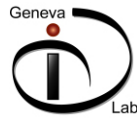


# Les arbres de décision



22 Septembre 2009

## Plan

- 1 Le partitionnement récursif
- 2 C4.5
- 3 CART
- 4 Evaluation de performances
- 5 Bilan

## Les données du Titanic

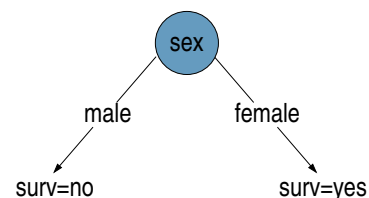
- Données historiques sur 2201 passagers du Titanic
- Tâche : prédire la survie d'un passager sur la base de 4 variables
- Var cible : survie {yes, no}
- Vars prédictives
  - classe {1st,2nd,3rd,crew}
  - age {adult, child}
  - sexe {male, female}

class	age	sex	surv
1st	adult	m	yes
crew	adult	m	no
3rd	child	m	no
2nd	adult	f	yes
...	...	...	...

## Le principe du partitionnement

Etant donné un ensemble de données  $S$  ayant  $d$  variables prédictives  
 Trouver un test permettant de prédire la valeur de la variable cible

- 1 Choisir une variable de test  $x$  suivant un critère défini
- 2 Partitionner les exemples suivant les valeurs de  $x$
- 3 Pour chaque partition, prédire la valeur de la variable cible



Un arbre à un noeud

Pour obtenir des arbres de complexité arbitraire, on applique cet algorithme de manière récursive.

# L'algorithme de partitionnement récursif

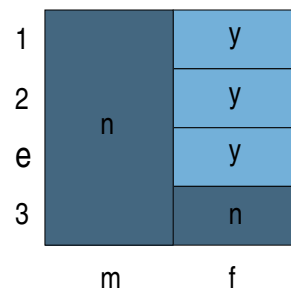
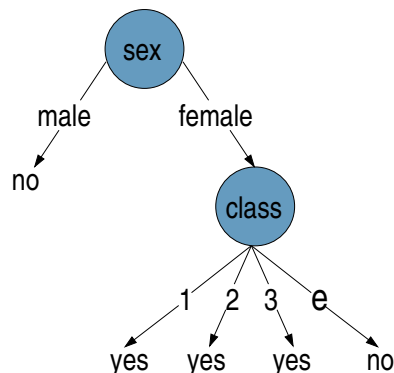
**DT**( $S$ : données,  $X$ : vars prédictives,  $Y$ : var cible)

- 1 Créer noeud  $T$
- 2 Si  $X = \emptyset$  ou si  $Y$  a la même valeur  $\forall s \in S$  alors retourner  $T$  avec prédiction :
  - en classification : classe majoritaire dans  $S$
  - en régression : moyenne des  $y_i$  dans  $S$
- 3 Choisir une variable de test  $x \in X$  suivant un critère défini\*
- 4 Partitionner  $S$  en  $m$  sous-ensembles  $S'$  suivant les valeurs de  $x^*$
- 5 Si  $x$  discrète,  $X \leftarrow X - \{x\}$
- 6 Pour chaque partition  $S'$   
 $DT(S', X, Y)$

\* Voir page 9

## Remarques sur l'algorithme

- algorithme glouton pour éviter la recherche combinatoire : ni regard en avant, ni backtrack
- découpe des hyperrectangles dans l'espace des instances : frontières perpendiculaires aux axes



## Variations sur l'algorithme de base

- Critère de choix de l'attribut de test (ligne 3)  
CART : index Gini  
C4.5 : gain d'information et rapport de gain
- Facteur de branchement  $b$  pour l'attribut de test  $X$  (ligne 4)  
CART :  $b = 2$  (arbre binaire)  
C4.5 : par défaut,  $b = |\mathcal{X}|$  si  $x$  nominal,  $b = 2$  continu
- Même stratégie pour restreindre la complexité de l'arbre :  
Construire l'arbre jusqu'au bout, puis élaguer.

