

Apprentissage supervisé

Partie II – Segmentation par arbres

Brigitte Gelein – bgelein@ensai.fr



École nationale
de la statistique
et de l'analyse
de l'information

Segmentation par arbres

Decision Trees

Sommaire

A – Généralités sur la discrimination par arbres 6

1. Principe de la segmentation	7
2. Exemples d'arbres	11
3. Construction d'un arbre	17
4. Règle d'arrêt	18

B – CART : Classification and Regression Trees 20

Introduction	21
1. Divisions binaires	25
2. Arbres de classement	28
2.1 Construire l'arbre maximal	30
2.2 Elaguer l'arbre maximal	57
2.2.1 Séquence d'élagage	65
2.2.2 Choix du meilleur sous-arbre	79
a. Echantillon-test	79
b. Validation croisée	97

Sommaire

2.3 Division équi-réductrice et équi-divisante	105
2.4 Résultats	109
3. Arbres de régression	110
3.1 Mesure de l'impureté d'un nœud	110
3.2 Meilleure division d'un nœud	114
3.3 Règle de prédiction	116
3.4 Coût de l'arbre	117
4.Exemple Spotify avec R	122
5.Bilan	126

C – CHAID 128

1. Arbres de classement	130
1.1 La statistique du χ^2	131
1.2 Construction de l'arbre	133
1.3 Regroupement des modalités des X_j	136
1.4 L'algorithme	142
1.5 Correction de Bonferroni	144

Sommaire

2. Arbres de régression 147

2.1 La statistique de Fischer 148

2.2 L'algorithme 151

D – Avantages et inconvénients des arbres de décision 154

1. Avantages 155

2. Inconvénients 158

A – Généralités sur la discrimination par arbres

1. Principe de la segmentation

- Construire un **arbre** à l'aide de divisions successives d'un ensemble d'individus appartenant à un échantillon.
- Chaque **division** (ou **scission**) conduit à deux (ou plus) **nœuds** (ou **segments**) :
 - le nœud divisé est appelé **nœud-parent**,
 - les nœuds générés par la division s'appellent **nœuds-enfants**.
- Les **nœuds contiennent des groupes d'individus** les plus **homogènes possible par rapport à une variable à expliquer Y** nominale, ordinale ou quantitative.

2. Principe de la segmentation

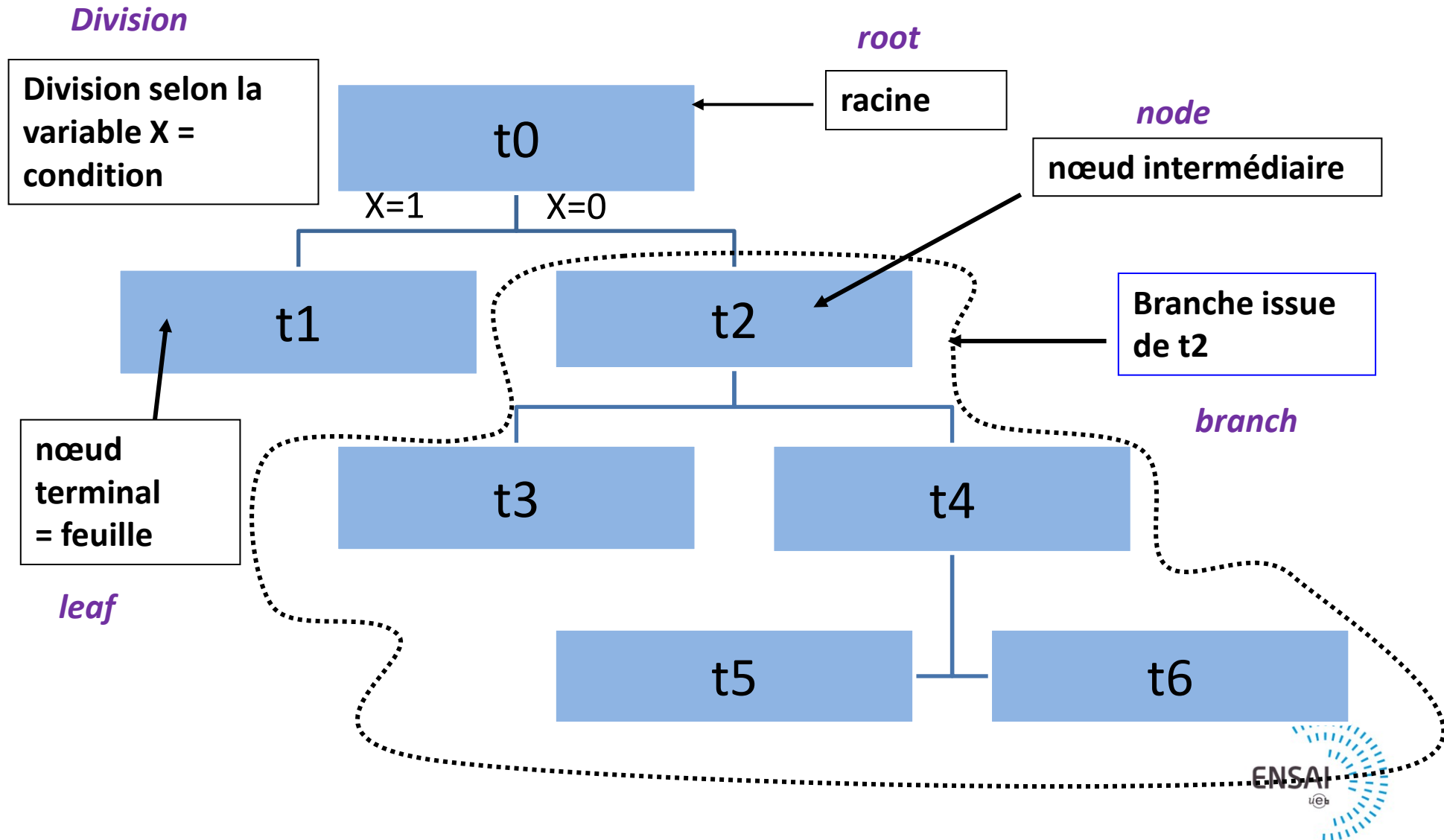
- Les divisions s'opèrent à partir de **variables explicatives** (ou prédicteurs) $X_1 \dots X_j \dots X_J$, qui peuvent être nominales, ordinales ou quantitatives.

Résultats obtenus sous la forme d'un arbre inversé :

- la **racine** (en haut de l'arbre)
représente l'échantillon total à segmenter,
- les autres nœuds sont
 - soit des **nœuds intermédiaires** (encore divisibles),
 - soit des **nœuds terminaux**.

L'ensemble des nœuds terminaux constitue une **partition** de l'échantillon initial en groupes.

2. Principe de la segmentation



2. Principe de la segmentation

- Si la variable à expliquer Y est qualitative, on parle :
 - **d'arbre de classement**
 - **de discrimination par arbre**
- Si la variable à expliquer Y est quantitative, on parle :
 - **d'arbre de régression**
 - **de régression par arbre**
- On parle d'**arbre binaire** si toutes les divisions conduisent à 2 nœuds (**divisions binaires**).

3. Exemples d'arbres

Exemple d'un arbre de classement

- **Exemple d'analyse de défaut de paiement :**
déterminer quels individus, candidats à un crédit auprès d'une banque, sont susceptibles de rembourser leur emprunt et lesquels ne le sont pas (variable à expliquer Y), d'après les informations qu'ils fournissent (variables explicatives).
- **Variable à expliquer :**
Y à deux modalités = bon payeur / mauvais payeur



3. Exemples d'arbres

Variables prédictives : nominales et ordinales

- Catégorie d'âge : ordinale
jeune / moyen / âgé
- Contrat de travail : nominale
CDI / CDD et autres
- Catégorie socioprofessionnelle : nominale
direction / cadre / employé de bureau / ouvrier / non qualifié

Les données portent sur 3230 individus.



\hat{f} ?



Noeud 0		
Catégorie	%	n
 Bon payeur	48	1550
 Mauvais payeur	52	1680
Total	100	3230

Type de contrat

CDI

CDD et autres contrats

Noeud 1		
Catégorie	%	n
 Bon payeur	84	1330
 Mauvais payeur	15	250
Total	49	1580

Noeud 2		
Catégorie	%	n
 Bon payeur	13	220
 Mauvais payeur	86	1430
Total	51	1650

Catégories d'âges



Catégories d'âges



Jeune (< 25)



Moyen (25-35); Agé (> 35)



Moyen (25-35); Jeune (< 25)

Agé (> 35)

Noeud 3		
Catégorie	%	n
 Bon payeur	51	250
 Mauvais payeur	48	240
Total	15	490

Noeud 4		
Catégorie	%	n
 Bon payeur	99	1080
 Mauvais payeur	1	10
Total	34	1090

Noeud 5		
Catégorie	%	n
 Bon payeur	9	150
 Mauvais payeur	90	1430
Total	49	1580

Noeud 6		
Catégorie	%	n
 Bon payeur	100	70
 Mauvais payeur	0	0
Total	2	70

3. Exemples d'arbres

Exemple d'un arbre de régression

- **Objectif** : prédire la durée de survie en mois de malades atteints d'un cancer
- **Variable à expliquer** : durée de survie en mois (continue)
- **Variables prédictives** :
 - 1 . chimio (oui/non)
 - 2 . délai apparition métastases (4 MODALITES)
 - 3 . Karnofsky (bon/mauvais)

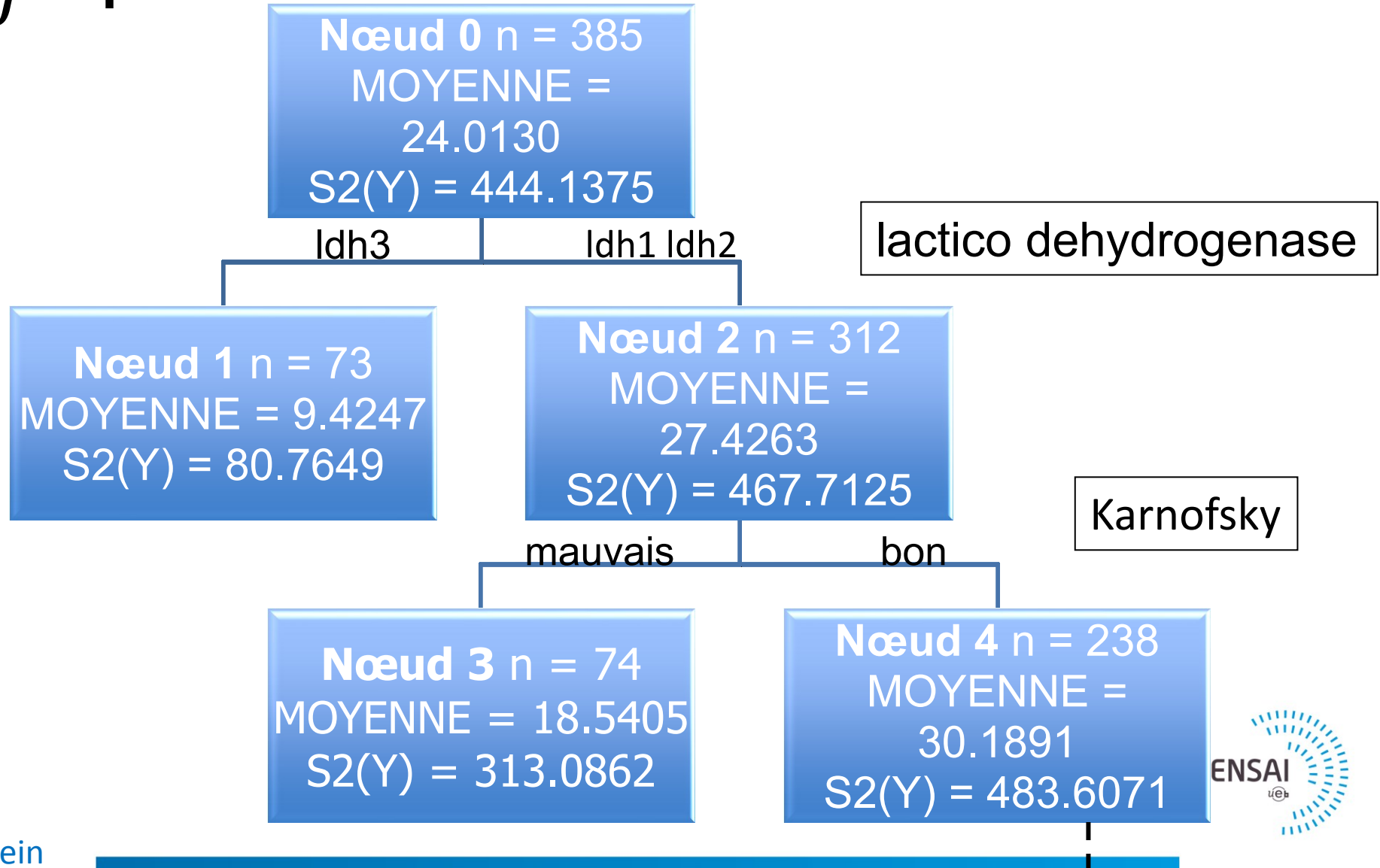
3. Exemples d'arbres

- 4 . metastases foie (oui/non)
- 5 . metastases poumon (oui/non)
- 6 . metastases plevre (oui/non)
- 7 . metastases os (oui/non)
- 8 . metastases cutanees (oui/non)
- 9 . nb sites metastatiques (3 MODALITES)
- 10 . lactico dehydrogenase (3 MODALITES)
- 11 . lymphocytes (2 MODALITES)
- 12 . albumine (2 MODALITES)

Les données portent sur 385 individus.

3. Exemples d'arbres

\hat{f} ?



4. Construction d'un arbre

1. Etablissement pour chaque nœud de l'ensemble des divisions possibles
2. Définition d'un critère permettant de sélectionner la meilleure division possible ***div****
3. Choix d'un (ou plusieurs) critère(s) d'arrêt des divisions
4. Choix d'une règle d'affectation de chaque nœud terminal (règle de décision pour prédire Y) :
 - à une **modalité** de Y si Y est **qualitative**
 - à une **valeur numérique** de Y si Y est **quantitative**
5. Estimation du coût d'erreur associé à l'arbre

5. Règles d'arrêt

- Ces ***règles d'arrêt*** sont des critères qui déterminent quand arrêter la génération de nœuds résultant de scissions.
- Ces critères sont de deux types :
 - "**statistiques**", et dépendent de la méthode de segmentation utilisée
 - "**arithmétiques**", communs à toutes les méthodes, présentés ci-après.

