Apprentissage, réseaux de neurones et modèles graphiques (RCP209)

Arbres de décision

Marin FERECATU
(prenom.nom@cnam.fr)
http ://cedric.cnam.fr/vertigo/Cours/ml2/

Département Informatique
Conservatoire National des Arts & Métiers, Paris, France

Plan du cours

Objectif

"La raison d'être des statistiques, c'est de vous donner raison." — Abe Burrows

- Arbres de décision : motivation, définition, exemples
- Apprentissage avec de arbres de décision : classification, régression
- Implémentation
  - ID3, C4.5, C5.0
  - CART
- Extensions
  - Graphes de décision
  - Bagging decision trees, Boosted trees
  - Forets aléatoires (random forrests) — prochaine séance

Plan du cours

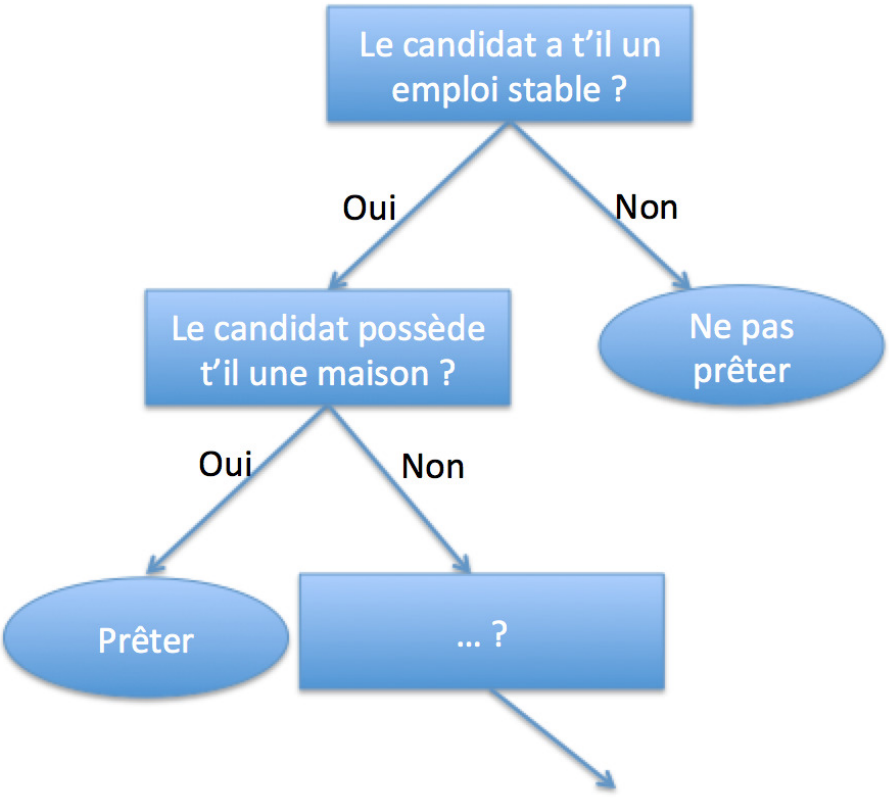
Arbres de décision (AD)

Arbres de décision :

- Outil utilisé dans l'exploration de données et informatique décisionnelle.
- Représentation hiérarchique de la structure des données sous forme des séquences de décision (tests) en vue de la prédiction d'un résultat ou d'une classe.

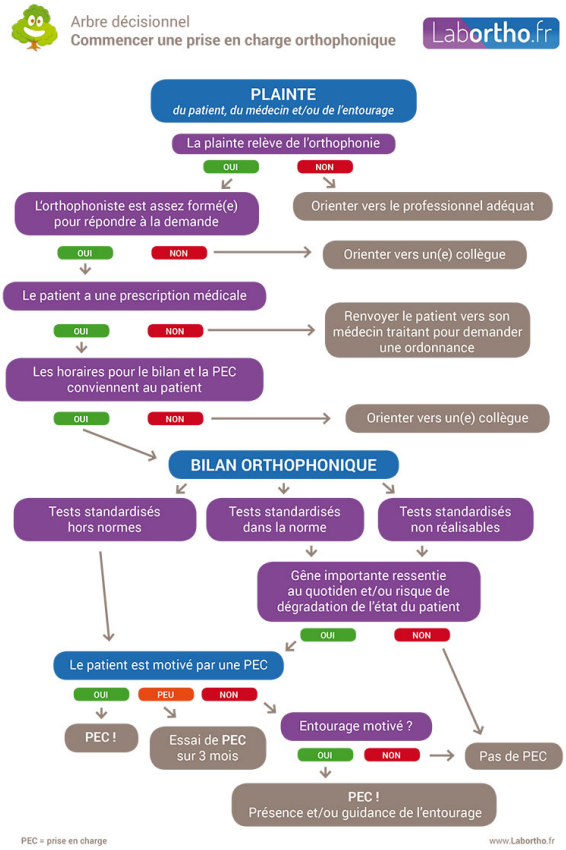
Problème à résoudre : comment répartir une population d'individus (e.g. clients, produit, utilisateurs etc.) en groupes homogènes selon un ensemble de variables discriminantes (e.g. âge, temps passé sur un site Web, etc.) et en fonction d'un objectif fixé (variable de sortie ; par exemple : chiffre d'affaires, probabilité de cliquer sur une publicité, etc.)

