'une variable**
- La variable est sélectionnée par une **mesure**
- L'algorithme s'arrête quand :
  - Toutes les données d'un noeud sont dans la même catégorie
  - Il n'y a plus de variables disponibles pour continuer le partitionnement
  - Il n'y a pas de données dans un noeud

Critère de partitionnement

Le critère usuellement de partitionnement est l'entropie :

- Un ensemble aléatoire de données possède une entropie forte
- Un ensemble de basse entropie est un ensemble où la distribution de la variable de sortie possède un unique pic



Mesure de discrimination classique

- Utilisation de l'entropie de Shannon :

$$H_S(C|A) = -\sum_i P(v_i) \sum_k P(c_k|v_i) \log(P(c_k|v_i))$$

- Mesure issue de la théorie de l'information
  - initiée par C.E. Shannon en 1948
- Mesure un **taux de désordre**

Construction de l'arbre : cas général

Algorithme d'apprentissage

1. Calculer $H(C|A_j)$ pour tous les attributs A_j
2. Choisir l'attribut A_j qui minimise $H(C|A_j)$
   - créer un **nœud** dans l'arbre de décision
3. A l'aide de A_j, **partitionner** la base d'apprentissage

Exemple

	Devoirs finis	Maman de bonne humeur	Temps = beau	Goûter pris	Décision
1	Vrai	Faux	Vrai	Faux	OUI
2	Faux	Vrai	Faux	Vrai	OUI
3	Vrai	Vrai	Vrai	Faux	OUI
4	Vrai	Faux	Vrai	Vrai	OUI
5	Faux	Vrai	Vrai	Vrai	NON
6	Faux	Vrai	Faux	Faux	NON
7	Vrai	Faux	Faux	Vrai	NON
8	Vrai	Vrai	Faux	Faux	NON

Exemple

	DF	BH	TB	GP	Décision
1	Vrai	Faux	Vrai	Faux	OUI
2	Faux	Vrai	Faux	Vrai	OUI
3	Vrai	Vrai	Vrai	Faux	OUI
4	Vrai	Faux	Vrai	Vrai	OUI
5	Faux	Vrai	Vrai	Vrai	NON
6	Faux	Vrai	Faux	Faux	NON
7	Vrai	Faux	Faux	Vrai	NON
8	Vrai	Vrai	Faux	Faux	NON

- Calcul de $H(c|DF)$, $H(c|BH)$, $H(c|TB)$ et $H(c|GP)$
- $H(c|DF) = \left(\frac{5}{8}\right) J(DF = Vrai) + \left(\frac{3}{8}\right) J(DF = Faux)$
- $J(DF = Vrai) = -\left(\frac{3}{5}\right) log_2 \left(\frac{3}{5}\right) - \left(\frac{2}{5}\right) log_2 \left(\frac{2}{5}\right)$
  $J(DF = Faux) = -\left(\frac{1}{3}\right) log_2 \left(\frac{1}{3}\right) - \left(\frac{2}{3}\right) log_2 \left(\frac{2}{3}\right)$
- $H(c|DF) = 0.95$, $H(c|BH) = 0.93$, $H(c|TB) = 0.8$ et $H(c|GP) = 1$