## Plan

- 1 Le partitionnement récursif
- 2 **C4.5**
- 3 CART
- 4 Evaluation de performances
- 5 Bilan

## Entropie d'une variable

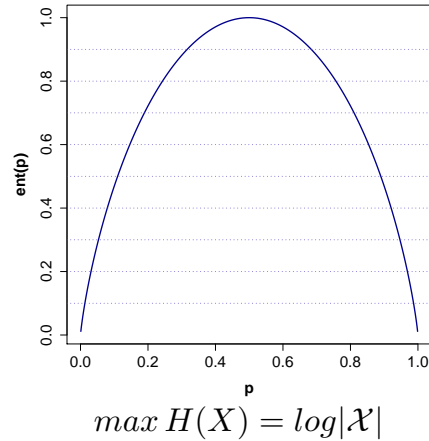
Soit une variable aléatoire  $X$  prenant ses valeurs dans l'alphabet  $\mathcal{X}$

- L'incertitude d'un événement  $x$ :  
l'inverse de sa probabilité  
 $\log \frac{1}{p(x)} = -\log p(x)$ .
- L'entropie de  $X$ :

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log p(x)$$

où  $\log$  s'entend à la base 2.

- L'entropie = nombre de bits  
requis pour décrire  $X$

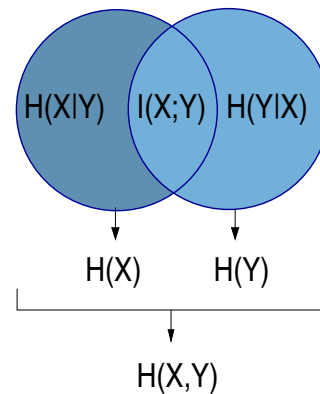


## Entropie conditionnelle

- Soient 2 v.a.  $X$  et  $Y$ .  
L'entropie conditionnelle  $H(Y|X)$

$$\begin{aligned} &= \sum_{x \in \mathcal{X}} p(x) H(Y|X = x) \\ &= \sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y|x) \log p(y|x) \end{aligned}$$

- Pour que  $X$  serve à prédire  $Y$ , il faut  $H(Y) - H(Y|X) > 0$



## Information mutuelle et choix de la variable

- L'information mutuelle entre  $X$  et  $Y$  = quantité d'info sur  $Y$  apportée par la connaissance de  $X$  et vice-versa = **gain d'information**

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) \\ &= H(X) - H(X|Y) \\ &= 0 \Leftrightarrow X \text{ et } Y \text{ indépendantes} \end{aligned}$$

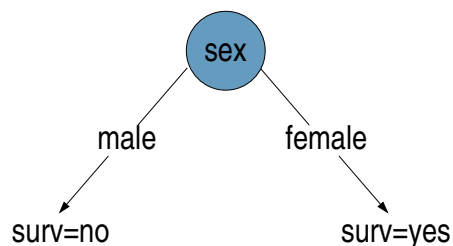
- Inconvénient :  $I(X;Y) \nearrow$  avec  $|\mathcal{X}|$  : favorise les variables ayant bcp de valeurs distinctes
- Solution dans C5 : normaliser le gain d'info par l'entropie de la variable prédictive  $X \rightarrow$  **rapport de gain** (critère par défaut)