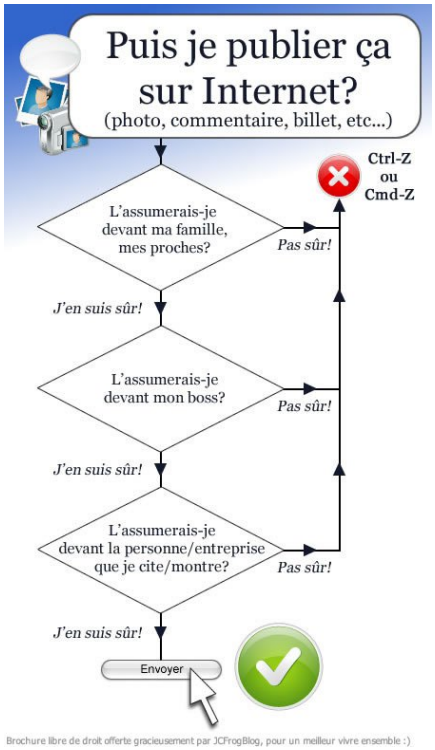
Arbres de décision : exemples



Source : https ://maximilienandile.github.io

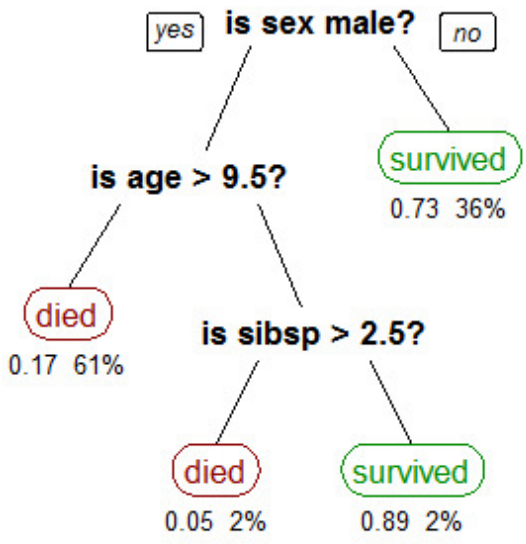
Arbres de décision : exemples





Source : http ://www.labortho.fr, https ://jeromechoain.files.wordpress.com

Arbres de décision : exemples



Survie de passagers sur le Titanic (https ://en.wikipedia.org).

Plan du cours

Apprentissage avec arbres de décision

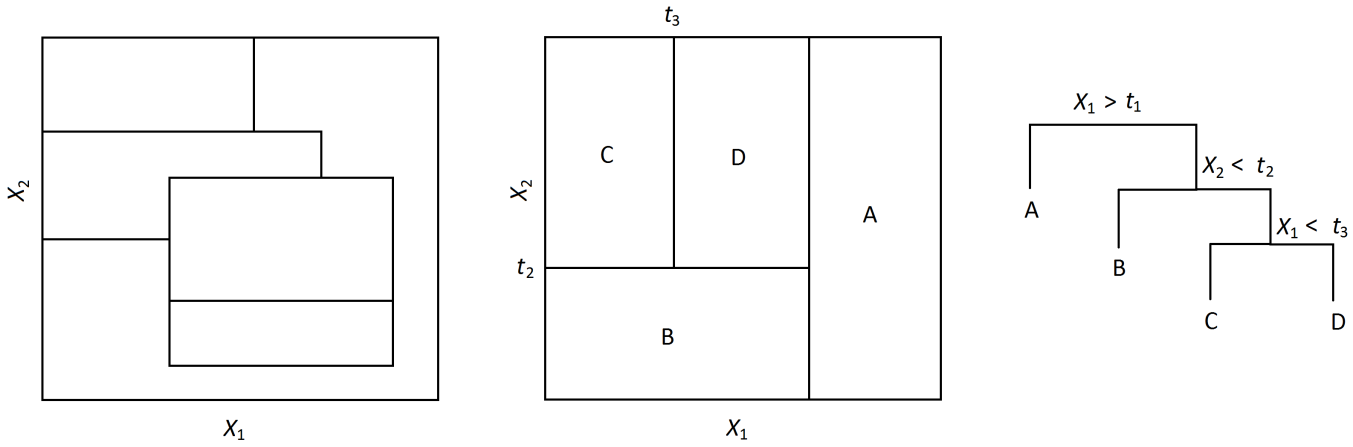
Représentation :

- Chaque nœud interne correspond à un attribut
- Chaque nœud teste l'attribut correspondant et génère plusieurs branches
  - Variable catégorielle : une branche par valeur de l'attribut
  - Variable numérique : test sur valeur
- Les *feuilles* spécifient les classes

Principe de la construction :