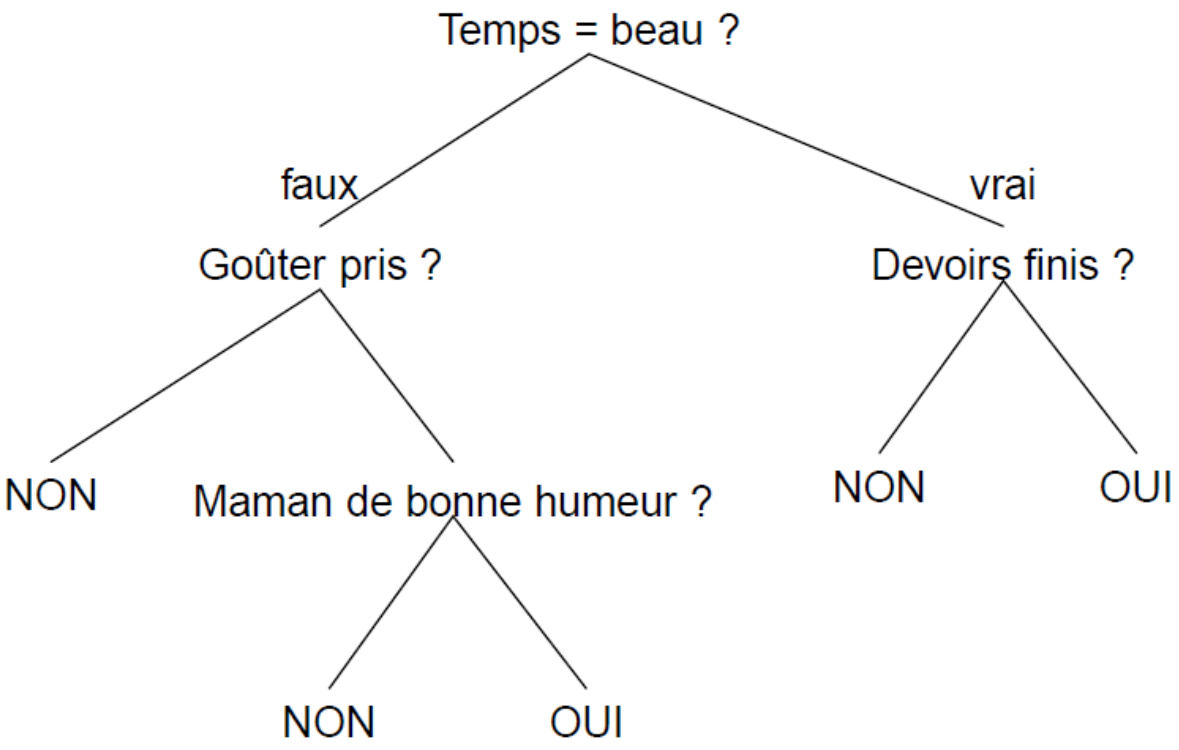
Exemple

Temps = beau ?

	DF	BH	GP	Décision
2	F	V	V	OUI
6	F	V	F	NON
7	V	F	V	NON
8	V	V	F	NON

	DF	BH	GP	Décision
1	V	F	F	OUI
3	V	V	F	OUI
4	V	F	V	OUI
5	F	V	V	NON

Exemple


Temps = beau ?
faux
vrai
Goûter pris ?
Devoirs finis ?
NON
Maman de bonne humeur ?
NON
OUI
NON
OUI

Exercice

Faire l'arbre de décision la dessus :

Num. exemple	Att 1	Att 2	Att 3	Classe
1	0	0	0	+
2	0	0	0	+
3	0	1	0	+
4	1	1	1	+
5	1	0	1	−
6	1	0	0	−

Exercice

- attribut 1 :
  $H(C|A1) = -\frac{3}{6}\left(\frac{3}{3}\log\frac{3}{3} + \frac{0}{3}\log\frac{0}{3}\right) - \frac{3}{6}\left(\frac{1}{3}\log\frac{1}{3} + \frac{2}{3}\log\frac{2}{3}\right)$
  ce qui donne $H(C|A1) =$
  $-0.5 * (0 + 0) - 0.5 * (0.33 * -1.58 + 0.66 * -0.58) = 0.459$
- attribut 2 :
  $H(C|A2) = -\frac{4}{6}\left(\frac{2}{4}\log\frac{2}{4} + \frac{2}{4}\log\frac{2}{4}\right) - \frac{2}{6}\left(\frac{2}{2}\log\frac{2}{2} + \frac{0}{2}\log\frac{0}{2}\right)$
  ce qui donne
  $H(C|A2) = -0.66 * (-0.5 - 0.5) - 0.33 * (0 + 0) = 0.666$
- attribut 3 :
  $H(C|A3) = -\frac{4}{6}\left(\frac{3}{4}\log\frac{3}{4} + \frac{1}{4}\log\frac{1}{4}\right) - \frac{2}{6}\left(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right)$
  ce qui donne
  $H(C|A2) = -0.66*(-0.311-0.5)-0.33*(-0.5-0.5) = 0.873$