$$IGR = \frac{I(X;Y)}{H(X)}$$

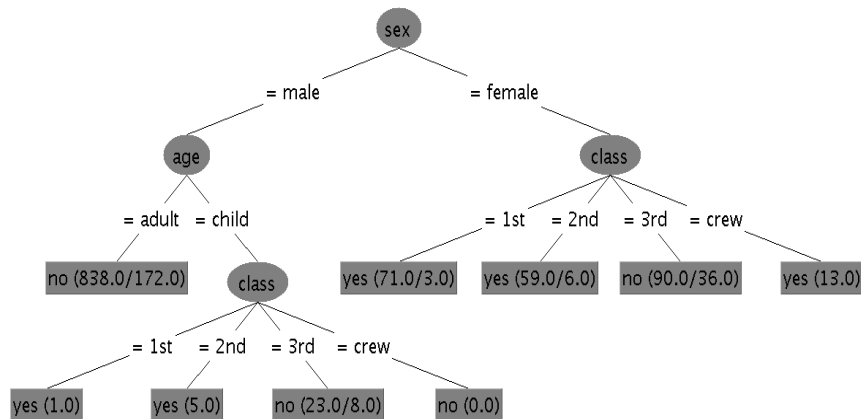
## Exemple : choix de la variable racine

Sur l'ensemble d'apprentissage TRN

$$\begin{aligned} H(surv) &= 0.908 \\ I(class; surv) &= 0.052 \\ I(age; surv) &= 0.005 \\ I(sex, surv) &= 0.139 \end{aligned}$$



## Arbre C4.5 sur le Titanic



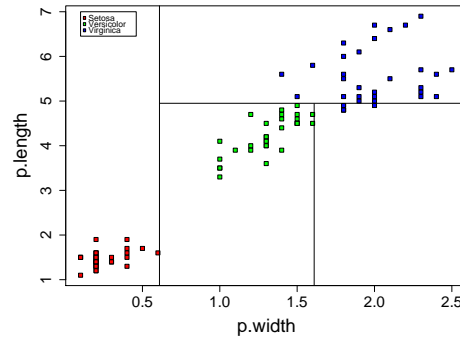
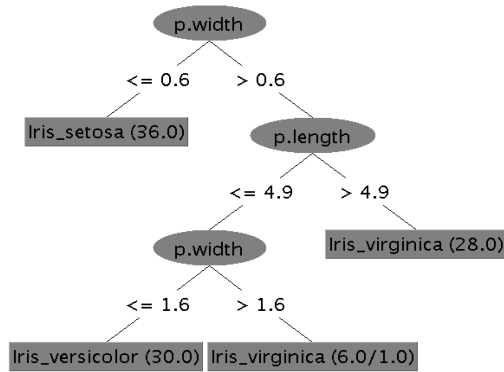
Paramètre  $m$  (nb min d'ex par feuille): contrôle la complexité  
 Ici:  $m = 2 \neq$  arbre p. 6:  $m = 20$

## Partitionnement binaire sur les variables continues

Idée: transformer une variable continue  $X$  en variable booléenne:  
 trouver un seuil  $t$  permettant d'avoir 2 groupes homogènes par  
 rapport à la variable cible

- 1 Identifier les seuils potentiels parmi les valeurs distinctes  $x_i$  de  $X$ 
  - Trier les exemples dans l'ordre croissant des  $x_i$
  - Les seuils potentiels =  $x_i$  adjacents ayant des classes différentes  
 Ex. iris: petal.length : 36 valeurs distinctes, mais 6 seuils potentiels
- 2 Choisir la partition qui maximise le critère choisi (gain d'info ou rapport de gain en C4.5)

## Arbre C4.5 sur les iris



## Post-élagage de l'arbre C4.5

Critère : réduction de l'erreur

- Partitionner les données en ensemble d'apprentissage TRN et ensemble de validation VAL
- Construire un arbre en utilisant TRN
- Convertir un noeud interne en feuille si son erreur sur VAL n'est pas supérieure à la somme d'erreur de ses fils

## Plan

- 1 Le partitionnement récursif
- 2 C4.5
- 3 **CART: Classification and Regression Trees**
- 4 Evaluation de performances
- 5 Bilan

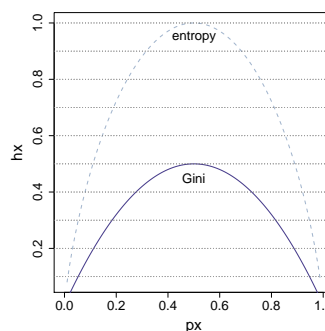
## Mesure d'impureté en classification

- L'index Gini  $G(t)$  mesure l'impureté des données au noeud  $t$

$$G(t) = \sum_i p_i(1 - p_i)$$

où  $i$  = valeur de la variable cible  $Y$  et  $p_i = p(i|t)$ .