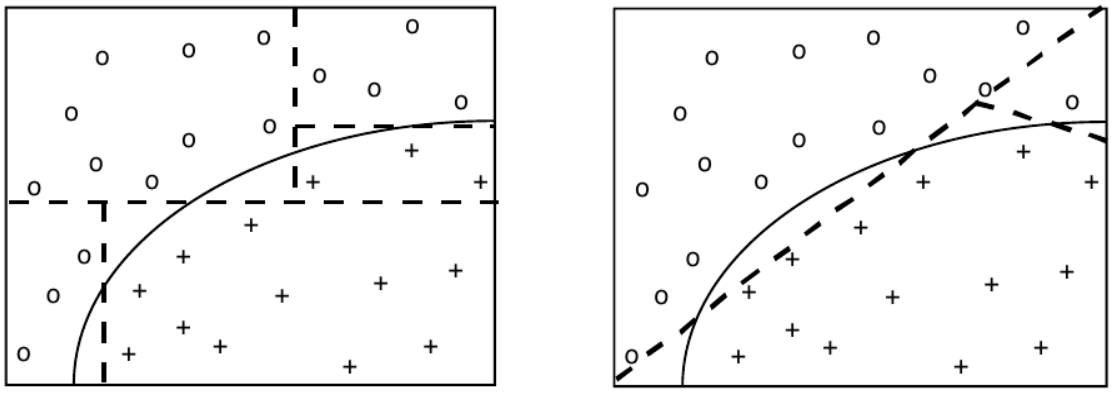
- L'arbre est construit par partition récursive de la base d'apprentissage en fonction de la valeur de l'attribut testé à chaque itération (top-down induction).
- Le processus s'arrête quand les éléments d'un nœud ont la même valeur pour la variable cible (homogénéité).

Apprentissage avec arbres de décision



Gauche : division de l'espace impossible à obtenir par partition récursive sur les attributs.
Milieu et droite : Partition récursive de l'espace et arbre obtenu. (source : wikimedia.org)

Apprentissage avec arbres de décision

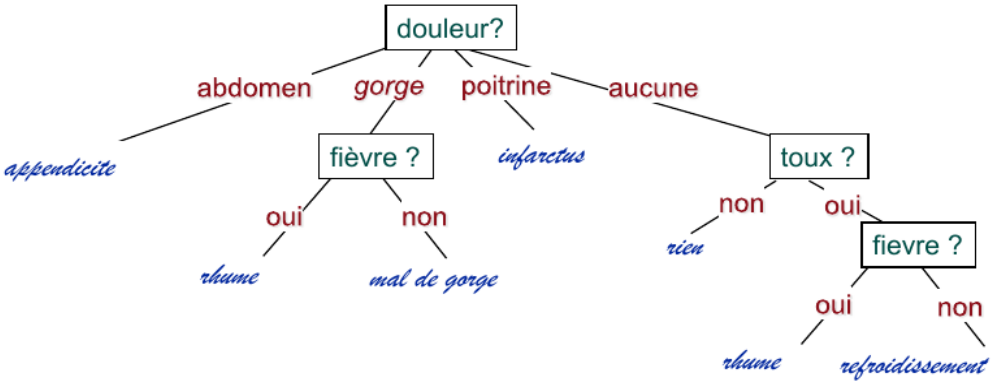


Gauche : séparation de classes par partition itérative des variables.
Droite : séparation par combinaison linéaire de plusieurs variables.

Apprentissage avec arbres de décision

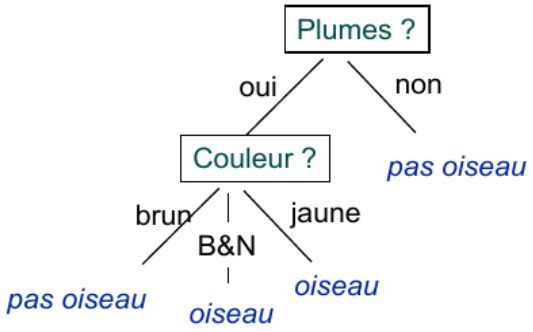
- Données d'entrée : points dans un "feature space" spécifié par ses attributs
  - variables catégorielles ou numériques
- Cible : classe (classification) ou valeur (régression)

	Toux	Fièvre	Poids	Douleur
Marie	non	oui	normal	gorge
Fred	non	oui	normal	abdomen
Julie	oui	oui	maigre	aucune
Elvis	oui	non	obese	poitrine



Apprentissage avec arbres de décision

	Couleur	Ailes	Plumes	Sonar	**Concept**
Faucon	jaune	oui	oui	non	*oiseau*
Pigeon	B&N	oui	oui	non	*oiseau*
chauve-souris	brun	oui	non	oui	*pas oiseau*



(Si Plumes = « non » Alors Classe= « pas-oiseau)
ou (Si Plumes = oui & Couleur= brun Alors Classe= pas-oiseau)
ou (Si Plumes = oui & Couleu r= B&N Alors Classe= oiseau)
ou (Si Plumes = oui & Couleu r= jaune Alors Classe= oiseau)

Plan du cours

ID3 (Iterative Dichotomiser 3)

Quinlan, J. R., Induction of Decision Trees. Mach. Learn. 1, (Mar. 1986), pp. 81-106

S un nœud interne :

- Partitionner S sur les valeurs de la cible en n groupes : $C_1, \ldots, C_m$
- p_i : probabilité qu'un élément de S se retrouve dans C_i ($p_i \approx |C_i|/|S|$)
- $H(S) = -\sum_{1=1}^{m} p_i log(p_i)$ entropie de S
- $H(S) = 0$ si S est homogène (tous les éléments sont dans la même classe : un $p_i = 1$, le reste à 0)
- $H(S) = max$ si tous les groupes C_i ont la même taille ($p_1 = \cdots = p_n = 1/n$)

ID3 (Iterative Dichotomiser 3)