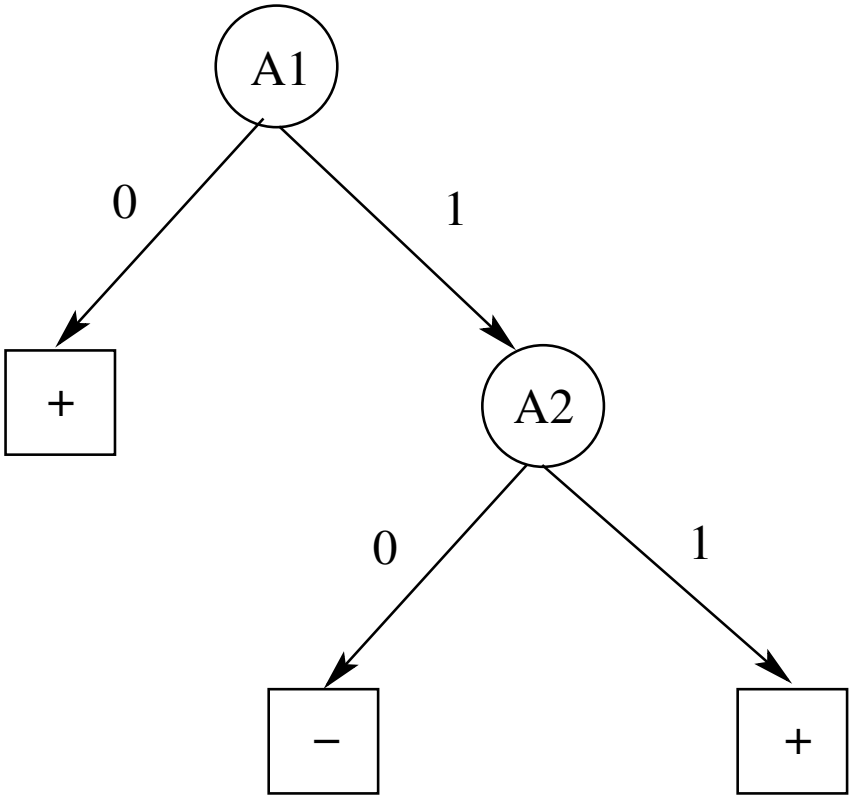
Exercice

L'attribut 1 est celui qui minimise l'entropie et qui est donc choisi comme racine.
Ensuite, on sépare la base d'apprentissage en 2 selon les valeurs de A1. Les exemples qui ont la valeur 0 sont de la même classe, on arrête donc la construction de la branche de l'arbre pour eux en créant une feuille dont la classe associée est $+$.
Pour ceux qui ont la valeur 1 pour A1, on calcule les entropies de A2 et de A3 :

- attribut 2 :
  $H(C|A2) = -\frac{2}{3}\left(\frac{0}{2}\log\frac{0}{2} + \frac{2}{2}\log\frac{2}{2}\right) - \frac{1}{3}\left(\frac{1}{1}\log\frac{1}{1} + \frac{0}{1}\log\frac{0}{1}\right)$
  ce qui donne $H(C|A2) = -0.66 * (0+0) - 0.33 * (0+0) = 0.0$
- attribut 3 :
  $H(C|A3) = -\frac{1}{3}\left(\frac{0}{1}\log\frac{0}{1} + \frac{1}{1}\log\frac{1}{1}\right) - \frac{2}{3}\left(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right)$
  ce qui donne
  $H(C|A3) = -0.33 * (0+0) - 0.66 * (-0.5 - 0.5) = 0.666$

Exercice

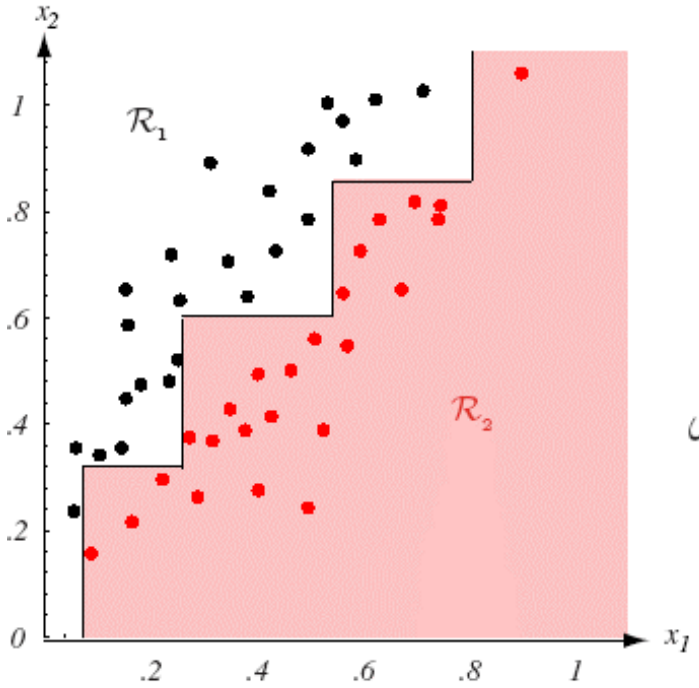
C'est donc l'attribut A2 qui minimise l'entropie ici et qui est utilisé dans l'arbre.
L'arbre final est donné dans la figure suivante :