5. Règles d'arrêt

Critères communs aux différentes méthodes

- ***Nombre maximum de niveaux de l'arbre :***
nombre de niveaux en-dessous du niveau racine.
- ***Nombre minimum d'observations par nœud :***
 - nœud-parent : un nœud ayant moins d'observations que la valeur spécifiée n'est pas scindé
 - nœud-enfant : une division conduisant à un nœud-enfant ayant moins d'observations que la valeur spécifiée n'est pas réalisée

B - CART :

Classification and Regression Trees

CART - Introduction

Algorithme CART

CART est un algorithme de moyennage local par partition de \mathcal{X} (moyenne ou vote à la majorité sur les éléments de la partition). Cette partition de l'espace \mathcal{X} engendré par les variables explicatives est basée sur des **divisions successives parallèles aux axes de \mathcal{X}** , et dépend des (X_i, Y_i)

L'algorithme CART nécessite donc :

- ① la définition d'un critère permettant de sélectionner la "meilleure" division d'un noeud parmi toutes celles admissibles pour les différentes variables explicatives ;
- ② une règle permettant de décider qu'un noeud est terminal : il devient ainsi une feuille
- ③ l'affectation de chaque feuille à l'une des modalités (discrimination) ou à une valeur (régression) de la variable à expliquer.

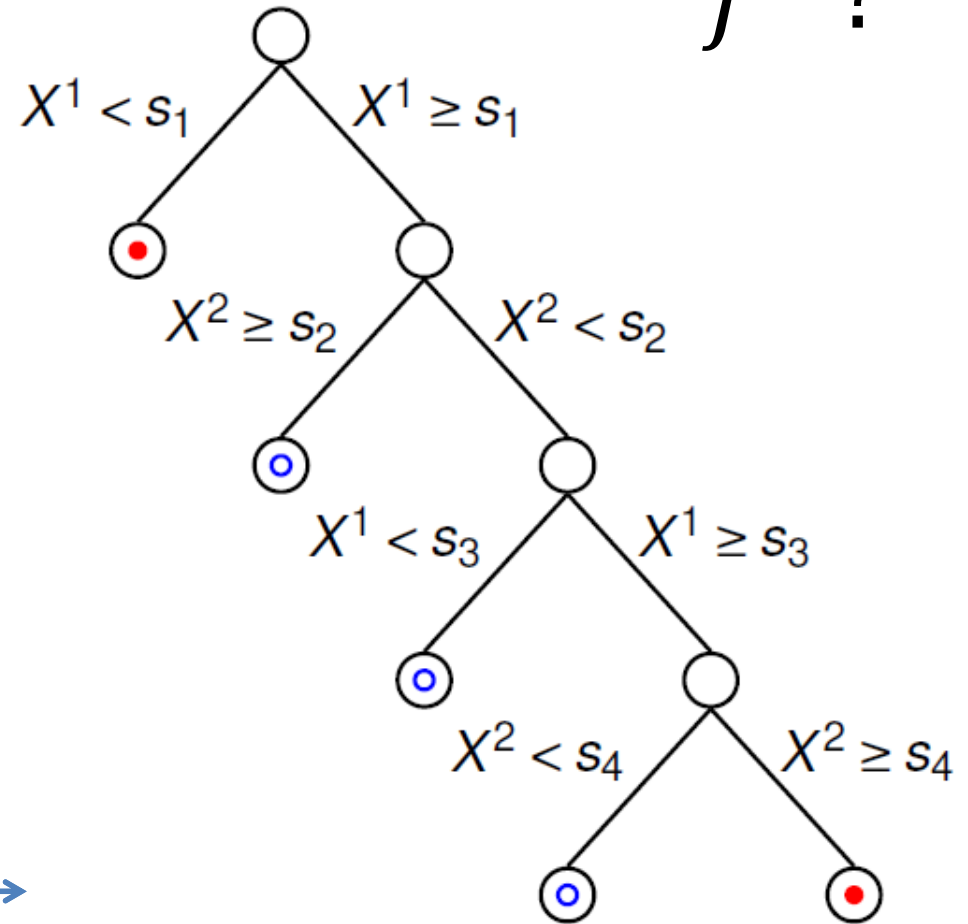
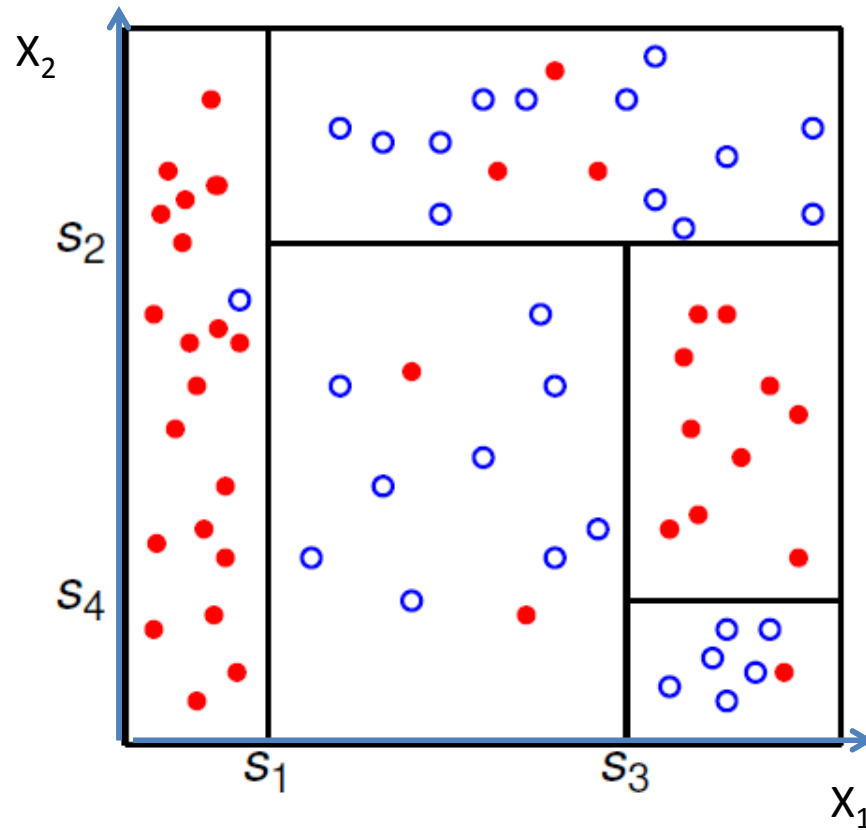
A chaque étape du partitionnement binaire, on cherche à séparer "au mieux" les données du noeud courant, en recherchant la **division** conduisant à la plus forte **diminution de l'hétérogénéité** des deux noeuds enfants.

- **Division** : Une variable explicative X_j et
 - ◇ un seuil s_j si X_j est quantitative ou
 - ◇ deux sous-ensembles de modalités si X_j est qualitative
- **Diminution de l'hétérogénéité** :
 - ◇ Impureté (Entropie de Shannon, indice de Gini) si Y est qualitative
 - ◇ Variance si Y est quantitative

Hétérogénéité : fonction de coût, loss function.

Arbres de décision : représentation

\hat{f} ?



CART - Introduction

- Variable cible : nominale, ordinale, ou quantitative
- Prédicteurs : variables nominales, ordinales ou quantitatives
- **divisions binaires** uniquement : deux nœuds enfants à chaque scission.
- pas (ou presque pas) de règle d'arrêt de la procédure de division, ce qui conduit à un arbre **maximal** de grande taille.
- Une procédure d'**élagage** permet alors d'en extraire un sous-arbre fiable, de taille plus réduite.

CART – 1. Divisions binaires

- **Divisions binaires possibles** selon le type de variables explicatives :
 - **Variable binaire** : 1 division
 - **Variable nominale à M modalités** : $2^{M-1} - 1$ divisions
 - **Variable ordinale à M modalités** : $M-1$ divisions
 - **Variable quantitative** prenant q valeurs distinctes : traitée comme une variable ordinale à q modalités, d'où $q-1$ divisions
- **Attention** :
variable binaire \neq division binaire selon une variable quelconque

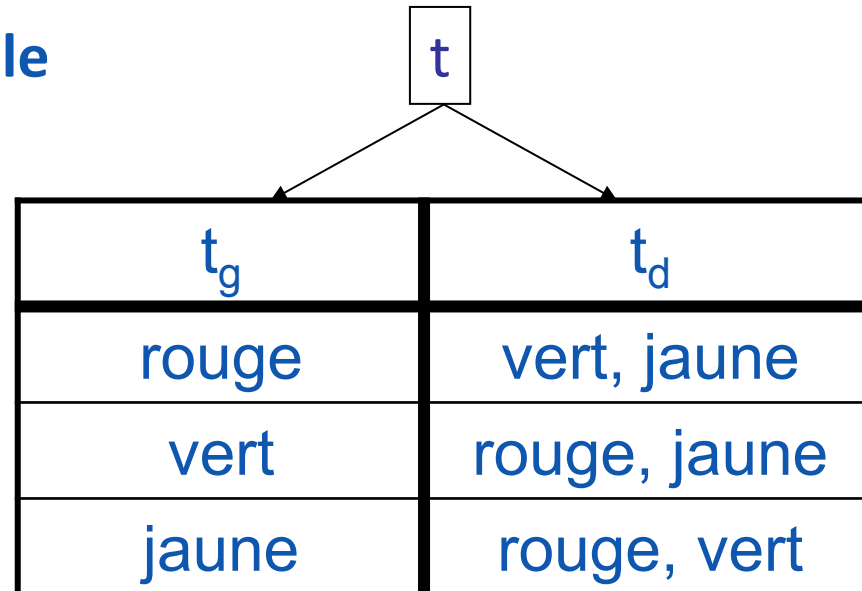
CART – 1. Divisions binaires

En effet, on peut avoir des arbres binaires où :

- Y binaire et X_1 à X_J quelconques
- Y qualitative à plusieurs modalités (classement) ou continue (régression) et X_1 à X_J quelconques

- **Exemple pour une variable nominale à 3 modalités :**

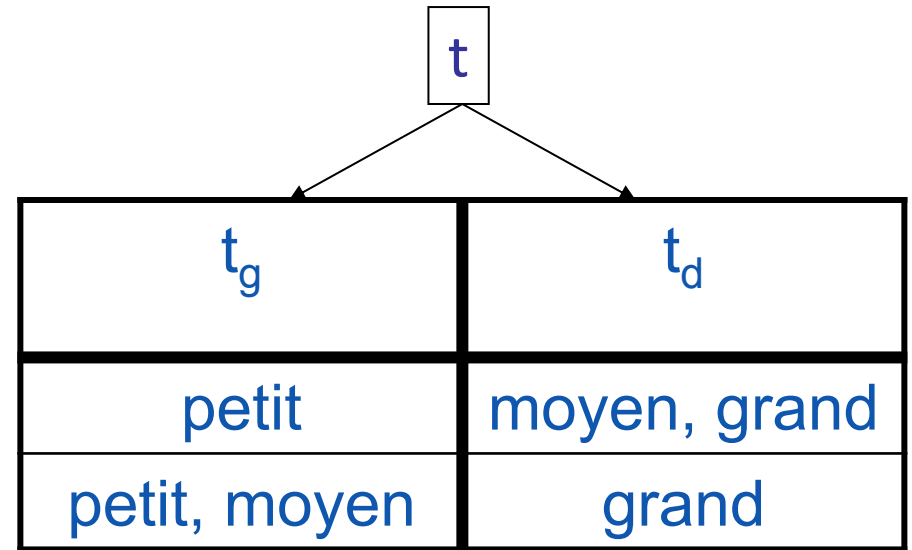
3 divisions binaires possibles d'un nœud t en deux nœuds t_g et t_d



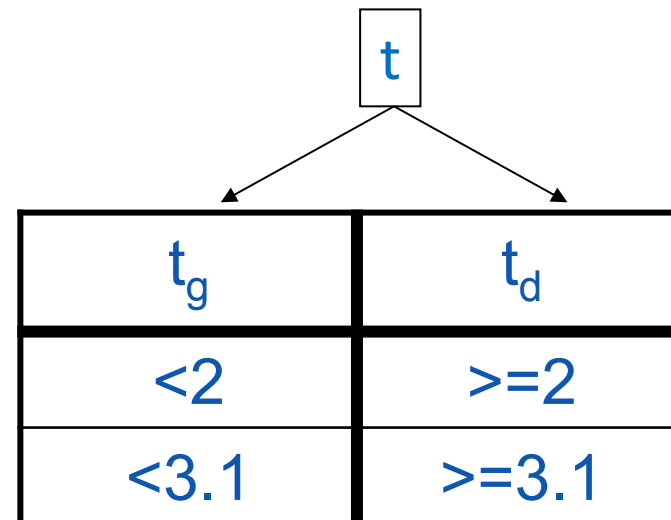
CART – 1. Divisions binaires

- **Exemple pour une variable ordinaire à 3 modalités :**

2 divisions binaires
possibles d'un nœud t en
deux nœuds t_g et t_d



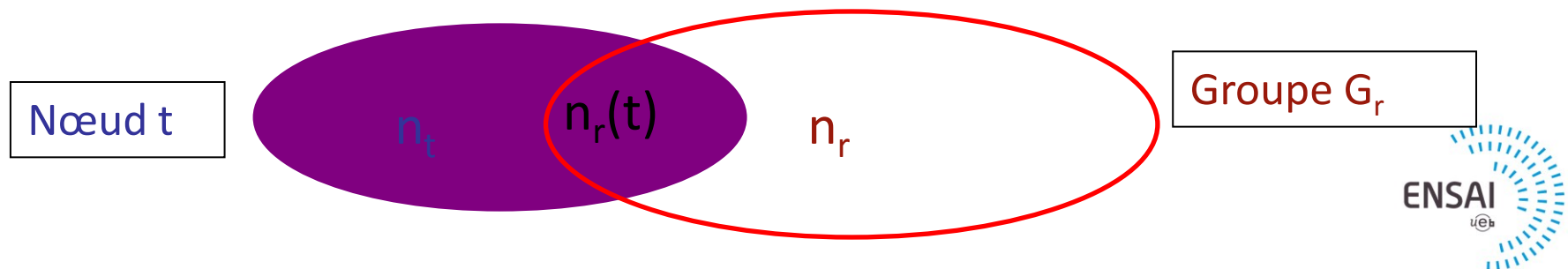
- **Exemple pour une variable continue à 3 valeurs distinctes que l'on ordonne : 1.5, 2, 3.1**
3-1 divisions binaires possibles
d'un nœud t en deux nœuds t_g
et t_d



CART – 2. Arbres de classement

La variable cible Y est une variable **nominale** (ou **ordinaire**) à R modalités, définissant R groupes d'individus $G_1, G_2, \dots, G_r, \dots, G_R$.