Quinlan, J. R., Induction of Decision Trees. Mach. Learn. 1, (Mar. 1986), pp. 81-106

S un nœud interne :

- Partitionner S sur les valeurs de l'attribut a en n sous-groupes : $S_1, \ldots, S_n$
- p_i : la probabilité qu'un élément de S appartient à S_i ($p_i \approx |S_i|/|S|$)
- $GI(S;a) = H(S) - \sum_{1=1}^{n} p_i H(S_i)$ le gain d'information sur l'attribut a

Algorithme :

- Calculer l'entropie de chaque attribut pas encore utilisé
- Choisir l'attribut de gain d'information maximal
- Créer un nœud test (décision) sur cet attribut et les sous-nœuds correspondants
- Récurrence sur les nœuds restants

ID3 Exemple

Exemple *[Quinlan,86]*

Attributs	**Pif**	**Temp**	**Humid**	**Vent**
Valeurs possibles	soleil,couvert,pluie	chaud,bon,frais	normale,haute	vrai,faux

N°	Pif	Temp	Humid	Vent	Golf
1	soleil	chaud	haute	faux	NePasJouer
2	soleil	chaud	haute	vrai	NePasJouer
3	couvert	chaud	haute	faux	Jouer
4	pluie	bon	haute	faux	Jouer
5	pluie	frais	normale	faux	Jouer
6	pluie	frais	normale	vrai	NePasJouer
7	couvert	frais	normale	vrai	Jouer
8	soleil	bon	haute	faux	NePasJouer
9	soleil	frais	normale	faux	Jouer
10	pluie	bon	normale	faux	Jouer
11	soleil	bon	normale	vrai	Jouer
12	couvert	bon	haute	vrai	Jouer
13	couvert	chaud	normale	faux	Jouer
14	pluie	bon	haute	vrai	NePasJouer

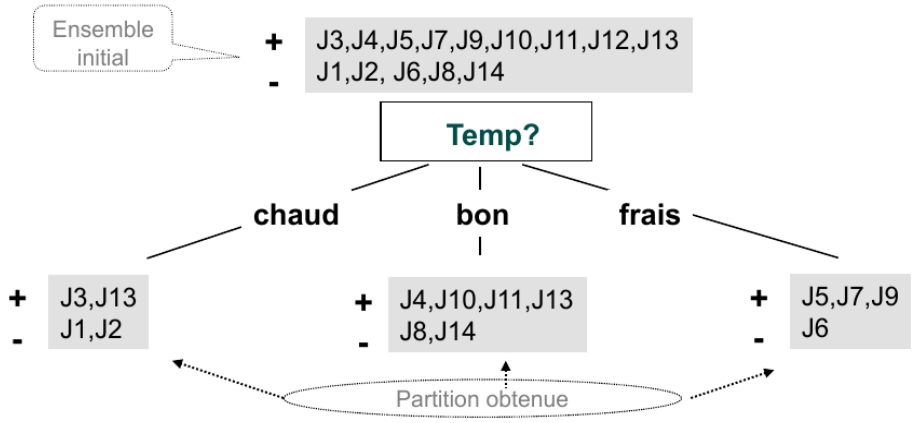
la classe

ID3 Exemple

Développement de l'arbres de décision : exemple

Si on choisit l'attribut « Temp »
pour le nœud racine

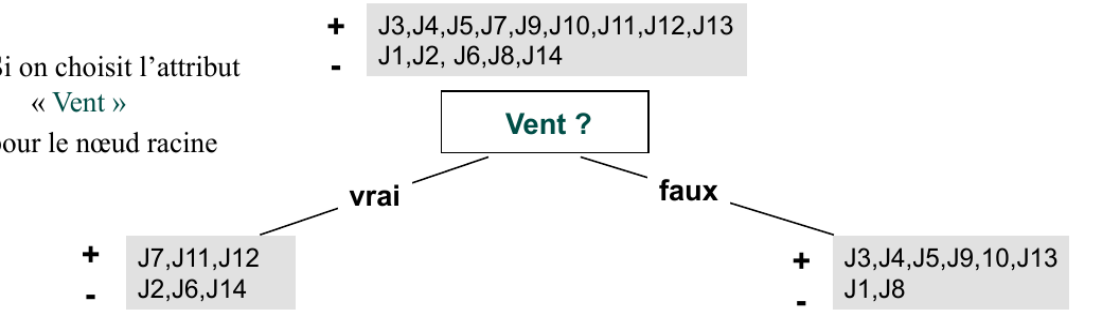
N°	Pif	Temp	Humid	Vent	Golf
1	soleil	chaud	haute	faux	NePasJoue
2	soleil	chaud	haute	vrai	NePasJoue
3	couvert	chaud	haute	faux	Jouer
4	pluie	bon	haute	faux	Jouer
5	pluie	frais	normale	faux	Jouer
6	pluie	frais	normale	vrai	NePasJoue
7	couvert	frais	normale	vrai	Jouer
8	soleil	bon	haute	faux	NePasJoue
9	soleil	frais	normale	faux	Jouer
10	pluie	bon	normale	faux	Jouer
11	soleil	bon	normale	vrai	Jouer
12	couvert	bon	haute	vrai	Jouer
13	couvert	chaud	normale	faux	Jouer
14	pluie	bon	haute	vrai	NePasJoue