- $Min(G) = 0 \Leftrightarrow$  tous les exemples ont la même valeur de la variable cible
- $Max(G) = 1 - \frac{1}{|Y|} \Leftrightarrow$  toutes les valeurs  $i$  équiprobables



## Mesure d'impureté en régression

- La prédiction  $y(t)$  au noeud  $t$ : la moyenne des  $y_i$ , valeurs de la variable cible dans les  $n_t$  exemples au noeud  $t$ .
- La mesure d'impureté = MSE (erreur quadratique moyenne)

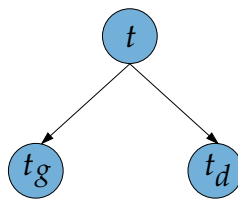
$$R(t) = \frac{1}{n_t} \sum_{i=1}^{n_t} (y_i - y(t))^2.$$

- Puisque  $y(t) = \bar{y}_i$ , la mesure d'impureté|d'erreur = la variance intra-noeud des  $y_i$ .
- L'erreur totale de l'arbre = somme des erreurs|variances aux feuilles.

## Dichotomie et impureté des données

- On mesure la baisse d'impureté des données  $S$  après une dichotomie effectuée au noeud  $t$ .
  - Classification :  $\Delta G(S, t) = G(t) - (p_g G(t_g) + p_d G(t_d))$
  - Régression :  $\Delta R(S, t) = R(t) - (p_g R(t_g) + p_d R(t_d))$

$t_g|t_d$  = fils gauche/droit  
 $p_g|p_d$  = proportion des cas  
 ∈ fils gauche/droit



$G(t)$   
 $R(t)$

$G(t_g) + G(t_d)$   
 $R(t_g) + R(t_d)$

- Calculer  $\Delta G$  pour toutes les dichotomies possibles au noeud  $t$ , puis choisir celle qui maximise  $\Delta G$ .

## Choix de la dichotomie/de la variable

- Pour chaque variable candidate  $X$  (dom.  $\mathcal{X}$ ) en lice au noeud  $t$ 
  - calculer  $\Delta G|\Delta R$  pour tous les tests binaires possibles
    - var continue  $x \leq t?$  ( $t$  = seuil potentiel, voir p. 13)
    - var discrète  $x \in \mathcal{X}' \subset \mathcal{X}?$
  - choisir la dichotomie qui maximise  $\Delta G|\Delta R$
- Choisir la variable concernée par la dichotomie ayant  $\Delta G|\Delta R$  maximale.

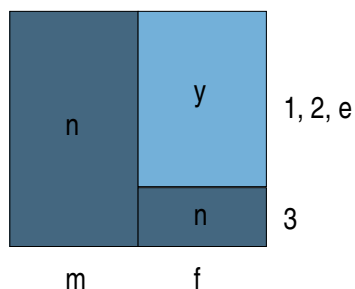
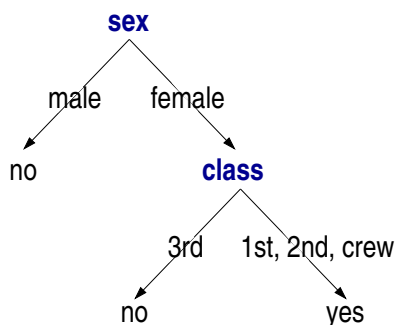
## Post-élagage de l'arbre CART