- Groupe noté r ou s
- $n(t)$: effectif du nœud t
- n_r : effectif du groupe G_r
- $n_r(t)$: effectif du nœud t appartenant à G_r



CART – 2. Arbres de classement

Rappel de l'idée de la méthode CART :

1. construire un arbre maximal
2. élaguer
3. définir l'arbre (optimal) fiable

CART – 2.1 Construire l'arbre maximal

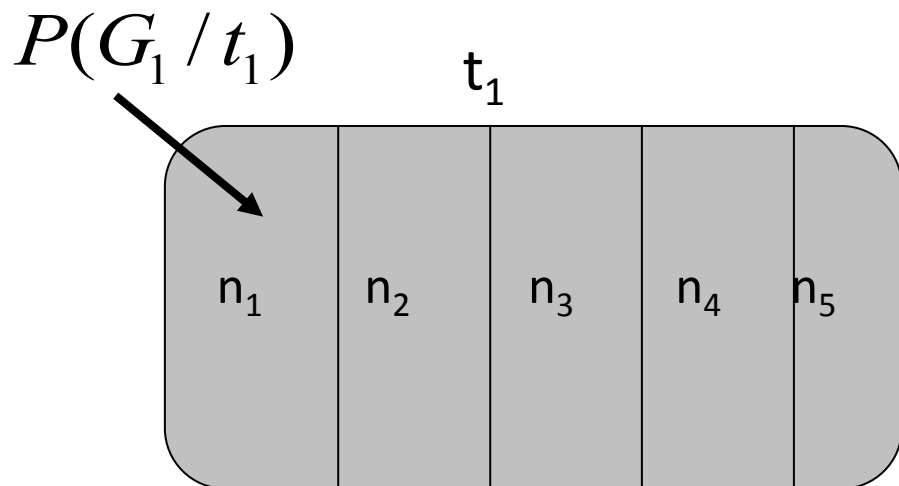
Principe de division des nœuds

- Un nœud est homogène, ou «relativement pur», si les individus du nœud appartiennent majoritairement à un groupe
- Inversement, un nœud est "impur" si les individus se répartissent uniformément dans tous les groupes
- Une division devra donc être telle que les nœuds-enfants soient plus "purs" que le nœud-parent : le mélange des groupes doit être moins important dans les nœuds-enfants que dans le nœud parent.

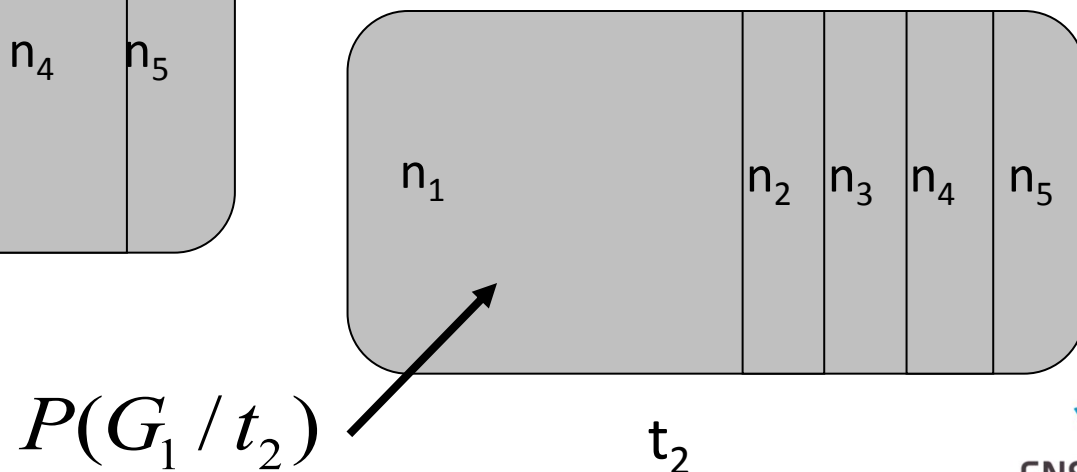
CART – 2.1 Construire l'arbre maximal

Mesure de l'impureté d'un noeud

- Utilisation de la probabilité $P(G_r / t)$



Exemple : Y a 5 modalités
 t_2 est plus homogène que t_1



CART – 2.1 Construire l'arbre maximal

On va mesurer l'impureté d'un nœud à partir d'une fonction qui vérifie certaines propriétés $\forall r = 1 \text{ à } R$:

- fonction non-négative des $P(G_r/t)$
- fonction concave des $P(G_r/t)$
- maximale quand $P(G_r/t)=1/R$
- minimale quand $P(G_r/t)=1$ et $P(G_s/t)=0$,
 $\forall s = 1 \text{ à } R \text{ avec } s \neq r$
- fonction symétrique des $P(G_r/t)$

CART – 2.1 Construire l'arbre maximal

Deux indicateurs d'impureté d'un nœud t :

- Entropie de Shannon

$$\begin{aligned} i(t) &= i[P(G_r / t), r = 1 \dots R] \\ &= - \sum_{r=1}^R P(G_r / t) \ln(P(G_r / t)) \end{aligned}$$

- Indice de diversité de Gini

$$\begin{aligned} i(t) &= i[P(G_r / t), r = 1 \dots R] \\ &= \sum_{r \neq s} P(G_r / t) P(G_s / t) = 1 - \sum_{r=1}^R P^2(G_r / t) \end{aligned}$$

CART – 2.1 Construire l'arbre maximal

Illustration des propriétés de l'indice de Gini dans le cas de 2 groupes

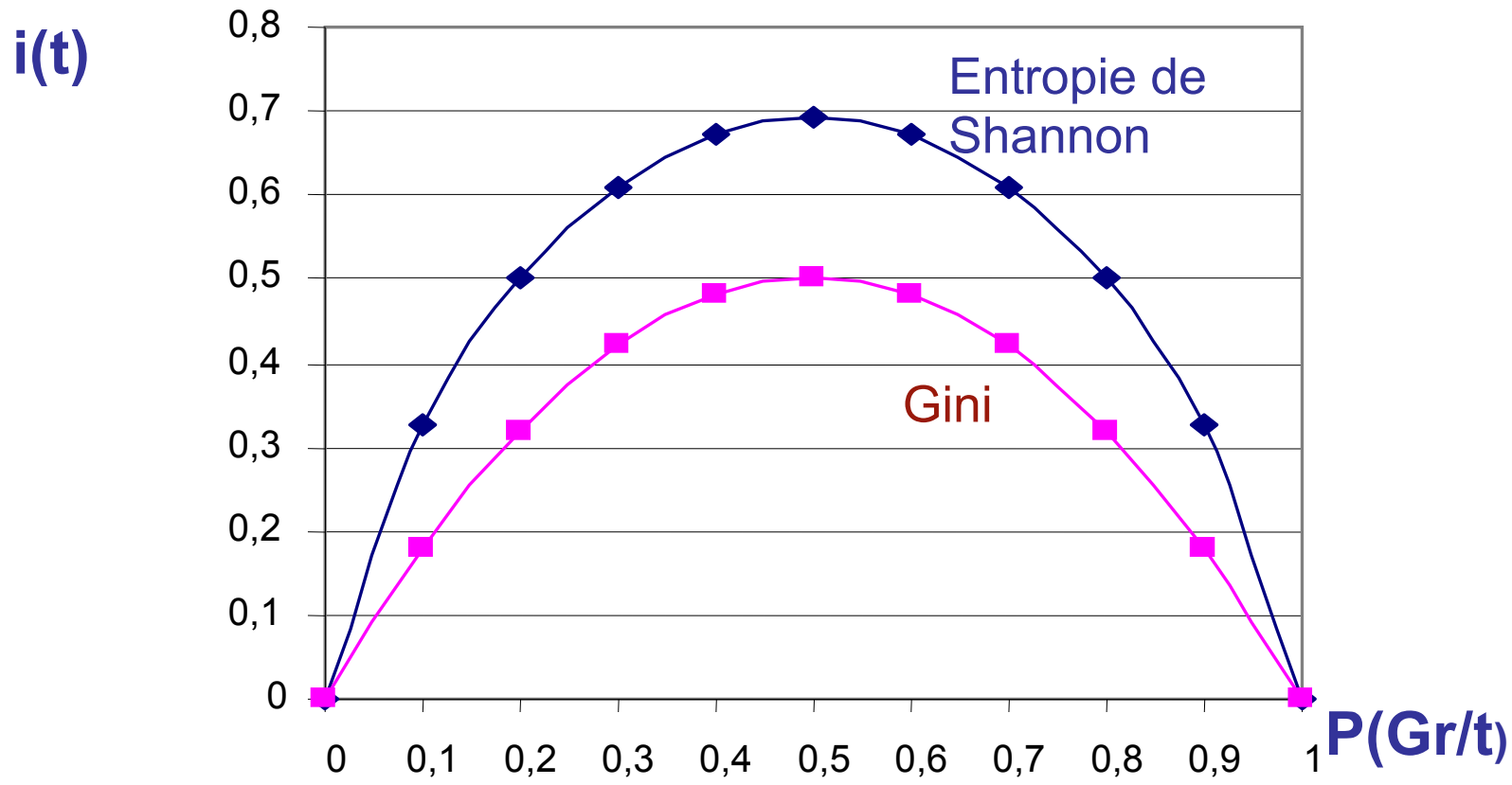
$$i(t) = 2 P(G_1/t) P(G_2/t) = 2p(1-p)$$

avec $p = P(G_1/t)$

- fonction non-négative des $P(G_r/t) \in [0;1]$
- $i(t)$ est minimale, et vaut 0, si $P(G_1/t) = 0$ ou $P(G_2/t) = 0$:
le nœud est pur
- $i(t)$ est maximale, et vaut $1/2$, si $P(G_1/t) = P(G_2/t) = 1/2$:
les groupes sont mélangés

CART – 2.1 Construire l'arbre maximal

Concavité et symétrie avec 2 groupes



CART – 2.1 Construire l'arbre maximal

- D'après la formule de Bayes :

$$P[G_r|t] = \frac{P[t|G_r]P[G_r]}{\sum_{s=1}^R P[t|G_s]P[G_s]} \longleftarrow = P(t)$$

1. Choisir les $P(G_r) = \pi_r$ probabilité a priori d'appartenir au groupe G_r
2. Estimer les $P(t/G_r)$

CART – 2.1 Construire l'arbre maximal

1. On peut considérer 3 possibilités pour les probabilités a priori :

- toutes égales : $\pi_r = \pi_s \quad \forall r, s = 1 \text{ à } R$
- égales aux fréquences empiriques $\pi_r = n_r / n$
- fixées par expertise antérieure

2. Estimer les $P(t/G_r)$

$$\hat{P}[t|G_r] = \frac{n_r(t)}{n_r}$$

CART – 2.1 Construire l'arbre maximal

On a alors :

$$P(t) = \sum_{s=1}^R P[t|G_s] P[G_s] = \sum_{s=1}^R \pi_s P[t|G_s]$$

$$\hat{P}(t) = \sum_{s=1}^R \pi_s \frac{n_s(t)}{n_s}$$

$$\hat{P}(G_r|t) = \frac{\pi_r \frac{n_r(t)}{n_r}}{\sum_{s=1}^R \pi_s \frac{n_s(t)}{n_s}}$$

CART – 2.1 Construire l'arbre maximal

Dans le cas particulier où l'on choisit les probabilités a priori égales aux fréquences empiriques $\pi_r = n_r/n$:

$$\hat{P}(G_r|t) = \frac{n_r}{n} \frac{n_r(t)}{n_r} \frac{n}{n(t)} = \frac{n_r(t)}{n(t)}$$

C'est donc la proportion d'individus du nœud t qui appartiennent au groupe r

CART – 2.1 Construire l'arbre maximal

Réduction de l'impureté d'un nœud par une division

- Chaque division ***div*** d'un nœud t en deux nœuds t_G et t_D provoque une réduction de l'impureté égale à $\Delta i(\mathbf{div}, t) \geq 0$
- Toute division d'un nœud entraîne une réduction positive ou nulle de l'impureté car on a une stricte concavité de la fonction d'impureté.

CART – 2.1 Construire l'arbre maximal

On note

p_G = proportion d'observations du nœud t
dans le nœud $t_G = n(t_G) / n(t)$

p_D = proportion d'observations du nœud t
dans le nœud $t_D = n(t_D) / n(t)$

$$i(t) = i(\hat{P}(G_r | t), r = 1 \dots R)$$

$$i(t_G) = i(\hat{P}(G_r | t_G), r = 1 \dots R)$$

$$i(t_D) = i(\hat{P}(G_r | t_D), r = 1 \dots R)$$

$$\Delta i(d, t) = i(t) - p_G i(t_G) - p_D i(t_D)$$

CART – 2.1 Construire l'arbre maximal

Rappel : si f est concave alors

$$a.f(u) + (1-a).f(v) \leq f[a.u + (1-a).v]$$

Où $a \in [0;1]$

On a donc

$$\begin{aligned} & p_G \cdot i(\hat{P}(G_r|t_G), r = 1 \dots R) + p_D \cdot i(\hat{P}(G_r|t_D), r = 1 \dots R) \\ & \leq i(p_G \cdot \hat{P}(G_r|t_G) + p_D \cdot \hat{P}(G_r|t_D), r = 1 \dots R) \\ & = i(\hat{P}(G_r|t), r = 1 \dots R) \end{aligned}$$

$$\Rightarrow i(\mathbf{t}) - p_G i(\mathbf{t}_G) - p_D i(\mathbf{t}_D) = \Delta i(\mathbf{d}, \mathbf{t}) \geq 0$$

Si $\forall r = 1 \dots R$, on a $\hat{P}(G_r|t_G) = \hat{P}(G_r|t_D) = \hat{P}(G_r|t)$

alors $\Delta i(\mathbf{d}, \mathbf{t}) = 0$

CART – 2.1 Construire l'arbre maximal

Sélection de la meilleure division d'un nœud

La meilleure division ***div***^{*} d'un nœud *t* est telle que la réduction de l'impureté est maximale

$$\mathbf{div}^* = \underset{div \in \mathcal{Div}}{\operatorname{argmax}} \Delta i(div, t)$$

où \mathcal{Div} est l'ensemble des divisions binaires possibles de *t*

= toutes les divisions binaires possibles de toutes les variables explicatives possibles

CART – 2.1 Construire l'arbre maximal

A chaque étape de division des noeuds de l'arbre, c'est ce critère ***div**** qui est utilisé pour choisir **quelle variable** et **quel seuil utiliser**.