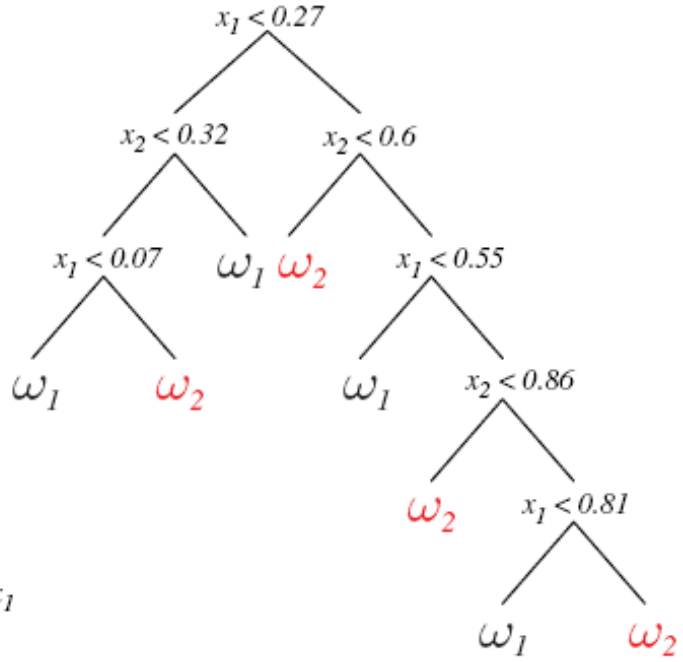


Autres algorithmes

- **Algorithme C5** [Quinlan 93] :
  - version plus récente de l'algorithme ID3 [Quinlan 79]
  - peut prendre en compte des attributs d'arité quelconque.
  - fonction entropie, mesurant le degré de mélange (et le gain), a privilégier les attributs possédant un grand nombre de valeurs biais, une fonction gain d'information est également disponible
  - élagage effectué avec l'ensemble d'apprentissage par une éva pessimiste de l'erreur.
- **Algorithme CART** [Breiman et al., 84] :
  - intégré à de nombreux environnements de fouille de données nombreuses variantes
  - fonction mesurant le degré de mélange (et le gain) = fonction versions diffusées proposent d'autres choix).
  - élagage par parcours ascendant de l'arbre : un sous arbre peu

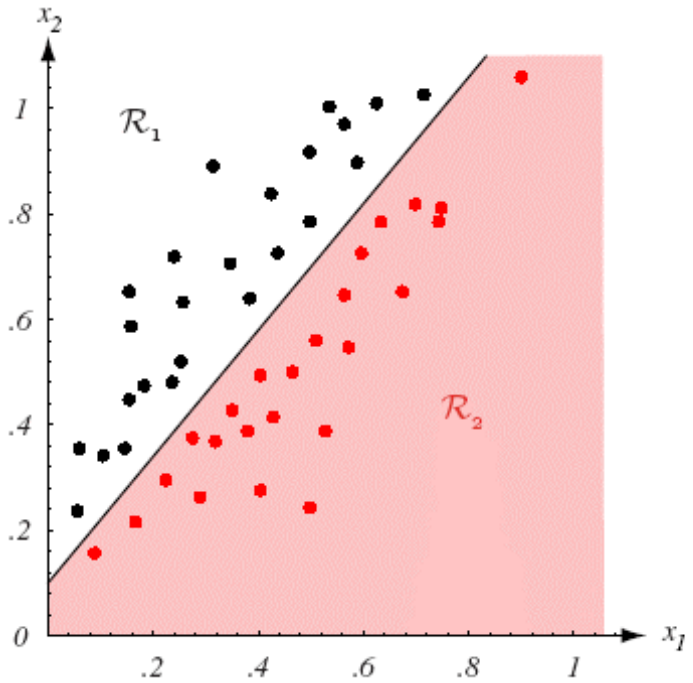
DT sur données continues

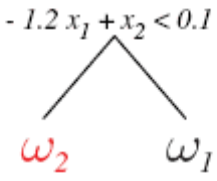




Le choix des variables à utiliser est très important $\Rightarrow$ Algorithmes de discrétisation des variables

DT sur données continues





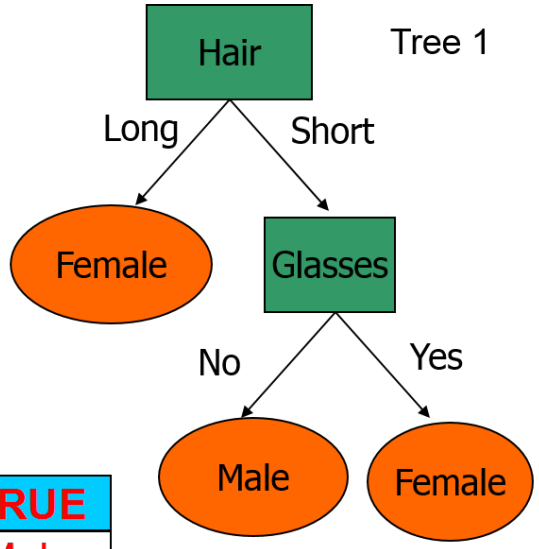
Sur-apprentissage

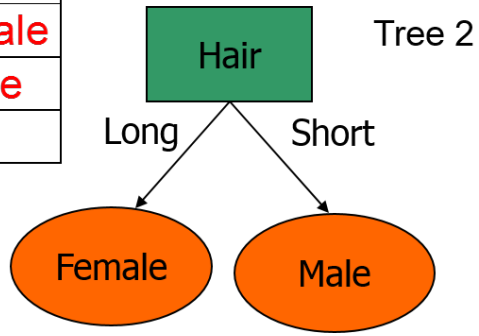
Training

Name	Hair	Glasses	Class
Mary	Long	No	Female
Mike	Short	No	Male
Bill	Short	No	Male
Jane	Long	No	Female
Ann	Short	Yes	Female