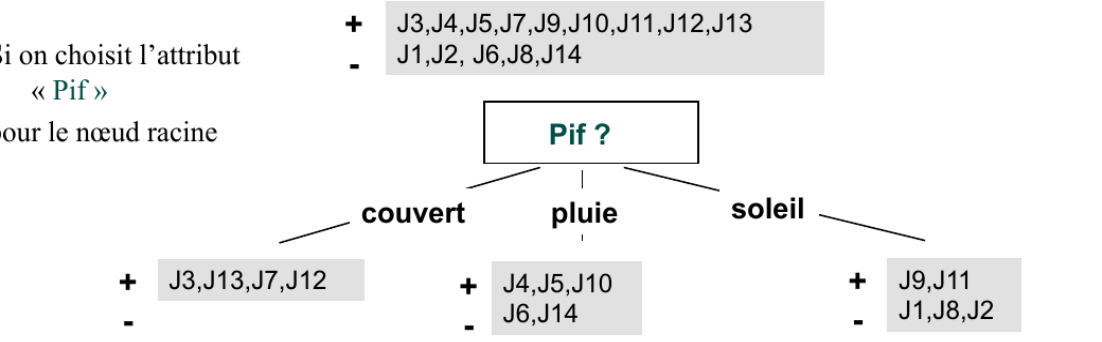
ID3 Exemple

Induction d'arbres de décision : sélection de l'attribut

Si on choisit l'attribut
« Vent »
pour le nœud racine

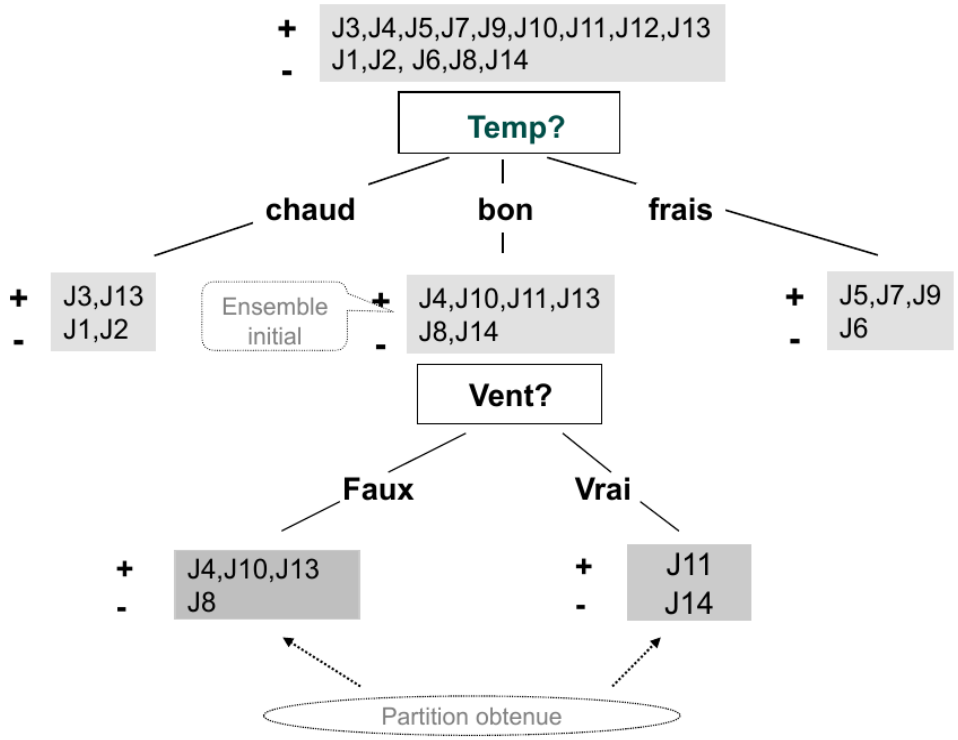


Si on choisit l'attribut
« Pif »
pour le nœud racine



ID3 Exemple

Développement de l'arbre à partir d'un noeud pendant (feuille)
Exemple



ID3 Exemple

3- Exemple

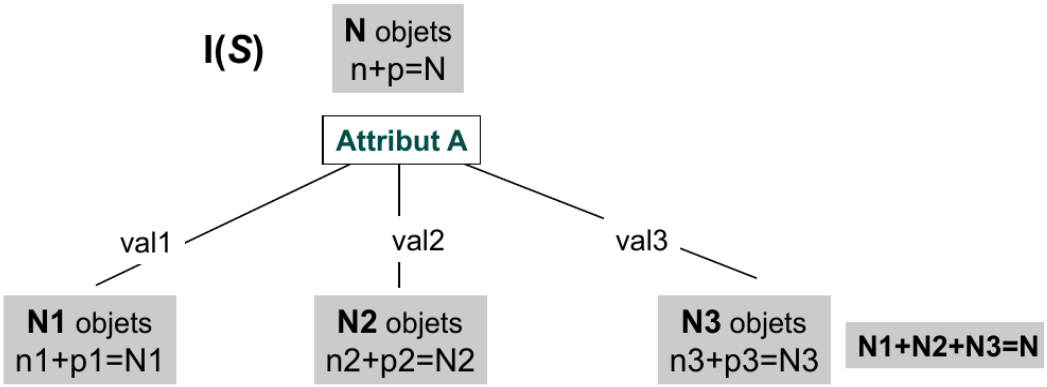
- Entropie de l'ensemble initial d'exemples