- Construire l'arbre maximal = enchaîner des divisions binaires ...mais jusqu'où diviser ?
- Rappel : le nœud t est déclaré **terminal** dès la construction de l'arbre maximal si :
 - t est pur
 - ou t n'a pas de division admissible
 - ou t a un effectif $n(t) < \text{seuil}$

CART – 2.1 Construire l'arbre maximal

Affectation du segment t

- Objectif de l'analyse : sachant le profil d'un individu sur les variables explicatives, affecter cet individu à un groupe de Y
- En discrimination par arbre, on affecte un segment entier à un groupe (donc tous les individus du segment)
- A chaque étape, affectation possible des nœuds t à un groupe (notamment les nœuds terminaux de l'arbre maximal)

CART – 2.1 Construire l'arbre maximal

- Règle : Affectation du nœud t à G_r si

$$\hat{P}(G_r|t) > \hat{P}(G_s|t), \forall r, s = 1 \dots R \text{ avec } s \neq r$$

- Donc si on choisit les probabilités a priori égales aux fréquences empiriques $\pi_r = n_r/n$, la règle s'écrit :

$$\hat{P}(G_r|t) = \frac{n_r(t)}{n(t)} > \hat{P}(G_s|t) = \frac{n_s(t)}{n(t)}$$

⇒ **affectation du segment t au groupe le plus représenté dans t**

Noeud 0		
Catégorie	%	n
■ Bon payeur	48	1550
■ Mauvais payeur	52	1680
Total	100	3230

Type de contrat

CDI

CDD et autres contrats

Noeud 1		
Catégorie	%	n
■ Bon payeur	84	1330
■ Mauvais payeur	15	250
Total	49	1580

Noeud 2		
Catégorie	%	n
■ Bon payeur	13	220
■ Mauvais payeur	86	1430
Total	51	1650

Catégories d'âges

Catégories d'âges

Jeune (< 25)

Moyen (25-35); Agé (> 35)

Moyen (25-35); Jeune (< 25)

Agé (> 35)

Noeud 3		
Catégorie	%	n
■ Bon payeur	51	250
■ Mauvais payeur	48	240
Total	15	490

Noeud 4		
Catégorie	%	n
■ Bon payeur	99	1080
■ Mauvais payeur	1	10
Total	34	1090

Noeud 5		
Catégorie	%	n
■ Bon payeur	9	150
■ Mauvais payeur	90	1430
Total	49	1580

Noeud 6		
Catégorie	%	n
■ Bon payeur	100	70
■ Mauvais payeur	0	0
Total	2	70

CART – 2.1 Construire l'arbre maximal

- Ce premier critère d'affectation d'un segment t à un groupe r ne dépend que des $\hat{P}(G_r|t)$
Ce classement entraîne des erreurs de classement.
- On pourrait introduire des **coûts différentiels de mauvais classement** et les intégrer dans la règle d'affectation

CART – 2.1 Construire l'arbre maximal

Coût d'erreur d'affectation d'un segment

- On définit les coûts d'erreur de classement a priori, pour un individu :
 - $c(s|r)$ = coût de l'affectation d'un individu au groupe s alors qu'il appartient au groupe r
 - $c(s|s)=0$
 - Cas particulier : $c(s|r)=1$ si $s \neq r$

CART – 2.1 Construire l'arbre maximal

- On définit le **coût d'affectation** du segment **t** au groupe G_s

$$CM_s(t) = \sum_{r=1}^R c(s|r) \hat{P}(G_r|t)$$

⇒ Nouveau critère d'affectation = minimisation du coût d'affectation et non plus maximisation de $\hat{P}(G_r|t)$

CART – 2.1 Construire l'arbre maximal

⇒ Nœud t affecté au groupe G_s si s est égal à :

$$\mathit{Argmin}_{s=1 \text{ à } R} \sum_{r=1}^R c(s|r) \hat{P}(G_r|t)$$

- Une fois le nœud t affecté à un groupe, on définit le coût de mauvais classement dû au nœud t :

$$\hat{C}(t) = c(t) \hat{P}(t)$$

CART – 2.1 Construire l'arbre maximal

- **Coût de mauvais classement de l'arbre T** = somme des coûts de mauvais classement des segments terminaux de T

$$\hat{C}(T) = \sum_{t \in \tilde{T}} \hat{C}(t) = \sum_{t \in \tilde{T}} c(t) \hat{P}(t)$$

On note $\boxed{\tilde{T}}$ l'ensemble des nœuds terminaux de T

CART – 2.1 Construire l'arbre maximal

Si on décompose :

$$\hat{C}(T) = \sum_{s=1}^R \sum_{t \in \tilde{I}_s} \hat{P}(t) \sum_{r=1}^R c(s|r) \hat{P}(G_r|t)$$

\tilde{I}_s : segments terminaux de T affectés à G_s

$n_{s/r} = \sum_{t \in \tilde{I}_s} n_r(t)$: nombre d'individus affectés à G_s
mais qui appartiennent à G_r

$$\hat{C}(T) = \sum_{s=1}^R \sum_{r=1}^R \pi_r c(s|r) \frac{n_{s|r}}{n_r}$$

CART – 2.1 Construire l'arbre maximal

Dans le cas particulier où :

- $c(s|r)=1$ si $s \neq r$
- $\pi_r = n_r / n$

$$\hat{C}(T) = \sum_{s=1}^R \sum_{\substack{r=1 \\ r \neq s}}^R \frac{n_{s|r}}{n_r}$$

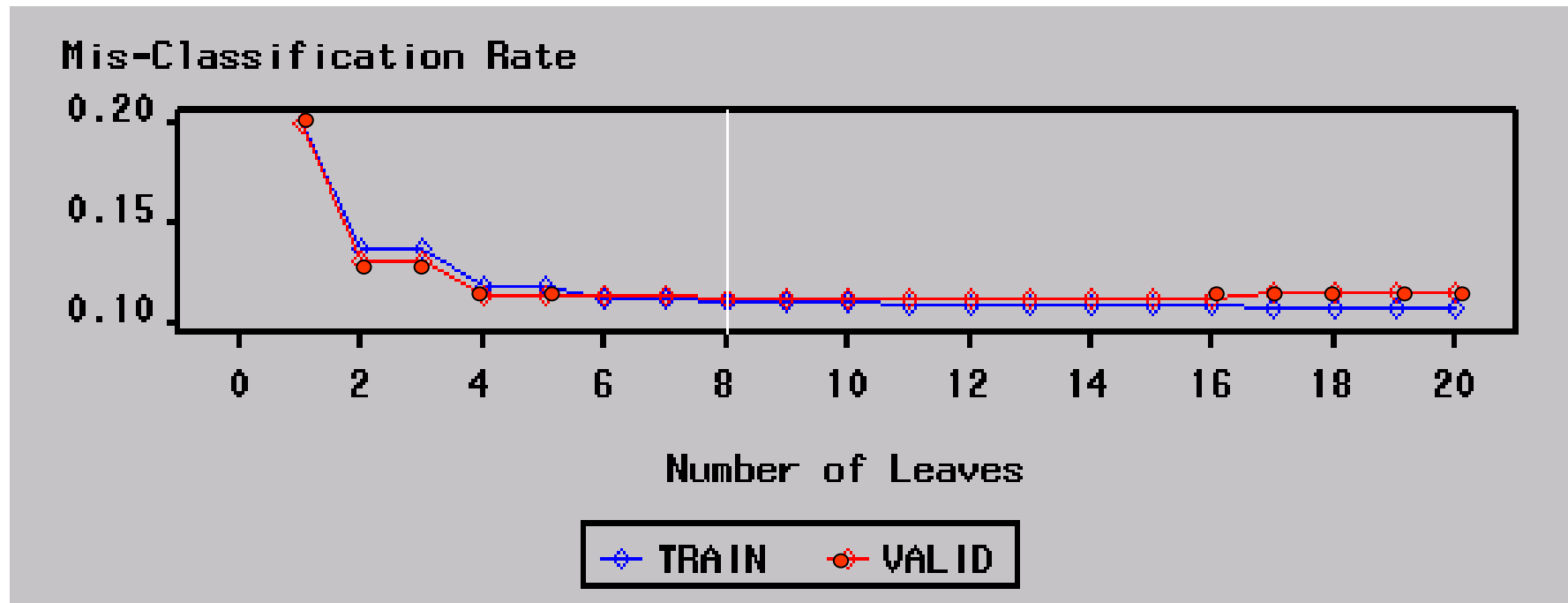
Le coût estimé des mauvais classements de l'arbre T est alors égal à la proportion d'observations mal classées par l'arbre T

CART – 2.1 Construire l'arbre maximal

- La construction de l'arbre maximal se fait par réduction de l'impureté à chaque division :
le coût de l'arbre maximal estimé par resubstitution, donc sur l'échantillon d'*apprentissage* , décroît au fur et à mesure de sa construction ...
- Mais le coût estimé par validation simple ou validation croisée décroît puis stagne puis remonte au fur et à mesure que l'arbre max s'allonge =
effet de sur-ajustement ou sur-apprentissage,

CART – 2.1 Construire l'arbre maximal

- Train = résultat pour échantillon d'apprentissage
- Valid = résultat pour échantillon d'élagage



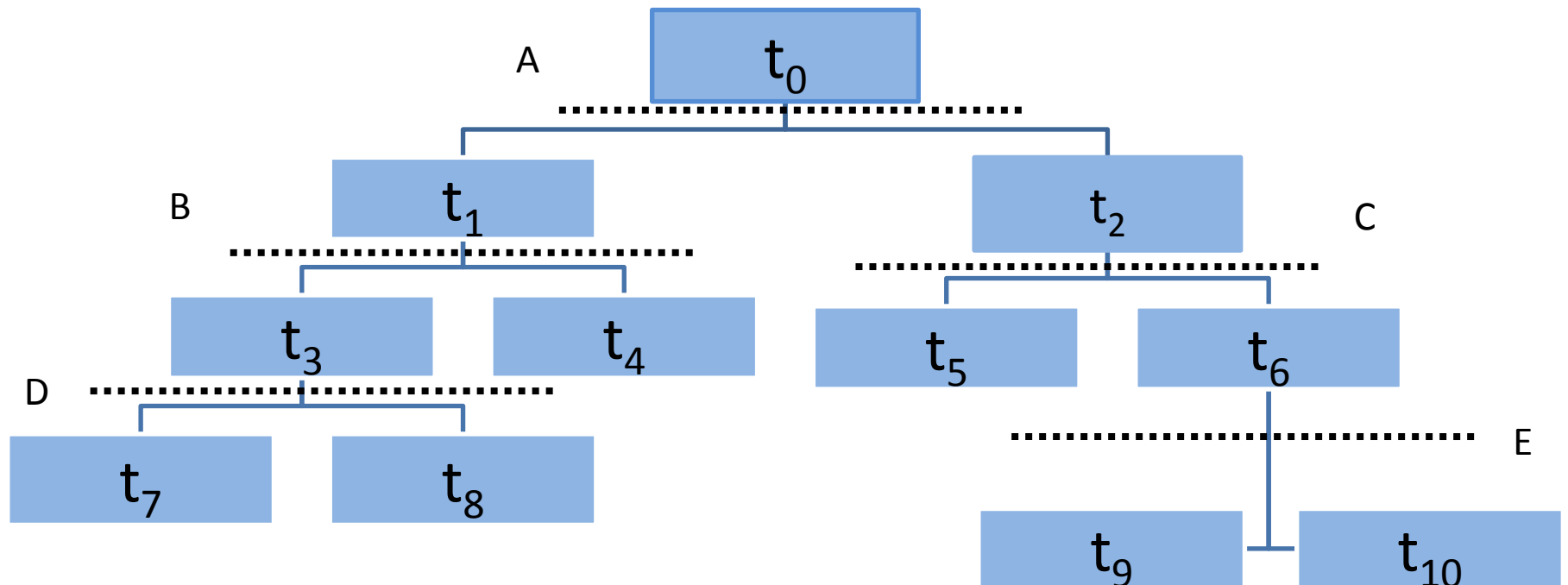
CART – 2.2 Elaguer l'arbre maximal

- L'arbre maximal est trop long :
les dernières divisions sont instables et peu fiables
- On va donc l'élaguer : couper les branches les moins informatives et les plus longues

L'élagage (ou pruning)

- Elaguer consiste à enlever tous les nœuds descendants d'un nœud $t \Rightarrow t$ devient nœud terminal
- Soit T^t la branche issue du nœud t , on a $T^t =$ descendants de t
- On note l'arbre élagué $= T - T^t$

CART – 2.2 Elaguer l'arbre maximal

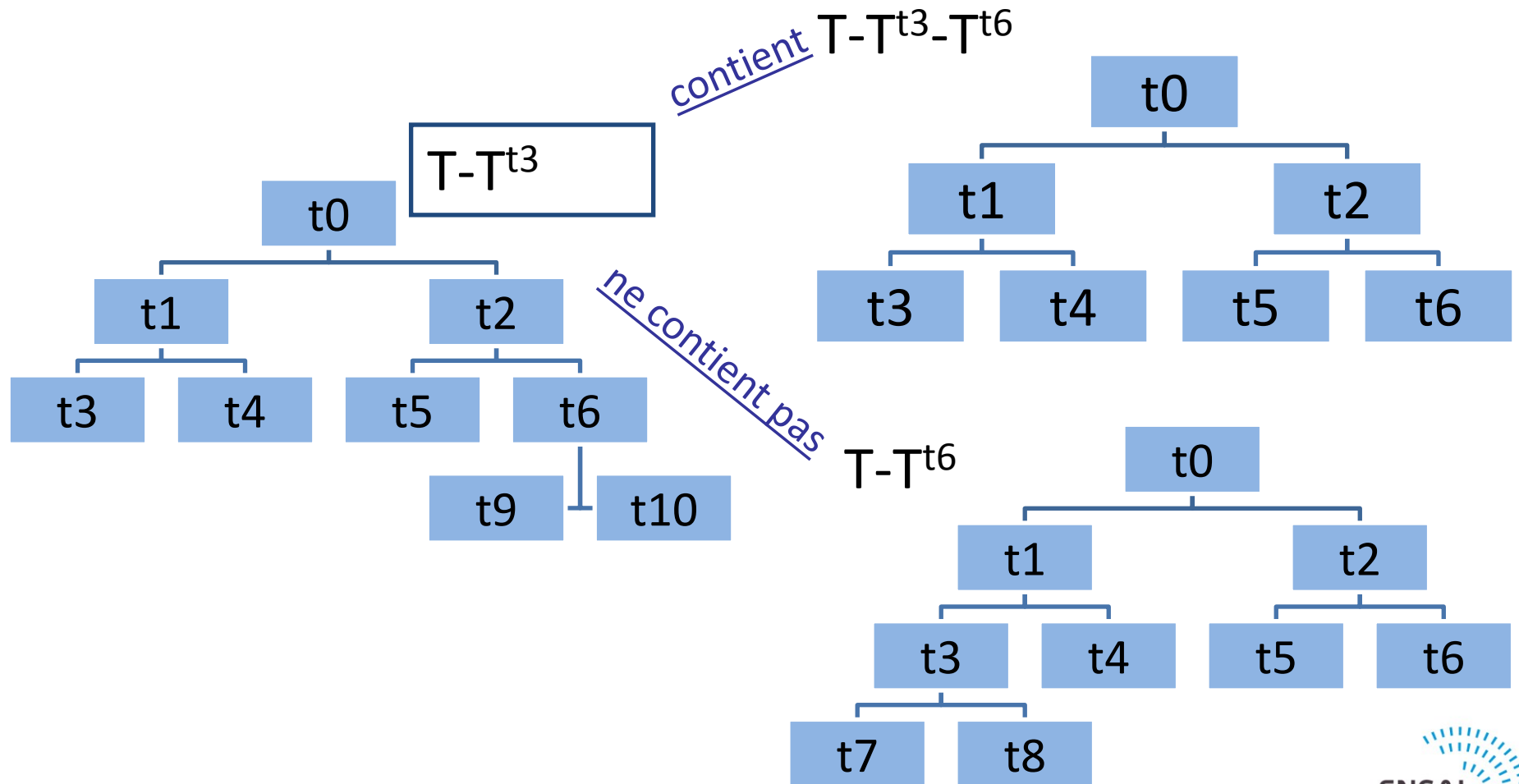


CART – 2.2 Elaguer l'arbre maximal

- Dans l'exemple précédent l'ensemble des sous-arbres possibles de T est :

$T-T^{t3}$: 5 nœuds terminaux - coupure en D
$T-T^{t3}-T^{t6}$: 4 nœuds terminaux - coupure en D&E
$T-T^{t3}-T^{t2}$: 3 nœuds terminaux - coupure en D&C
$T-T^{t1}$: 4 nœuds terminaux - coupure en B
$T-T^{t1}-T^{t6}$: 3 nœuds terminaux - coupure en B&E
$T-T^{t1}-T^{t2}$: 2 nœuds terminaux - coupure en B&C
$T-T^{t6}$: 5 nœuds terminaux - coupure en E
$T-T^{t2}$: 4 nœuds terminaux - coupure en C
$T-T^{t0}$: racine $\{t_0\}$ - coupure en A

CART – 2.2 Elaguer l'arbre maximal



CART – 2.2 Elaguer l'arbre maximal

- L'élagage produit des sous-arbres T_h
- Ces sous-arbres ont des coûts $C(T_h)$ et des nombres de segments terminaux $|\tilde{I}_n|$ différents
- Quel sous-arbre choisir parmi tous les sous-arbres possibles ?