Testing

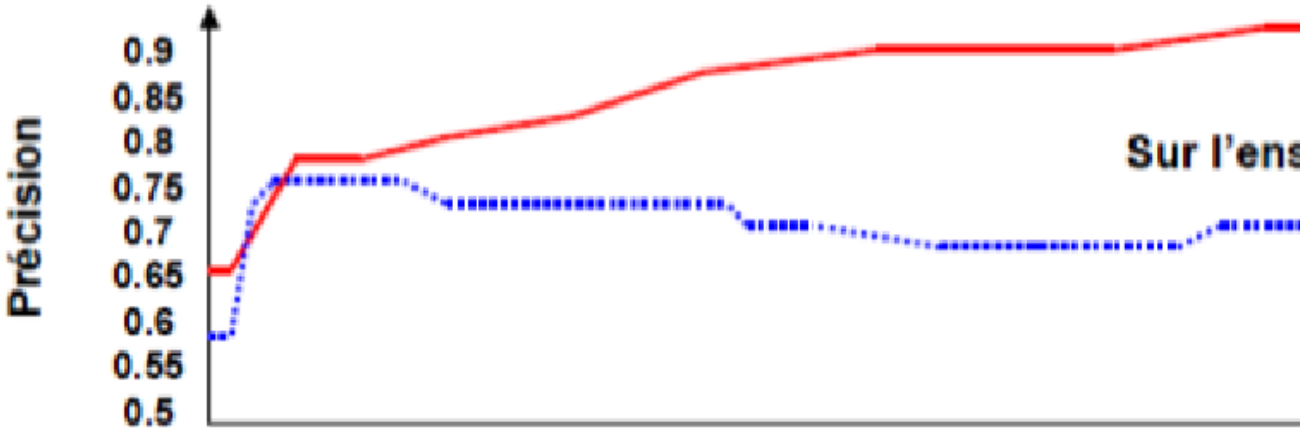
Hair	Glasses	Tree 1	Tree 2	TRUE
Short	Yes	Female	Male	Male
Short	No	Male	Male	Female
Long	No	Female	Female	Female
Short	Yes	Female	Male	Male
	Error:	75%	25%	





Sur-apprentissage

Dans le cas extrême, on peut apprendre une branche par exemple d'apprentissage



Pré-élagage : Arréter les développement d'une branche avant que tous les exemples de cette branche soient bien classés

Post-élagage : Apprendre un arbre en entier, élaguer par la suite

Sur-Apprentissage

Elagage à posteriori

Idée : Elaguer après la construction de l'arbre entier, en remplaçant les sous-arbres optimisant un critère d'élagage par un noeud. Nombreuses méthodes. Encore beaucoup de recherches.

- Minimal Cost-Complexity Pruning (MCCP) (Breiman et al.,84)
- Reduced Error Pruning (REP) (Quinlan,87,93)
- Minimum Error Pruning (MEP) (Niblett & Bratko,86)
- Critical Value Pruning (CVP) (Mingers,87)
- Pessimistic Error Pruning (PEP) (Quinlan,87)
- Error-Based Pruning (EBP) (Quinlan,93) (utilisé dans C4.5)
- ...

Forêts aléatoires

- Méthode introduite par Leo Breiman en 2001.
- Idées plus anciennes :
  - Bagging (1996)
  - Arbres de décisions CART (1984)
- Preuves de convergences récentes (2006,2008)
- Un site web utile :
  http://www.stat.berkeley.edu/~breiman/RandomForests

Idée

Les forêts aléatoires consistent à faire tourner en parallèle un grand nombre d'arbres de décisions construits aléatoirement, avant de les moyenner