- Critère : compromis entre erreur et complexité d'un arbre  $T$  :

$$C_\lambda(T) = E(T) + \lambda|T|$$

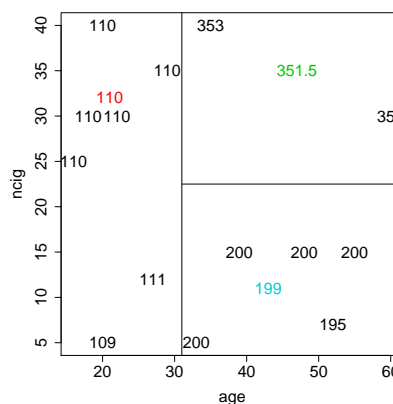
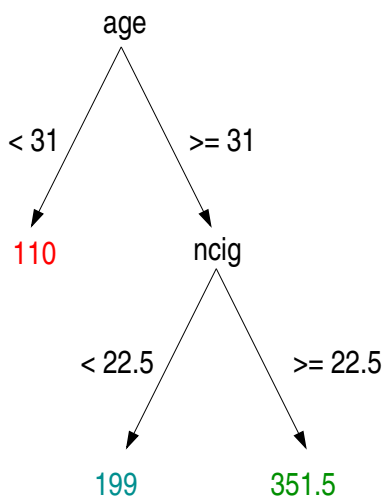
- $E(T)$  = erreur totale (TBC ou MSE) aux feuilles de  $T$
- $|T|$  = nombre de feuilles de l'arbre  $T$ : mesure de la complexité du modèle
- $\lambda$  = un paramètre de **régularisation** qui contrôle le compromis entre performance et complexité
- la valeur de  $\lambda$  est déterminée empiriquement (validation croisée, à voir plus tard)

# Arbre de classification CART



$m = 20$ . cf. l'arbre C4.5 page 6

# Arbre de régression CART



# Plan

- 1 Le partitionnement récursif
- 2 C4.5
- 3 CART
- 4 **Evaluation de performances**
- 5 Bilan

# Principes de base de l'évaluation

Pour évaluer l'efficacité d'un prédicteur, il faut

- 1 une mesure de performance
  - classification :  $TBC = \frac{\text{nb de cas bien classés}}{\text{nb total de cas}}$
  - régression :  $MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$
- 2 une référence de base (baseline)
  - le classifieur par défaut : prédit toujours la classe majoritaire
  - le régresseur par défaut : prédit toujours la moyenne de la var. cible
- 3 un ensemble de test indépendant de l'ensemble d'apprentissage

## Répartition des données

- Pour éviter le surapprentissage, nous répartissons les données en 2 sous-ensembles (proportions suivant la taille des données)
  - l'ensemble d'entraînement (TRN) : servira à construire nos modèles
  - l'ensemble de test (TST) : servira à valider les modèles construits
- Les partitions doivent être stratifiées : doivent conserver la distribution d'origine de la variable cible

	tout	trn	tst	%
surv=ooui	711	356	355	32
surv=non	1490	744	746	68
total	2201	1100	1101	100

## Evaluation des modèles sur le Titanic

Classification			
TBC	TRN [n=1100]	TST [n=1101]	CPX
Baseline	68%	68%	
J48, m=2	79.5%	78.564%	T =9
J48, m=20	79.0%	77.66%	T =5
CART, m=20	79.0%	77.66%	T =3
Régression			
MSE	TRN [n=14]	TST [n=6]	CPX
Baseline	6768.78	11749.89	
CART, m=5	1.89	3809.25	T =3

# Bilan

- Avantages
  - apprentissage très rapide
  - compréhension du modèle
  - robustesse aux variables non pertinentes
- Inconvénients
  - instabilité : très sensible aux variations des données
  - incapacité à détecter les interactions entre variables
  - puissance de représentation assez limitée : découpes orthogonales trop peu adaptées aux problèmes demandant des frontières obliques et lisses