CART – 2.2 Elaguer l'arbre maximal

Première piste :

Soit T_{max} à L nœuds terminaux

- On construit la séquence de sous-arbres $\{ T_{max}, T_1, T_2, \dots, \{t_0\} \}$
- A chaque valeur de H , $0 \leq h < L-1$, correspond la classe CL_h des sous-arbres à $L-h$ nœuds terminaux.
- Pour chaque valeur de h , on sélectionne dans CL_h le sous-arbre ayant le coût estimé minimum

CART – 2.2 Elaguer l'arbre maximal

Inconvénient de cette méthode:

- La séquence de sous-arbres n'est pas emboîtée :
 T_{h+1} n'est pas nécessairement un sous-arbre de T_h et des branches coupées peuvent réapparaître ensuite.
- Nombre d'arbres possible vite énorme

CART – 2.2 Elaguer l'arbre maximal

Breiman et al. (1984) – méthode CART

- Méthode en deux étapes :
 1. Construction d'une séquence de sous-arbres emboîtés de coût-complexité minimum
 2. Sélection du sous-arbre de la séquence le plus fiable par validation simple ou croisée

CART – 2.2 Elaguer l'arbre maximal

2.2.1 Séquence d'élagage

Etape 1 : séquence d'élagage

- Établir une séquence d'élagage à partir de la notion de coût-complexité :
 - Séquence d'arbres emboîtés
 - Arbres de coût-complexité croissant
 - Séquence telle que chaque sous-arbre a le plus petit coût estimé par resubstitution parmi les sous-arbres de même taille

CART – 2.2 Elaguer l'arbre maximal

2.2.1 Séquence d'élagage

- **La notion de coût-complexité**
pénalisation de la complexité de l'arbre
en utilisant le nombre de nœuds terminaux

On note

$|\tilde{I}|$: nb de segments terminaux de T

α : paramètre de complexité

$\hat{C}^{res}(t)$: coût estimé par resubstitution sur
l'échantillon d'apprentissage

$$C_{\alpha}(T) = \hat{C}^{res}(T) + \alpha |\tilde{I}|$$

CART – 2.2 Elaguer l'arbre maximal

2.2.1 Séquence d'élagage

- Obtention de T_1
 - On part de T_{max}

$$C_\alpha(T_{max}) = \hat{C}^{res}(T_{max}) + \alpha |\tilde{T}_{max}|$$

- En posant $\alpha=0$, on a

$$C_\alpha(T_{max}) = \hat{C}^{res}(T_{max})$$

- T_{max} est l'arbre de coût-complexité minimum quand $\alpha=0$
- En général, $T_1 = T_{max}$

CART – 2.2 Elaguer l'arbre maximal

2.2.1 Séquence d'élagage

- **Obtention de T_1 :**

On effectue un « premier nettoyage »

- Pour tout segment intermédiaire t de T_1 donnant 2 segments terminaux, on a :

$$\hat{C}^{res}(t) \geq \hat{C}^{res}(t_G) + \hat{C}^{res}(t_D)$$

- En cas d'égalité, on supprime la division du segment t : il devient terminal.

CART – 2.2 Elaguer l'arbre maximal

2.2.1 Séquence d'élagage

- T_1 est donc le sous-arbre de T_{max} ayant le même coût que T_{max} :

$$\hat{C}^{res}(T_1) = \hat{C}^{res}(T_{max})$$

- Pour tout segment intermédiaire t de T_1 on a

$$\hat{C}^{res}(t) > \hat{C}^{res}(T_1^t)$$

où T_1^t est la branche issue du segment t

- On pose $\alpha_1=0$

CART – 2.2 Elaguer l'arbre maximal

2.2.1 Séquence d'élagage

- **Obtention de T_2**

On obtient T_2 par élagage de T_1

Pour chaque nœud intermédiaire t de T_1 on a

- Coût-complexité d'1 nœud t :

$$\hat{C}_\alpha(t) = \hat{C}^{res}(t) + \alpha$$

- Coût-complexité de la branche issue de t :

$$\hat{C}_\alpha(T_1^t) = \hat{C}^{res}(T_1^t) + \alpha |\tilde{I}_1|$$

CART – 2.2 Elaguer l'arbre maximal

2.2.1 Séquence d'élagage

- Tant qu'on a

$$\hat{C}_{\alpha}(T_1^t) < \hat{C}_{\alpha}(t)$$

la branche a 1 coût-complexité plus faible que le nœud t et T_1 est préférable à $T_2 = T_1 - T_1^t$

- Pour $\alpha=0$, cette inégalité est toujours vérifiée... donc pas d'élagage !
- Pour pouvoir élaguer l'arbre, il va donc falloir augmenter progressivement alpha

CART – 2.2 Elaguer l'arbre maximal

2.2.1 Séquence d'élagage

- On augmente progressivement α , jusqu'à ce que les **2 coûts-complexité deviennent égaux pour un nœud intermédiaire t^* de T_1** :
 - \Rightarrow nœud t^* plus court que branche pour un même coût-complexité
 - \Rightarrow nœud t^* préférable à la branche
- Avec $t = t^*$, α doit donc vérifier l'égalité :

$$\hat{C}_\alpha(T_1^t) = \hat{C}_\alpha(t) \Leftrightarrow \alpha = \frac{\hat{C}^{res}(t) - \hat{C}^{res}(T_1^t)}{|\tilde{I}_1| - 1}$$

CART – 2.2 Elaguer l'arbre maximal

2.2.1 Séquence d'élagage

- On définit la fonction :

$$\begin{cases} g_1(t) = \frac{\hat{C}^{res}(t) - \hat{C}^{res}(T_1^t)}{|\tilde{T}_1^t| - 1} & \text{si } t \notin \tilde{T}_1^t \\ g_1(t) = +\infty & \text{si } t \in \tilde{T}_1^t \end{cases}$$

- On cherche t^* , parmi tous les segments intermédiaires, tel que :

$$t^* = \underset{t \in \tilde{T}_1^t \quad (t \in \tilde{T}_1^t)}{\text{Arg min}} g_1(t)$$

CART – 2.2 Elaguer l'arbre maximal

2.2.1 Séquence d'élagage

Interprétation de $\min \frac{\hat{C}^{res}(t) - \hat{C}^{res}(T_1^t)}{|\tilde{T}_1^t| - 1}$

- On cherche parmi tous les segments intermédiaires de T_1 , celui pour lequel :
 - le gain en termes de coût obtenu par la branche qui en découle par rapport au segment seul $\hat{C}^{res}(t) - \hat{C}^{res}(T_1^t)$
 - est le plus petit, compte tenu de la longueur de cette branche $|\tilde{T}_1^t| - 1$

CART – 2.2 Elaguer l'arbre maximal

2.2.1 Séquence d'élagage

- On coupe alors la branche correspondant à t^* , et on pose :

$$\alpha_2 = g_1(t^*)$$

- L'arbre $T_2 = T_1 - T_1^{t^*}$ est préférable à T_1 pour un niveau

$$\alpha_2 > \alpha_1 (= 0)$$

CART – 2.2 Elaguer l'arbre maximal

2.2.1 Séquence d'élagage

- **Obtention de T_3 par élagage de T_2**
 - Parmi tous les segments intermédiaires de T_2 , on choisit celui (t^*) qui minimise $g_2(t)$

$$g_2(t) = \frac{\hat{C}^{res}(t) - \hat{C}^{res}(T_2^t)}{|\tilde{T}_2^t| - 1} \quad \text{si } t \notin \tilde{T}_2^t$$

et on pose

$$\alpha_3 = g_2(t^*)$$

Etc... jusqu'à obtention d'un sous-arbre à un seul segment terminal = racine de T_{max}

CART – 2.2 Elaguer l'arbre maximal

2.2.1 Séquence d'élagage

- Récapitulatif de construction de la séquence

- $T_0 = T_{max}$
- $T_1 = T_0 - \text{branches} /$
- $T_2 = T_1 - \text{branche(s)} /$
- $T_3 = T_2 - \text{branche(s)} /$
- ...
- $T_r = \text{racine}$

$$\hat{C}(t) = \hat{C}(t_G) + \hat{C}(t_D)$$

$$\min \left[\alpha(t) = \frac{\hat{C}^{res}(t) - \hat{C}^{res}(T_1^t)}{|\tilde{T}_1^t| - 1} \right]$$
$$\min \left[\alpha(t) = \frac{\hat{C}^{res}(t) - \hat{C}^{res}(T_2^t)}{|\tilde{T}_2^t| - 1} \right]$$

CART – 2.2 Elaguer l'arbre maximal

2.2.1 Séquence d'élagage

- Par élagage successif, on a obtenu une séquence de sous-arbres emboîtés

$$S_T = \{T_1 \succ \dots \succ \dots \succ \{t_0\}\}$$

- de niveau croissant de pénalisation de la complexité :

$$S_\alpha = \{\alpha_1 = 0 < \alpha_2 < \dots < \alpha_h < \dots < \alpha_{\{t_0\}}\}$$

- Chaque sous-arbre T_h de la séquence est, parmi tous les sous-arbres de T_{max} ayant $|\tilde{I}_{n}|$ segments terminaux, le sous-arbre dont le coût estimé par **resubstitution** est le plus faible.

CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – validation simple

Etape 2 : Choix du meilleur sous-arbre

A - Méthode de la validation simple avec l'échantillon d_{γ} :

1. Application des règles d'affectation issues des sous-arbres T_h de la séquence S_T à l'échantillon d_{γ}
2. Estimation des coûts d'erreur de généralisation

$$S_C = \{\hat{C}^{vs}(T_1), \hat{C}^{vs}(T_2), \dots, \hat{C}^{vs}(T_h), \dots, \hat{C}^{vs}(\{t_0\})\}$$

CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – validation simple

- Première idée : choisir avec S_C le sous-arbre T_{h_0} tel que

$$\hat{C}^{vs}(T_{h_0}) = \min[\hat{C}^{vs}(T_h); T_h \in S_T]$$

- En pratique, on observe une décroissance de $\hat{C}^{vs}(T_{h_0})$ en fonction du nombre de segments terminaux, puis une zone stable, puis une remontée.

- Breiman propose de choisir arbre T_h tel que T_h est le plus petit sous-arbre vérifiant :

$$\hat{C}^{vs}(T_h) < \hat{C}^{vs}(T_{h_0}) + e.t. [\hat{C}^{vs}(T_{h_0})]$$

Avec $e.t.$ = écart-type et T_{h_0} défini avec la première idée.

CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – validation simple

$$\hat{C}^{vs}(T) = \sum_{\substack{s=1 \\ s \neq r}}^R \sum_{r=1}^R \frac{n^{dv}_{s/r}}{n^{dv}}$$

est une fréquence empirique d'individus mal classés dans un échantillon de validation de taille n^{dv}

Loi binomiale ***Bin***(n^{dv}, p) d'où :

$$\widehat{e.t.}[\hat{C}^{vs}(T)] = \sqrt{\frac{\hat{C}^{vs}(T)[1 - \hat{C}^{vs}(T)]}{n^{dv}}}$$

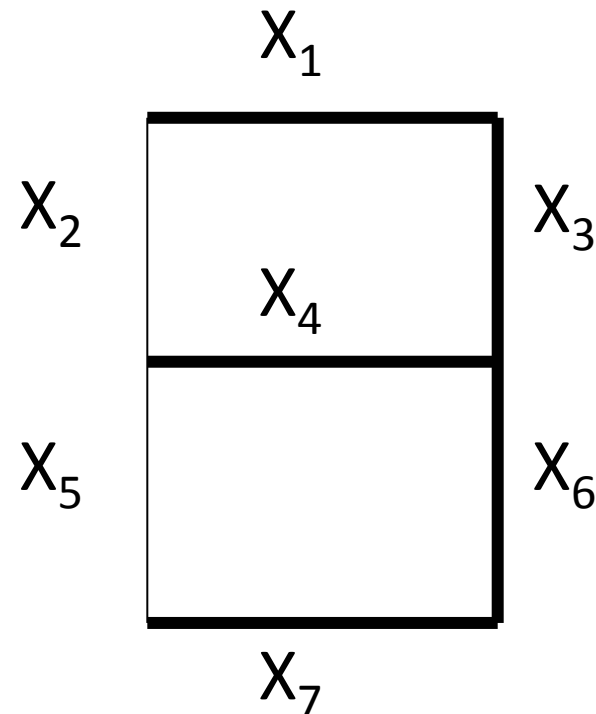
CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – validation simple

Exemple sur les données DIGIT

(voir Nakache et Confais, 2003)

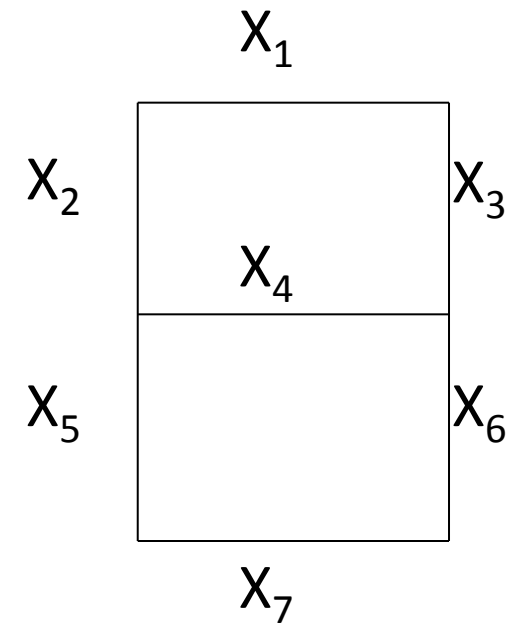
- Reconnaissance de 10 chiffres (Y) à partir de 7 digits (binaires)
- Exemple :
on obtient $Y=3$
avec $X_2=X_5=0$
et les autres $X_j=1$



CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – validation simple

Y	X_1	X_2	X_3	X_4	X_5	X_6	X_7
1	0	0	1	0	0	1	0
2	1	0	1	1	1	0	1
3	1	0	1	1	0	1	1
4	0	1	1	1	0	1	0
5	1	1	0	1	0	1	1
6	1	1	0	1	1	1	1
7	1	0	1	0	0	1	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1
0	1	1	1	0	1	1	1



Dans ce tableau, on a des données sans erreur

CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – validation simple

- Les données sont obtenues par simulation :
 - échantillon de base = 200 observations,
 - échantillon-test = 2000 observations
- On a inversé certains digits de façon aléatoire = des erreurs de reconnaissance de forme sont possibles
- Chaque modalité de Y apparaît 20 fois dans l'échantillon de base et 200 fois dans l'échantillon de validation d_v (représentation équilibrée)

CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – validation simple

On choisit

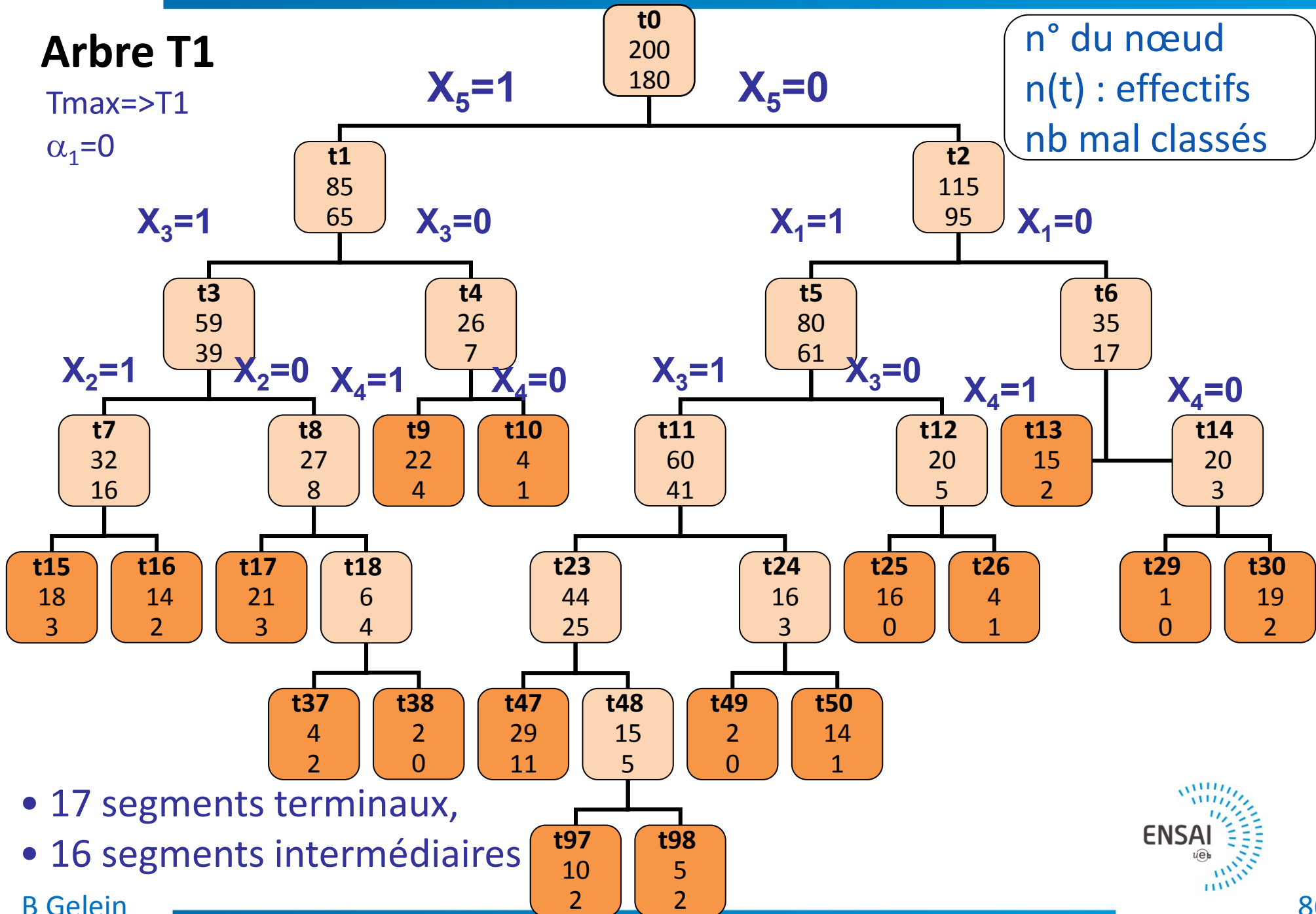
- Indice d'impureté = Gini,
- coûts de mauvais classement égaux
- probabilités a priori d'appartenance aux groupes proportionnelles aux effectifs de l'échantillon de base
⇒ coût de l'arbre = taux de mal classés dans segments terminaux
- On obtient T_{\max} avec 36 segments terminaux

Arbre T1

$T_{\max} \Rightarrow T1$

$\alpha_1 = 0$

n° du nœud
n(t) : effectifs
nb mal classés



CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – validation simple

Elagage de T_1

- $MC(t)$ = nombre de mal classés dans le segment intermédiaire t
- $MC(T_1^t)$ = nombre de mal classés dans le sous-arbre issu de t
- On recherche le plus petit $\alpha(t) > \alpha_1 = 0$
 $\alpha_2 = 0.0050$

t	MC(t)	MC(T_t)	$ \tilde{T}_t $	$\alpha(t)$
0	180	36	17	0.0450
1	65	15	7	0.0417
2	95	21	10	0.0411
3	39	10	5	0.0362
4	7	5	2	???
5	61	17	7	0.0366
6	17	4	3	0.0600
7	16	5	2	0.0550
8	8	5	3	0.0075
11	41	16	5	0.0321
12	5	1	2	0.0200
14	3	2	2	0.0050
18	4	2	2	0.0100
23	25	15	3	0.0250
24	3	1	2	0.0100
48	5	4	2	0.0050

CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – validation simple

Si on applique la formule au nœud $t=2$:

$$\alpha = \frac{\hat{C}^{res}(t) - \hat{C}^{res}(T_1^t)}{|\tilde{I}_1| - 1} \Rightarrow \alpha(2) = \frac{\frac{95}{200} - \frac{21}{200}}{10 - 1} = 0.0411$$

Ce sont les nœuds 14 et 48 qui présentent la plus petite valeur de $\alpha(t) > \alpha_1 : 0.005$

\Rightarrow on coupe les branches T_1^{t14} et T_1^{t48}

\Rightarrow on pose $\alpha_2 = 0.005$

CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – validation simple

Avec cet élagage de T_1 , on obtient l'arbre T_2 :

$$T_2 = T_1 - T_1^{t14} - T_1^{t48}$$

15 segments terminaux

14 segments intermédiaires

$$\hat{C}^{res}(T_2) = \frac{38}{200} = 0.19$$

Bilan :

2 mal classés en plus

2 segments terminaux en moins

CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – validation simple

Elagage de T_2

- On recherche le plus petit $\alpha(t) > \alpha_2$
 $\alpha_3 = 0.0075$
- On supprime la branche issue de t_8
- $T_3 = T_2 - T_2^{t_8}$
- Bilan : + 3 MC,
-2 segments terminaux

$$\hat{C}^{res}(T_3) = \frac{41}{200} = 0.205$$

t	MC(t)	MC(T_2^t)	$ \tilde{T}_{2^t} $	$\alpha(t)$
0	180	38	15	0.0507
1	65	15	7	0.0417
2	95	23	8	0.0514
3	39	10	5	0.0362
4	7	5	2	0.0100
5	61	18	6	0.0430
6	17	5	2	0.0600
7	16	5	2	0.0550
8	8	5	3	0.0075
11	41	17	4	0.0400
12	5	1	2	0.0100
18	4	2	2	0.0100
23	25	16	2	0.0450
24	3	1	2	0.0100

CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – validation simple

- Élagages successifs pour arriver à $T_{12} = 1$ segment terminal = racine

t	MC(t)	MC(T_{12}^t)	$ \tilde{T}_t $	$\alpha(t)$
1	180	160	2	0.1

$$\hat{C}^{res}(T_{12}) = \frac{180}{200} = 0.90$$

- Séquence de sous-arbres emboîtés :

$$T_1 \supset \dots \supset T_{12}$$

$$\alpha_1 < \alpha_2 < \alpha_3 < \dots < \alpha_{12}$$

CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – validation simple

Arbre T	Nombre segments terminaux	Coût (resubstitution)	Coût relatif *	Paramètre de complexité
1	17	0.180	0.200	0.0000
2	15	0.190	0.211	0.0050
3	13	0.205	0.228	0.0075
4	10	0.235	0.261	0.0100
5	9	0.280	0.311	0.0450
6	8	0.335	0.372	0.0550
7	7	0.395	0.439	0.0600
8	6	0.460	0.511	0.0650
9	5	0.535	0.594	0.0750
10	3	0.705	0.783	0.0850
11	2	0.800	0.888	0.0950
12	1	0.900	1.000	0.1000

* Coût relatif de T = Coût (resubstitution) de T / Coût (resubstitution) de T12



CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – validation simple

- Choisir l'arbre optimal dans la séquence à l'aide des coûts estimés par échantillon-test ou validation croisée
- Par échantillon-test (rappel) :
 - choisir le sous-arbre T_{h_0} tel que
$$\hat{C}^{vs}(T_{h_0}) = \min[\hat{C}^{vs}(T_h); T_h \in S_T]$$
 - choisir arbre T_h tel que T_h est le plus petit sous-arbre vérifiant

$$\hat{C}^{vs}(T_h) < \hat{C}^{vs}(T_{h_0}) + \widehat{e.t.}[\hat{C}^{vs}(T_{h_0})]$$

CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – validation simple

On a :

$$\hat{C}^{vs}(T_{h_0}) = \hat{C}^{vs}(T_2) = 0.2979$$

En appliquant la règle d'un écart-type :

$$\hat{C}^{vs}(T_{h_0}) + \hat{e.t.}[\hat{C}^{vs}(T_{h_0})] = 0.2979 + 0.0105 = 0.3084$$

Le plus petit arbre dont le coût
est inférieur à 0.3084 :

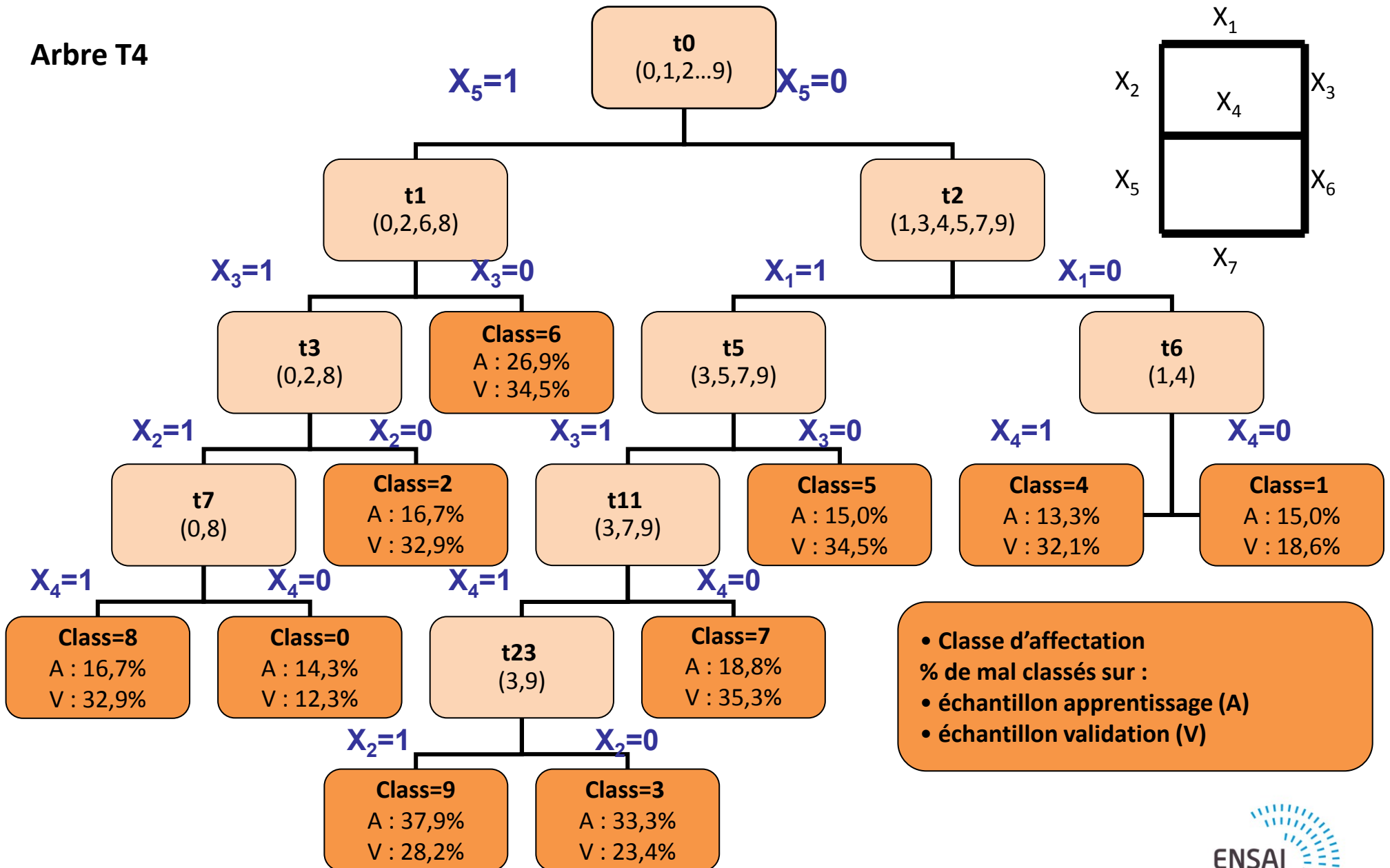
T₄ (10 nœuds terminaux)

CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – validation simple

Arbre T	Nombre de segments terminaux	Coût relatif Ech. Val.	Coût Ech val.	Coût Ech val. + écart-type
1	17	0,338	0,304	
2	15	0,331	0,298	0,308
3	13	0,338	0,304	
4	10	0,334	0,301	
5	9	0,401	0,361	
...	