Bagging d'arbres

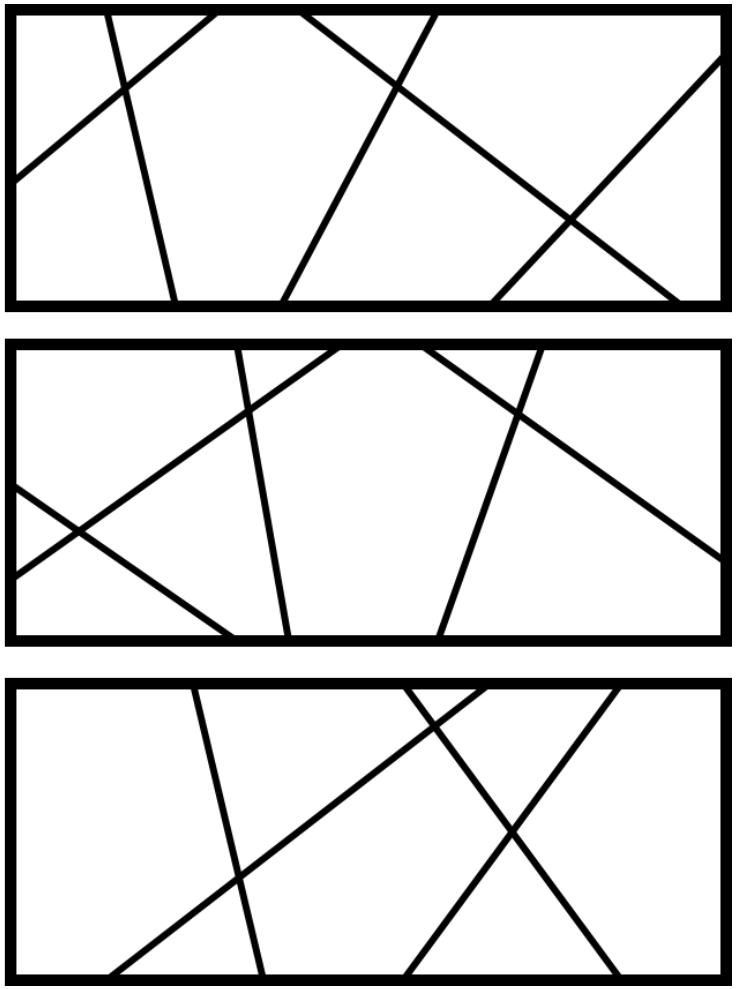
Soit un ensemble $X = x_1, ...x_n$ d'apprentissage, et les étiquettes associées $Y = y_1, ..., y_n$. Le principe consiste à :

- Pour $b = 1, ...B$:
  - Tirer avec replacement n exemples d'apprentissage de $X, Y \Rightarrow X_b, Y_b$
  - Apprendre un arbre de décision f_b sur X_b, Y_b
- Le classifieur final est : $f(x) = \frac{1}{B} \sum_{b=1}^{B} f_b(x)$ ou bien un vote majoritaire dans le cas catégoriel.

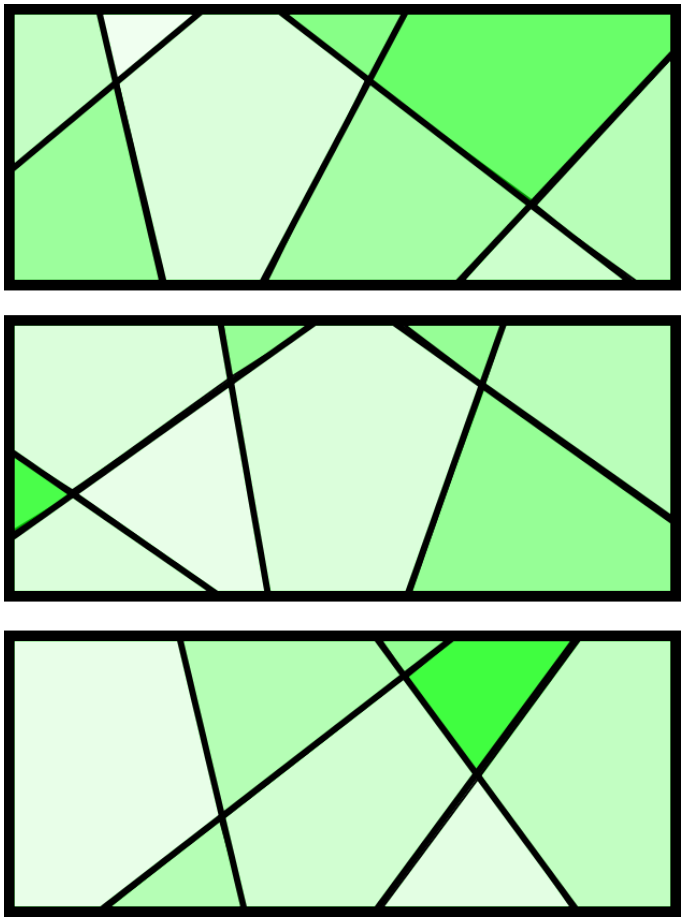
Ce type d'approche fait baisser la variance du modèle, sans augmenter son biais. Le nombre d'arbres est souvent sélectionné par cross-validation.

Comment ça marche ?