$$I(p,n) = -\,9/14 \log_2(9/14) - 5/14 \log_2(5/14)$$

- Entropie des sous-arbres associés au test sur Pif ?
  - $p_1 = 4 \;\; n_1 = 0 : \;\; I(p_1,n_1) = 0$
  - $p_2 = 2 \;\; n_2 = 3 : \;\; I(p_2,n_2) = 0.971$
  - $p_3 = 3 \;\; n_3 = 2 : \;\; I(p_3,n_3) = 0.971$
- Entropie des sous-arbres associés au test sur Temp ?
  - $p_1 = 2 \;\; n_1 = 2 : \;\; I(p_1,n_1) = 1$
  - $p_2 = 4 \;\; n_2 = 2 : \;\; I(p_2,n_2) = 0.918$
  - $p_3 = 3 \;\; n_3 = 1 : \;\; I(p_3,n_3) = 0.811$

ID3 Exemple

Exemple (cas de 2 classes)



$$\mathbf{E(N,A)}= N1/N \times I(p1,n1) + N2/N \times I(p2,n2) + N3/N \times I(p3,n3)$$

Le gain d'entropie de A vaut: $\mathbf{GAIN(A)= I(S)-E(N,A)}$

ID3 Exemple

- Pour les exemples initiaux

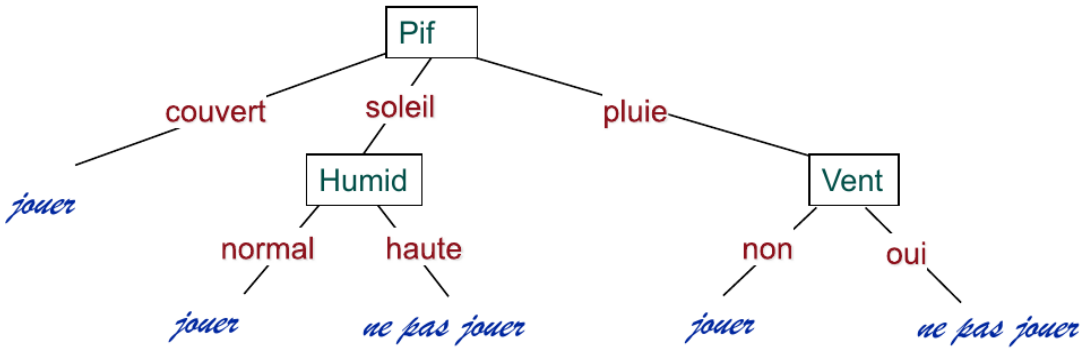
  $I(S) = - 9/14 \log_2(9/14) - 5/14 \log_2(5/14)$
- Entropie de l'arbre associé au test sur Pif ?
  - $E(Pif) = 4/14\ I(p_1,n_1) + 5/14\ I(p_2,n_2) + 5/14\ I(p_3,n_3)$

    $Gain(Pif) = 0.940 - 0.694 = 0.246$ bits
  - $Gain(Temp) = 0.029$ bits
  - $Gain(Humid) = 0.151$ bits
  - $Gain(Vent) = 0.048$ bits

Choix de l'attribut Pif pour le premier test

ID3 Exemple

- Arbre final obtenu :



ID3 (Iterative Dichotomiser 3)

Sortie de la récursivité :

- Tous les éléments de S sont dans la même classe ($H(S) = 0$) : S devient nœud feuille
- Pas d'attributs non utilisés : nœud feuille sur le classe majoritaire
- $S = \emptyset$: nœud feuille sur le classe majoritaire du parent (ce cas est nécessaire pour la classification de nouveau échantillons)

Problèmes :

- Solution globale non garantie (optimum local, amélioration : backtracking)
- Over-fitting (pour éviter : préférer les arbres de taille réduite)
- Pas efficace pour des données numériques continues

C4.5 (Iterative Dichotomiser 4.5)

C4.5 : extension de ID3

- Le critère de division est le gain d'information normalisé maximal (différence d'entropie avant et après la division)
- Chaque attribut peut avoir un poids (coût)
- Traitement de variables continues en cherchant des seuils qui maximise le gain d'information
- Traitement de valeurs manquantes
- Étape d'élagage après la création pour remplacer des branches inutiles par des feuilles

C5.0 : extension de ID4.5

- Vitesse et utilisation mémoire
- Arbres plus petits
- Pondération des cas et erreurs de classification

Classification and Regression Trees (CART)

Breiman, Friedman, Olshen, Stone, *Classification and regression trees*, Monterey, Brooks/Cole Advanced Books, 1984.

CART : Arbres de classification et régression

- CART pose seulement de questions test binaires (arbres binaires)
- Fonctionne aussi pour des attributs aux valeurs continues
- CART cherche tous les attributs et tous les seuils pour trouver celui qui donne la meilleure homogénéité du découpage

Classification and Regression Trees (CART)

Un noeud interne S est coupé sur l'attribut j, seuil a_j :

- Sous-noeud gauche S_g ($p_g \approx |S_g|/|S|$) et
- Sous-noeud droit S_d ($p_d \approx |S_d|/|S|$)

Soit $I(S)$ la fonction de l'impureté de S par rapport à la classe cible.
CART étudie le changement de l'impureté par rapport au seuil et pour tous les attributs :

- $E[I(S_{gd})] = p_g I(S_g) + p_d I(S_d)$
- $\Delta I(S) = I(S) - E[I(S_{gd}) = I(S) - p_g I(S_g) - p_d I(S_d)$

Problème d'optimisation :

- $\arg\max_{j;a_j} \Delta I(S)$

Classification and Regression Trees (CART)

Pb. de **classification** optimise l'**index (ou impureté) de Gini** :

- La vraisemblance qu'un élément du nœud sera incorrectement labellisé par un tirage aléatoire qui respecte la loi statistique de la cible estimé dans le nœud.