Arbre T4



CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – Valid. croisée

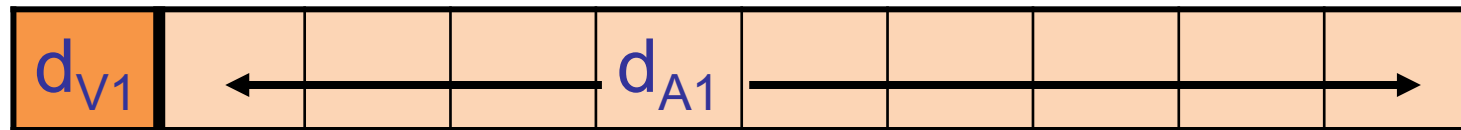
Méthode de la validation croisée :

- On détermine la séquence de sous-arbres emboîtés de coût-complexité minimum de T_{\max} à partir de l'échantillon \mathbf{d}_n : $S_T = \{T_1, T_2, \dots, T_L\}$
- A cette séquence correspondent les 2 autres séquences
$$S_{\hat{C}} = \{\hat{C}(T_1), \hat{C}(T_2), \dots, \hat{C}(T_h), \dots, \hat{C}(\{t_0\})\}$$
$$S_{\alpha} = \{\alpha_1 = 0 < \alpha_2 < \dots < \alpha_h < \dots < \alpha_{\{t_0\}}\}$$
- On va estimer la séquence de coûts S_C par validation croisée

CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – Valid. croisée

- L'échantillon d_n est divisé en K sous-ensembles (souvent $K=10$) ce qui permet de former K couples d'échantillons (d_{A_k}, d_{V_k}) avec $k= 1$ à K .
 - $d_{V1} \dots d_{VK}$ sous-échantillons de taille = $\text{card}(d_n)/K$ (exemple = $1/10$ de d_n)
 - Et K sous- échantillons complémentaires :
 $d_{A1} \dots d_{AK}$ (de taille = $9/10$ de d_n)



CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – Valid. croisée

- Pour chaque échantillon d_{Ak} (9/10 de d_n), $k= 1$ à K
 - On construit un arbre maximal
 - On élague selon la méthode du coût-complexité
 - On obtient la séquence

$$S_T^k = \{T_1^k, T_2^k \dots T_{l_k}^k\}$$

- On estime le coût de chaque sous-arbre de cette séquence sur chaque échantillon de validation d_{vk} (1/10 de d)

$$S_C^k = \{\hat{C}_1^k, \hat{C}_2^k \dots \hat{C}_{l_k}^k\}$$

CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – Valid. croisée

- **Objectif de la validation croisée :**
estimer les coûts associés aux sous-arbres de S_T à partir des coûts estimés des sous-arbres des K séquences $S_{T_k}, k=1 \text{ à } K$
- A noter : le nombre de sous-arbres obtenus peut être différent d'une séquence S_{T_k} à l'autre

CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – Valid. croisée

- Pour chaque paire d'échantillons (d_{A_k} , d_{V_k}) avec $k = 1$ à K on a :
 - Deux séquences construites et calculées sur un échantillon d'apprentissage d_{A_k}

$$S_T^k = \{T_1^k, T_2^k \dots T_{l_k}^k\}$$

$$S_\alpha^k = \{\alpha_1^k, \alpha_2^k, \dots, \alpha_{l_k}^k\}$$

- Une séquence estimée sur échantillon de validation d_{V_k}

$$S_{\hat{C}}^k = \{\hat{C}_1^k, \hat{C}_2^k \dots \hat{C}_{l_k}^k\}$$

CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – Valid. croisée

- On prend dans chaque séquence S_{T_k} le sous-arbre $T_{h'}^k$ le plus proche de T_h de S_T en terme de coût-complexité (i.e. de α_h)
- Méthode :

$$\alpha_h \text{ de } T_h \quad \Rightarrow \alpha'_h = \sqrt{\alpha_h \alpha_{h+1}}$$
$$\Rightarrow \alpha_h < \alpha'_h < \alpha_{h+1}$$

On prend dans S_{T_k} le sous-arbre $T_{h'}^k$ dont $\alpha_{h'}^k$ est le plus proche de α'_h par valeur inférieure ($T_{h'}^k$ « estime » T_h)

CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – Valid. croisée

- On obtient ainsi K estimations de $C(T_h)$

$C(T_h)$ estimé par $\hat{C}^{vc}(T_h)$

= la moyenne de ces K estimations

- Les auteurs de la méthode CART proposent de calculer

$$\widehat{e.t.}[\hat{C}^{vc}(T_h)] = \sqrt{\frac{\hat{C}^{vc}(T_h)[1 - \hat{C}^{vc}(T_h)]}{n}}$$

CART – 2.2 Elaguer l'arbre maximal

2.2.2 Choix du meilleur sous-arbre – Valid. croisée

- On sélectionne le plus petit sous-arbre tel que ?
- L'arbre sélectionné est celui de la séquence originale apprise sur l'ensemble de l'échantillon d'apprentissage

CART – 2.3 Division équi-réductrice et équi-divisante

- La première division équi-réductrice (ou **concurrente, competing rule**) d'un segment t est celle qui correspond à une réduction de l'impureté la plus proche de celle de la meilleure division div^* . C'est en fait la deuxième meilleure division du segment.
- On définit aussi les 2ème, 3ème, ..., divisions équi-réductrices .

CART – 2.3 Division équi-réductrice et équi-divisante

Division équidivisante (suppléantes, surrogate)

- Recherche de la division la plus semblable à la meilleure division ***div****:
on croise les divisions possibles des autres variables avec ***div**** (2 segments descendants chacune)
- 1^{ère} division équidivisante :
celle qui maximise la concordance avec ***div****
- Puis 2^{nde}, 3^{ème}, etc ... divisions suppléantes

CART – 2.3 Division équi-réductrice et équi-divisante

- **Intérêt des divisions équi-réductrices :**
Choisir lors de la construction une variable alternative qui soit plus pertinente (d'un point de vue médical par ex), ou moins coûteuse à collecter
- **Intérêt des divisions équi-divisantes :**
Permet de classer un individu ayant des valeurs manquantes parmi les régresseurs

CART – 2.4 Résultats

- Une fois choisi l'**arbre optimal** au sens de Breiman, on a une **règle de décision**
- On estime son risque moyen par le taux de mal classés par validation simple ou croisée.
- On peut calculer la sensibilité, spécificité, taux de faux-positifs, faux-négatifs, ...etc (voir partie sur la comparaison de méthodes).
- Sans oublier d'enrichir l'analyse avec la prise en compte des variables **équi-réductrices** et d'utiliser des variables **équi-divisantes** si nécessaire

CART – 3. Arbres de régression

- La variable cible Y est une variable **quantitative**
- L'objectif est encore de prédire la valeur prise par Y pour les individus
- La procédure est similaire à celle des arbres de classement, mais la **mesure de l'impureté d'un nœud** et de celle **du coût** doivent être adaptées à la nature de Y .

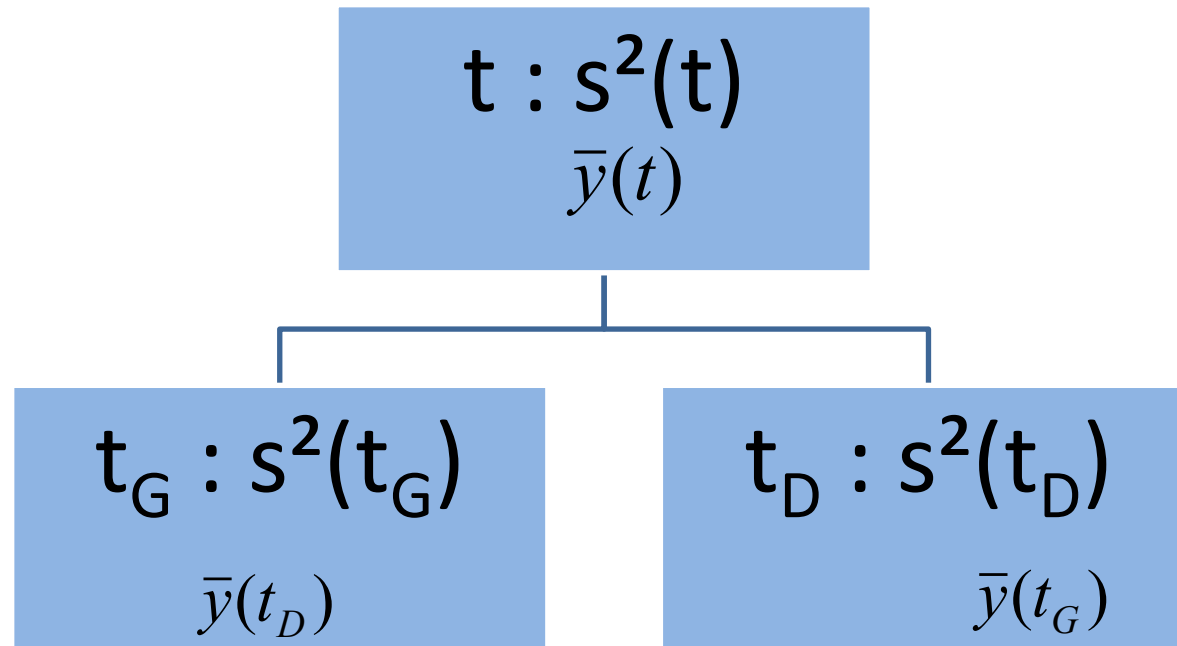
CART – 3.1 Mesure de l'impureté d'un nœud

Lorsque la variable à expliquer Y est continue, le critère de sélection de la meilleure division d'un nœud est fondé sur la variance de Y dans les nœuds enfants : un nœud t est divisé en deux nœuds t_G et t_D aussi homogènes que possible en Y .

Notations	Dans t	Dans t_G	Dans t_D
Moyenne de Y	$\bar{y}(t)$	$\bar{y}(t_G)$	$\bar{y}(t_D)$
Variance de Y	$s^2(t)$	$s^2(t_G)$	$s^2(t_D)$

CART – 3.1 Mesure de l'impureté d'un nœud

L'*impureté* $i(t)$ du nœud t est mesurée par la **variance** de Y à l'intérieur du nœud.



Division \Rightarrow décomposer la variance de Y

CART – 3.1 Mesure de l'impureté d'un nœud

Variance totale = variance inter + variance intra

$$s^2(t) =$$

Variance intra = $w(\text{div}, t)$

$$\frac{n(t_G)}{n(t)} s^2(t_G) + \frac{n(t_D)}{n(t)} s^2(t_D)$$

+

Variance inter = $b(\text{div}, t)$

$$\frac{n(t_G)}{n(t)} [\bar{y}(t_G) - \bar{y}(t)]^2 + \frac{n(t_D)}{n(t)} [\bar{y}(t_D) - \bar{y}(t)]^2$$

CART – 3.1 Mesure de l'impureté d'un nœud

Chaque division **div** d'un nœud t en deux nœuds t_G et t_D provoque une réduction de l'impureté égale à :

$$\Delta i(t, div) = i(t) - (n(t_G) / n(t)) \cdot i(t_G) - (n(t_D) / n(t)) \cdot i(t_D)$$

Soit :

$$\Delta i(t, div) = s^2(t) - (n(t_G) / n(t)) \cdot s^2(t_G) - (n(t_D) / n(t)) \cdot s^2(t_D)$$

= variance totale – variance intra

= variance inter

CART – 3.2 Meilleure division d'un nœud

La meilleure division ***div***^{*} d'un nœud *t* est telle que la variance inter est maximale, ou la variance intra est minimale.

$$\mathbf{div}^* = \underset{div \in \mathcal{Div}}{\operatorname{argmin}} w(div, t)$$

avec \mathcal{Div} = ensemble des divisions binaires possibles de *t* (toutes les divisions binaires possibles de toutes les variables explicatives possibles)

CART – 3.3 Règle de prédiction

Règle de prédiction au sein d'un nœud terminal

- Les observations d'un nœud terminal t se voient affecter comme valeur de Y la moyenne dans le nœud.

$$\forall i \in t, \hat{y}_i = \bar{y}(t)$$

- Erreur de prédiction pour une observation de t :

$$\forall i \in t, e_i = y_i - \hat{y}_i = y_i - \bar{y}(t)$$

CART – 3.4 Coût de l'arbre

Estimation du coût de l'arbre par resubstitution

- Coût d'un segment t :
 - = son hétérogénéité par rapport à Y
 - = variance de Y dans ce segment
 - = moyenne des carrés des erreurs de prédiction
- Coût de l'arbre T calculé sur les nœuds terminaux

CART – 3.4 Coût de l'arbre

- Coût de l'arbre T estimé par resubstitution

$$\widehat{C}^{res}(T) = \sum_{t \in \tilde{T}} p(t) s^2(t)$$

$$\text{avec } p(t) = \frac{n(t)}{n}, \quad s^2(t) = \frac{1}{n(t)} \sum_{i \in t} (y_i - \bar{y}_t)^2$$

- Ce coût est aussi appelé erreur apparente de prévision

CART – 3.4 Coût de l'arbre

- En divisant ce coût de l'arbre T par $s^2(t_0)$ = variance de Y à la racine, on obtient l'équivalent du $1 - R^2$ de la régression linéaire multiple.
- On obtient donc la part de variance totale non expliquée par les variables qui interviennent dans l'arbre de segmentation.