Les différents
Arbres sont autant
De partitions de
L'espace d'entrée

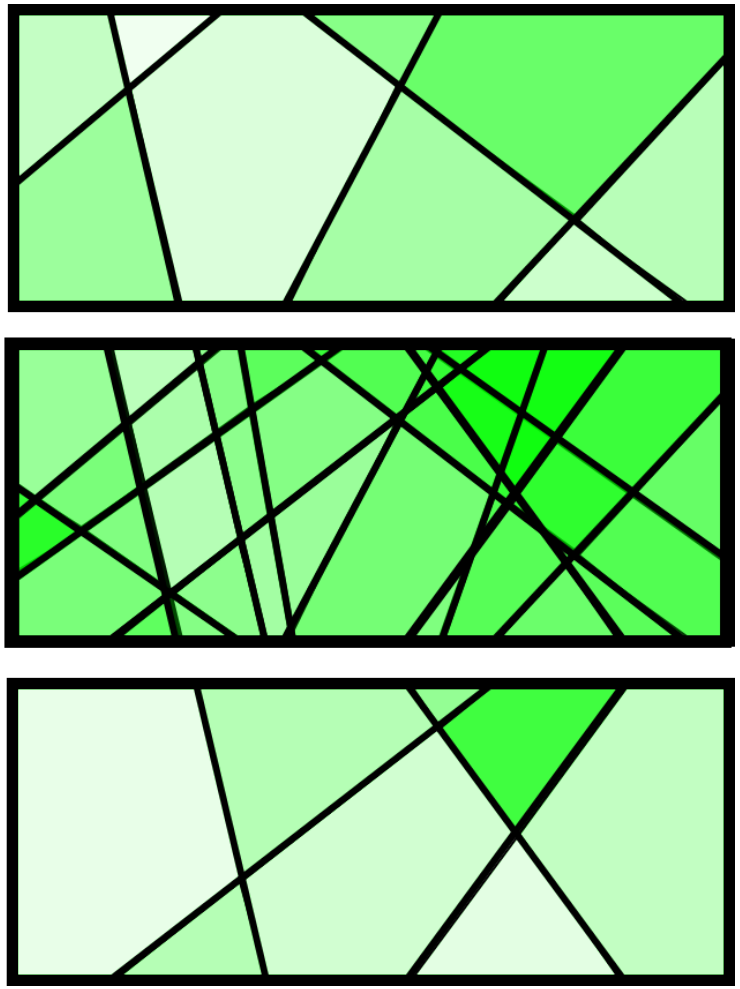


Comment ça marche ?



Comment ça marche ?

La combinaison
Permet une divi-
-sion plus fine de
L'espace



Random Forests

Les *random forests* sont basées sur le même principe, avec une modification dans l'algorithme d'apprentissage des arbres de décision :

- A chaque noeud de l'arbre, un sous-ensemble d'attributs est sélectionné aléatoirement, le découpage s'effectuant sur ce sous-ensemble d'attributs

Algorithme

- Pour $b = 1, ...B$:
  - Tirer avec replacement n exemples d'apprentissage de $X, Y \Rightarrow X_b, Y_b$
  - Apprendre un arbre de décision f_b sur X_b, Y_b, avec **randomisation** des variables
- Le classifieur final est : $f(x) = \frac{1}{B} \sum_{b=1}^{B} f_b(x)$ ou bien un vote majoritaire dans le cas catégoriel.

Forêts extrêmement aléatoires

Dans le cas précédent :

- La sélection du split se fait sur un critère de maximisation de gain d'information, basé sur un sous-ensemble de candidats

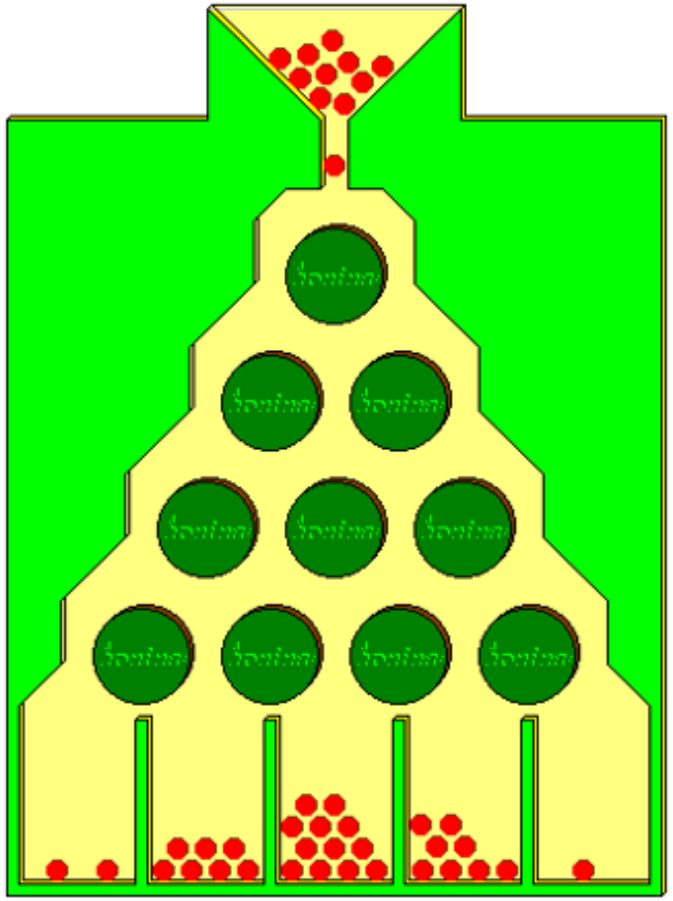
Pourquoi ne pas faire encore plus simple :