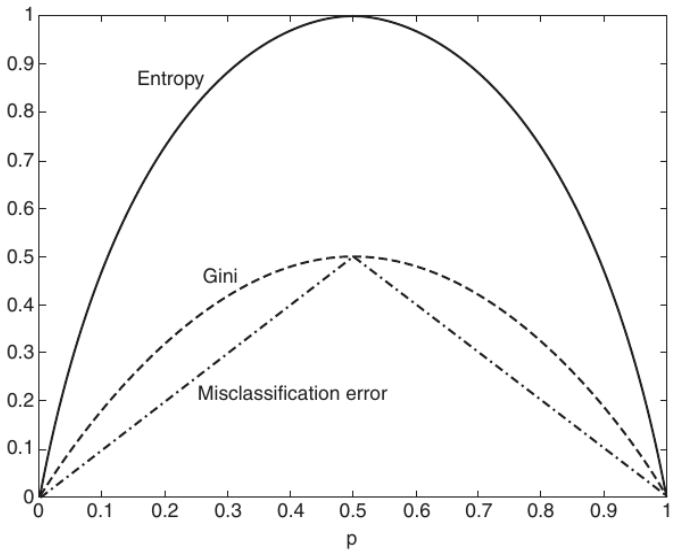
S un nœud interne :

- Partitionner S sur les valeurs de la cible en n groupes : $C_1, \ldots, C_m$
- p_i : probabilité estimé qu'un élément de S se retrouve dans C_i ($p_i \approx |C_i|/|S|$)
- $I_G(S) = \sum_{1=1}^{m} p_i(1-p_i) = \sum_{1=1}^{m}(p_i - p_i^2) = 1 - \sum_{1=1}^{m} p_i^2$
- $I_G(S) = \sum_{i \neq j} p_i p_j$ index de Gini
- $I_G(S) = 0$ si S est homogène (tous les éléments sont dans la même classe — impureté du groupe nulle)

Classification and Regression Trees (CART)

Classification : autres types de mesures d'**impureté** :

- $H(s) = -\sum_i p_i log(p_i)$ (entropie)
- $E(s) = 1 - \max_i p_i$ (erreur de classification)



Comparaison mesures d'impureté des noeuds.

Classification and Regression Trees (CART)

Pb. de **régression** optimise le **résidu quadratique moyen** : minimise la variance moyenne des groupes.

- $\arg\min_{j;a_j} p_g Var(S_g) + p_d Var(S_d)$

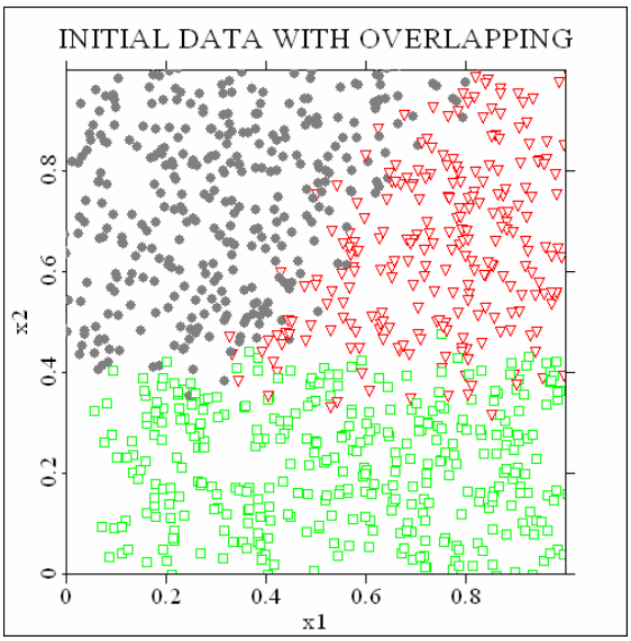
Classification de nouvelles données :

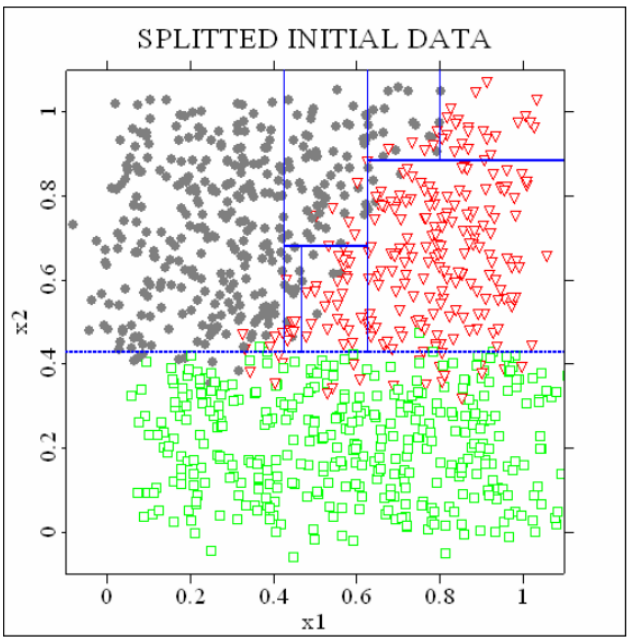
- Parcours de l'arbre pour arriver dans une feuille
- La classe dominante (majoritaire) dans ce noeud donne la classification
- Pour la régression : on considère les valeurs dominantes dans les feuilles