- L'effort principal provien du calcul du gain d'information
- On pourrait choisir un split aléatoire (sur un attribut aléatoire) dans chaque noeud

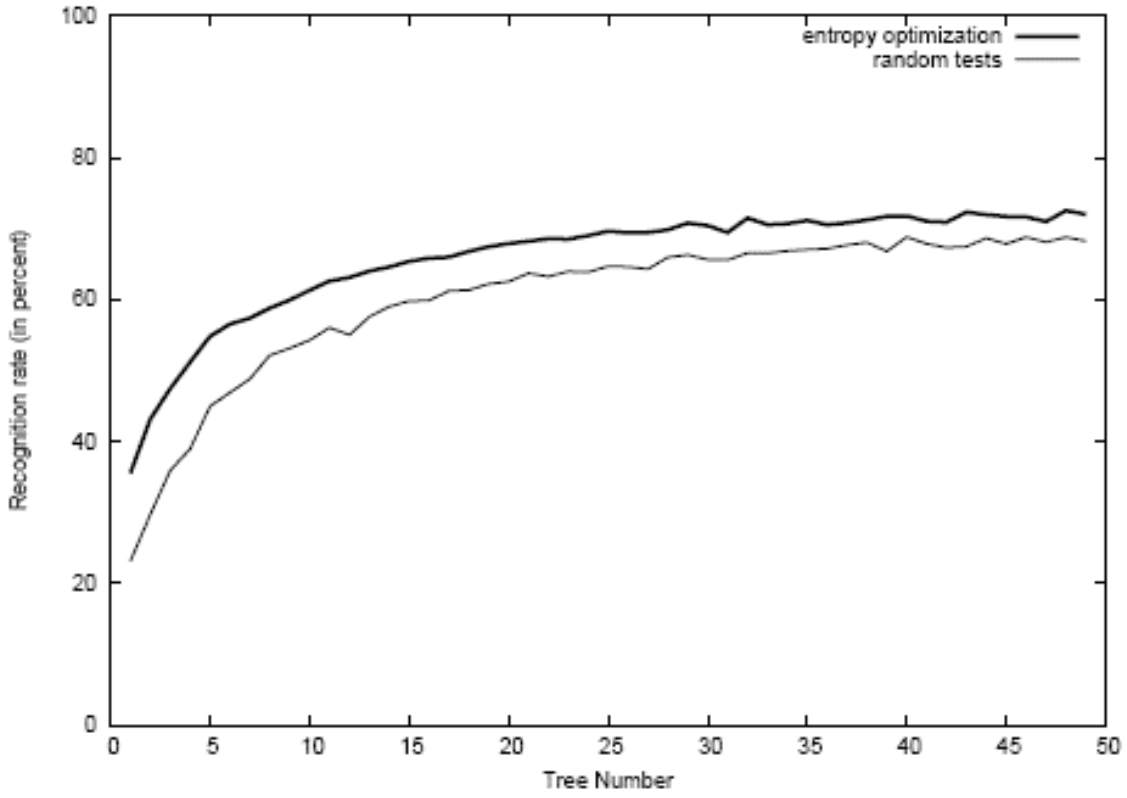
$\Rightarrow$ arbre aléatoire

Arbre extrèmement aléatoire

- Random query pour chaque noeud de l'arbre
- Les feuilles contiennent la distribution a posteriori des labels



Arbre extrèmement aléatoire



Les forêts extrèmement aléatoires donnent des performances quasi similaires aux forêts aléatoires, mais au prix d'un apprentissage beaucoup moins coûteux

Arbres de décision

- Modèles d'apprentissage de règles de production
- Sur tous types de données
- Permet de sélectionner des variables discriminantes : il est donc utile aussi comme algorithme de prétraitement
- Performant
- Beaucoup d'outils disponibles
- Par contre :
  - Les performances se dégradent quand le nombre de classes est élevé
  - L'algorithme n'est pas incrémental et il doit être re-appris à chaque fois que les données évoluent