Avantages CART :

- Forme non paramétrique
- Pas de sélection de variables nécessaire
- Invariable aux transformation monotones des attributs
- Bonne gestion des *ouliers*

Classification and Regression Trees (CART)



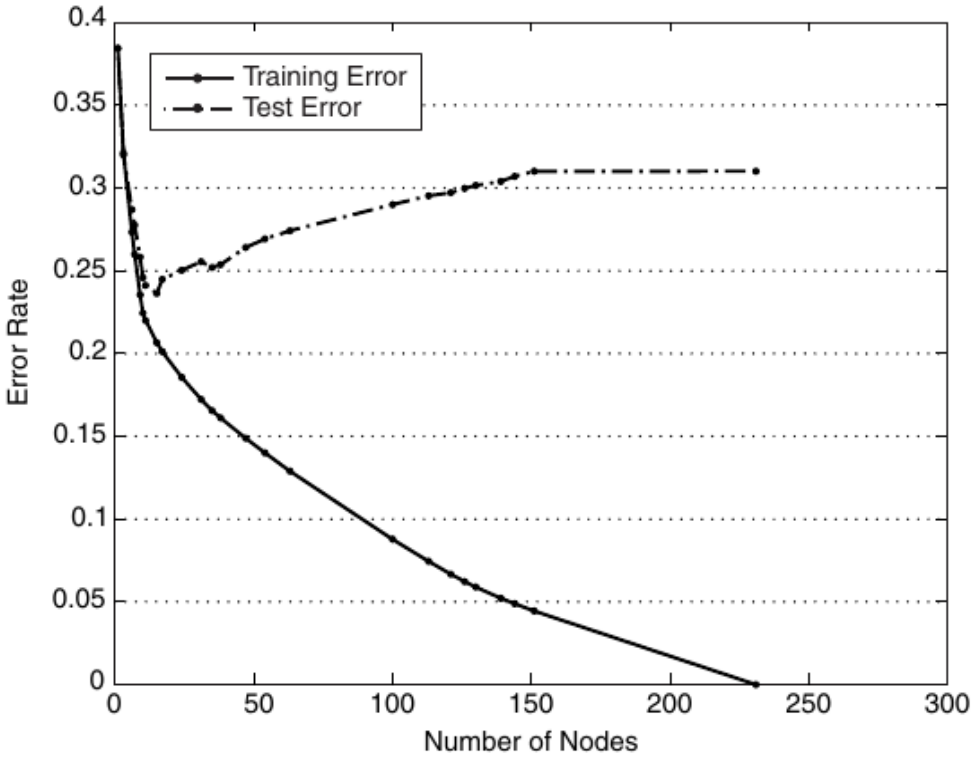


Classification and Regression Trees (CART)

Sur-apprentissage :

- Pour des pb. non-linéaires CART peut donner des arbres de grande tailles avec beaucoup de feuilles qui ont peu d'éléments (souvent un seul)
- Les premiers *splits* sont généralement les plus importants et les moins dépendants de l'échantillon, tandis que les suivants décrivent des particularités plus subtiles, pouvant être propres à l'échantillon .
- Il est donc souhaitable, afin de garder un niveau correct de généralité, d'élaguer l'arbre construit.
- Un taux d'erreur de prédiction par validation croisée est calculé pour différentes tailles de l'arbre (i.e., différents nombres de feuilles terminales) : l'arbre est alors à élaguer au niveau offrant l'erreur minimale.

Classification and Regression Trees (CART)



Taux d'erreurs : construction versus test.

Classification and Regression Trees (CART)

Gestion des données manquantes :

- *Surrogate splits* ou variables-substituts : l'opération continue sur un autre attribut qui, à l'apprentissage, a donné un split similaire

Plan du cours

Extensions

- **Bagging decision trees** : construction plusieurs arbres par re-échantillonnage avec remise ; prise de décision par vote consensuel
- **Forêts d'arbres décisionnels (ou forêts aléatoires)** : apprentissage sur de multiples arbres de décision entraînés sur des sous-ensembles de données légèrement différents.

Références

Livres et articles :

- Rokach, Lior ; Maimon,Data mining with decision trees : theory and applications. World Scientific Pub Co Inc., 2008
- Quinlan, Induction of Decision Trees. Machine Learning 1 : 81-106, Kluwer Academic Publishers 1986
- Hastie, Tibshirani, Friedman, The elements of statistical learning : Data mining, inference, and prediction. New York : Springer Verlag, 2006
- Breiman, Friedman, Olshen, Stone, Classification and regression trees. Monterey, CA : Wadsworth and Brooks/Cole Advanced Books 1984
- Roman Timofeev, Classification and Regression Trees (CART) Theory and Applications, Master Thesis, Université Humbold, Berlin, 2004