CART – 3.4 Coût de l'arbre

Estimation du coût de l'arbre par validation simple

- Avec l'échantillon de validation de taille n^{dv} et les valeurs prises par les individus $i \in d_v$ Y_i
- Coût d'un **segment terminal** :

$$\hat{C}^{vs}(t) = p(t) s_{d_v}^2(Y/t) \quad \text{avec} \quad p(t) = \frac{n^{dv}(t)}{n^{d_v}}$$

$$s_{d_v}^2(Y/t) = \frac{1}{n^{d_v}(t)} \sum_{\substack{i \in t \\ i \in d_v}} \left(y_i - \bar{y}(t)^{base} \right)^2$$

sur l'échantillon de base



CART – 3.4 Coût de l'arbre

Coût de l'arbre T estimé par validation simple

$$\begin{aligned}\hat{C}^{vs}(T) &= \sum_{t \in \tilde{I}} p(t) s_{d_v}^2(Y/t) \\ &= \frac{1}{n^{d_v}} \sum_{t \in \tilde{I}} \sum_i \left(y_i - \bar{y}(t)^{base} \right)^2 \\ &= \frac{1}{n^{d_v}} \sum_{i \in d_v} \left(y_i - \hat{y}_i \right)^2\end{aligned}$$

CART – 3.4 Coût de l'arbre

- Pour l'application de la règle d'1 écart-type pour sélectionner sous-arbre optimal, on calcule la variance estimée

$$\begin{aligned} & \hat{Var}(\hat{C}^{vs}(T)) \\ &= \frac{1}{(n^{d_v})^2} \sum_{i \in d_v} \left[(y_i - \hat{y}_i)^4 \right] - \left[\hat{C}^{vs}(T) \right]^2 \end{aligned}$$

- Autre méthode possible pour estimer le coût de l'arbre : la validation croisée

4 - CART avec R - Spotify

```
> library(rpart)
> cart <- rpart(data=spotify[,c(-15,-16)],like~.,
+               minsplit=50,xval=10)
> summary(cart)
Call:
rpart(formula = like ~ ., data = spotify[, c(-15, -16)], minsplit = 50,
      xval = 10)
n= 2017
```

	CP	nsplit	rel error	xerror	xstd
1	0.27683049	0	1.0000000	1.0230692	0.02252162
2	0.06619860	1	0.7231695	0.7372116	0.02167902
3	0.06318957	2	0.6569709	0.7111334	0.02150695
4	0.04814443	3	0.5937813	0.6339017	0.02089466
5	0.01003009	4	0.5456369	0.5747242	0.02031481
6	0.01000000	6	0.5255767	0.5636911	0.02019536

Variable importance

loudness	instrumentalness	energy	speechiness	acousticness
23	21	17	14	11
valence	duration	tempo	danceability	liveness
6	5	1	1	1

4 - CART avec R - Spotify

Node number 1: 2017 observations, complexity param=0.2768305

predicted class=1 expected loss=0.4942985 P(node) =1

class counts: 997 1020

probabilities: 0.494 0.506

left son=2 (988 obs) right son=3 (1029 obs)

Primary splits:

instrumentalness < 5.685e-05 to the left, improve=81.86019, (0 missing)

loudness < -4.895 to the right, improve=56.55462, (0 missing)

danceability < 0.7265 to the left, improve=43.82988, (0 missing)

duration < 279099.5 to the left, improve=35.95577, (0 missing)

energy < 0.2045 to the left, improve=33.50939, (0 missing)

Surrogate splits:

duration < 252074 to the left, agree=0.624, adj=0.233, (0 split)

loudness < -6.754 to the right, agree=0.612, adj=0.209, (0 split)

acousticness < 0.01155 to the right, agree=0.580, adj=0.142, (0 split)

speechiness < 0.06965 to the right, agree=0.566, adj=0.114, (0 split)

valence < 0.3805 to the right, agree=0.561, adj=0.103, (0 split)

4 - CART avec R - Spotify

Node number 58: 99 observations

predicted class=0 expected loss=0.4141414 P(node) =0.0490828

class counts: 58 41

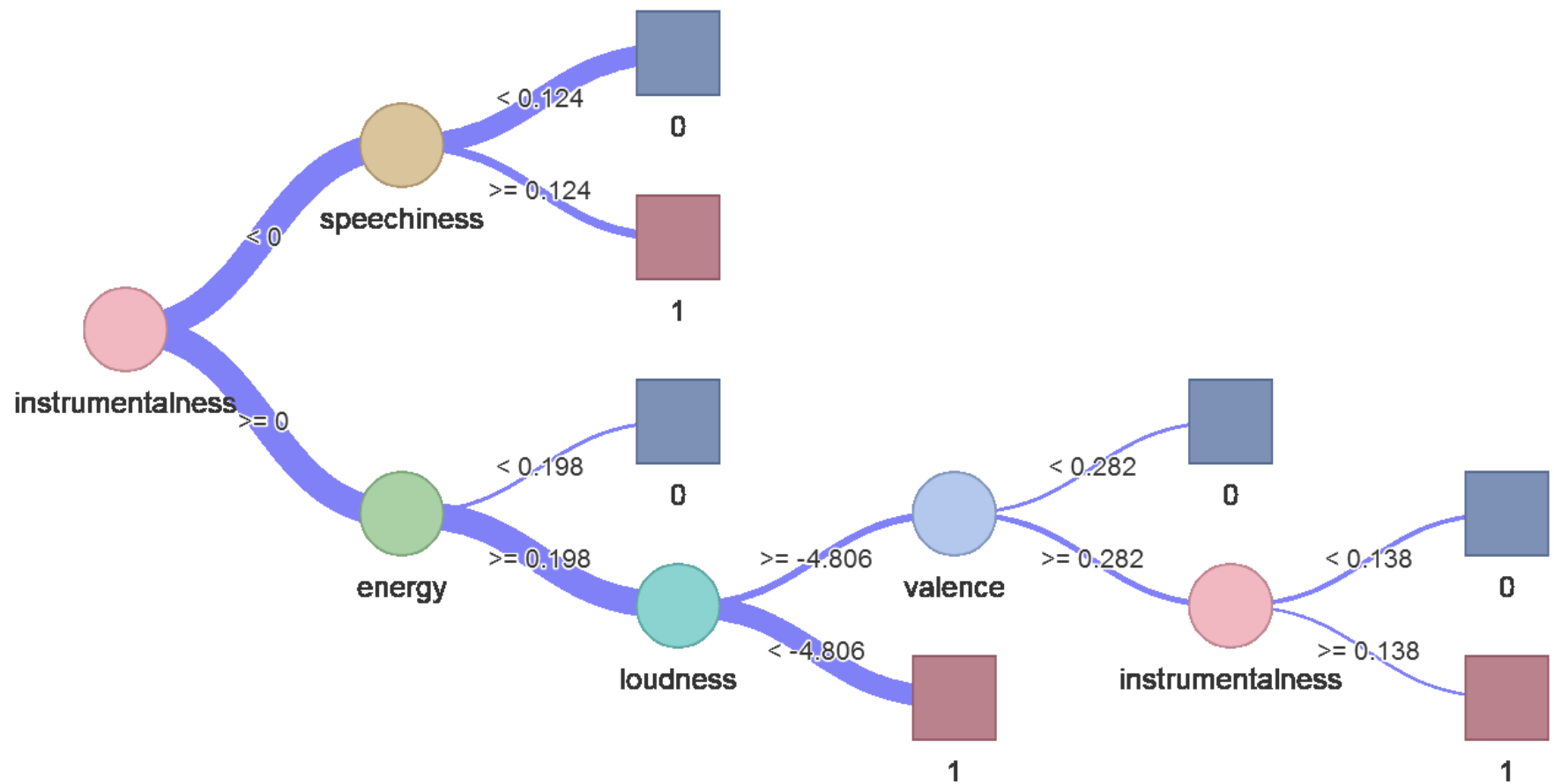
probabilities: 0.586 0.414

Node number 59: 42 observations

predicted class=1 expected loss=0.2619048 P(node) =0.020823

class counts: 11 31

probabilities: 0.262 0.738



CART – Bilan

Bilan sur l'utilisation des échantillons

- **Construction de l'arbre maximal :**
Echantillon de base (apprentissage)
- **Elagage de l'arbre maximal :** procédure basée sur le critère de coût-complexité = échantillon d'apprentissage
- **Estimation du coût réel des arbres de la séquence et choix du meilleur arbre :** validation simple sur échantillon de validation ou croisée sur échantillon d'apprentissage (divisé en sous-échantillons apprentissage+ validation).

CART – Bilan

- Une fois choisi l'**arbre optimal** au sens de Breiman, on a une **règle de décision** dont on a estimé l'efficacité par son **coût réel**
- On peut enrichir l'analyse avec la prise en compte des variables **équi-réductrices** et utiliser des variables **équi-divisantes** si nécessaire

C - CHAID

CHAID - Généralités

CHAID = CHi2 Automatic Interaction Detector
(Kass, 1980)

- Variable cible : nominale, ordinale, ou continue
- Prédicteurs : variables nominales, ordinales ou continues
- Divisions en un nombre de nœuds quelconque
- Règles d'arrêt : critères statistiques

CHAID – 1 . Arbres de classement

- Y = variable à expliquer ***nominale***, à R modalités $G_1 \dots G_r \dots G_R$.
- $X_1 \dots X_j \dots X_J$ = variables explicatives ***nominales***
- On note $X_j^1, X_j^2, \dots, X_j^{M_j}$ les M_j modalités de X_j .
- **Deux algorithmes utilisés** : un pour construire l'arbre et un pour regrouper les modalités des variables explicatives

CHAID – 1.1 Statistique du χ^2

On peut trouver deux variantes de la statistique du χ^2 utilisée dans l'algorithme de regroupement des modalités et dans celui de construction de l'arbre :

- Le χ^2 de **Pearson** pour deux variables qualitatives à (respectivement) R et M_j modalités

$$X^2 = \sum_{r=1}^R \sum_{l=1}^{M_j} \frac{\left(n_{rl} - \frac{n_{r.}n_{.l}}{n}\right)^2}{\frac{n_{r.}n_{.l}}{n}}$$

CHAID – 1.1 Statistique du χ^2

- Le χ^2 du **ratio de vraisemblance** pour deux variables qualitatives à (resp.) R et M_j modalités

$$X^2_{RV} = -2 \log \theta$$

$$\theta = \frac{\prod_{r=1}^R \prod_{l=1}^{M_j} \left(\frac{n_{r.} n_{.l}}{n^2} \right)^{n_{rl}}}{\prod_{r=1}^R \prod_{l=1}^{M_j} \left(\frac{n_{rl}}{n^2} \right)^{n_{rl}}}$$

Sous l'hypothèse d'indépendance H_0 ,
cette statistique suit asymptotiquement une
loi du khi-deux à $(R-1) \times (M_j - 1)$ degrés de liberté.

CHAID – 1.2 Construction de l'arbre

Principe de construction de l'arbre

1. L'échantillon (d'apprentissage) complet constitue la **racine de l'arbre**.
2. Pour chaque variable explicative X_j , on effectue un **regroupement "optimal" de modalités**

CHAID – 1.2 Construction de l'arbre

3. On cherche parmi les J variables explicatives X_j , leurs modalités ayant été regroupées, celle qui donne la meilleure partition du nœud au sens d'un certain critère :

Le nœud est alors scindé en un nombre de nœuds enfants égal au nombre de modalités de la variable sélectionnée (avec éventuel regroupement).

On retourne à l'étape 2 pour chaque nœud ainsi constitué.

CHAID – 1.2 Construction de l'arbre

Le processus s'arrête si, pour chaque nœud, il n'existe aucune variable explicative qui permette de créer une nouvelle partition "**significativement**" meilleure.

CHAID – 1.3 Regroupement des modalités des X_j

Algorithme de regroupement des modalités :

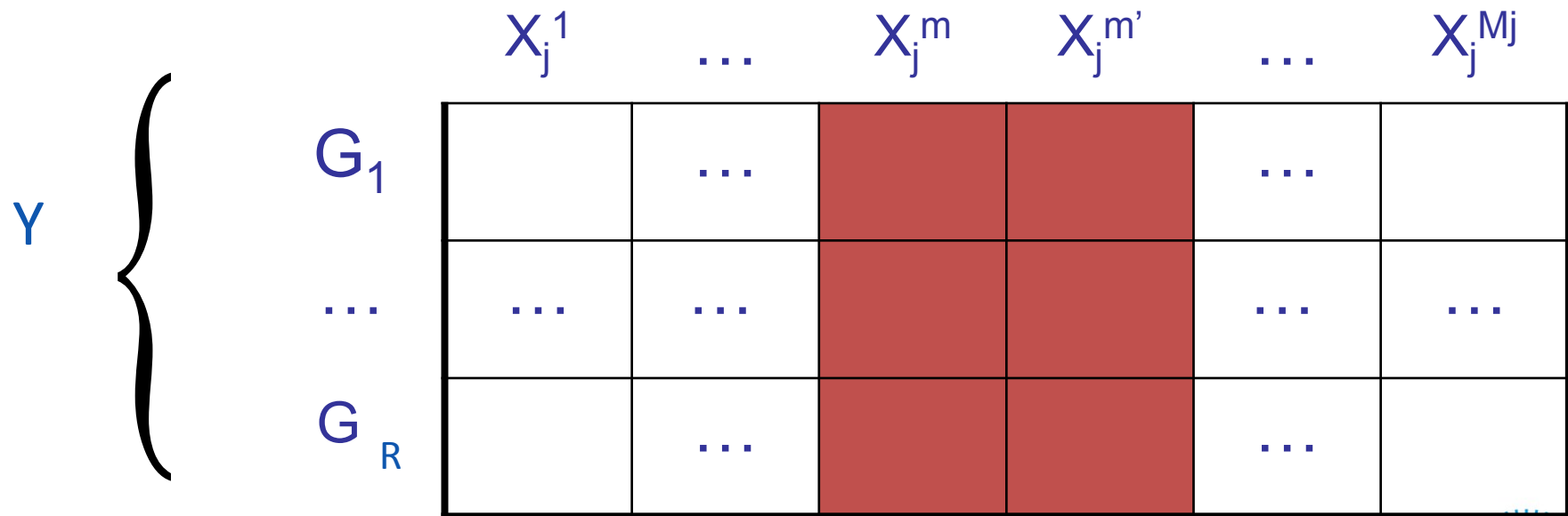
On considère un nœud donné t (l'indice du nœud sera omis dans les notations).

Pour une variable explicative X_j , on considère **le tableau de contingence T_j** qui croise la variable à expliquer Y (en ligne) et la variable explicative X_j (en colonne) :

- en ligne les R modalités G_r de Y
- en colonnes les M_j modalités de X_j .

CHAID – 1.3 Regroupement des modalités des X_j

1. Pour chaque couple $(X_j^m, X_j^{m'})$ de colonnes, on calcule le χ^2 du sous-tableau de T_j , de taille $R \times 2$ croisant Y et ces deux colonnes.



		X_j^1	...	X_j^m	$X_j^{m'}$...	X_j^{Mj}
Y	G_1		

	G_R		

Tableau de contingence T_j

CHAID – 1.3 Regroupement des modalités des X_j

2. On choisit le couple $(X_j^m, X_j^{m'})$ qui a le plus faible χ^2 observé :

Si ce χ^2 est inférieur au χ^2 théorique correspondant à un seuil α fixé (i.e. si la liaison entre Y et les deux modalités considérées est faible), **on regroupe ces deux colonnes**, et on revient à l'étape 1, avec un tableau ayant le même nombre de lignes mais une colonne en moins.

CHAID – 1.3 Regroupement des modalités des X_j

On obtient ainsi un tableau de contingence ayant $M_j^* \leq M_j$ colonnes, correspondant à M_j – nombre de regroupements de modalités.

De façon optionnelle, l'algorithme permet ensuite la remise en cause de certains regroupements.

CHAID – 1.3 Regroupement des modalités des X_j

Pour chaque modalité obtenue par regroupement de $u > 2$ modalités originelles :

3. **On examine toutes les divisions binaires** (i.e. les partages des u modalités en deux sous-ensembles de modalités) de l'ensemble des u modalités, et on calcule pour chacune d'elles le χ^2 du tableau croisant Y et les deux colonnes correspondant à ces deux sous-ensembles.

CHAID – 1.3 Regroupement des modalités des X_j

4. On choisit la division qui a le χ^2 le plus élevé : si ce χ^2 est supérieur au χ^2 théorique correspondant à un seuil α fixé (i.e. si la liaison entre Y et les nouvelles colonnes est forte), on effectue cette division binaire, et on revient à l'étape 1, avec une colonne en plus par rapport à l'étape précédente.

Remarque : dans la pratique, l'algorithme effectue rarement des divisions après un regroupement.

CHAID – 1.4 L'algorithme CHAID

Algorithme de construction de l'arbre :

1. Pour un nœud n donné, on examine successivement chaque variable explicative X_j (pour laquelle des regroupements de modalités ont pu être réalisés avec l'algorithme précédent) :

on calcule le χ^2 du tableau croisant Y et X_j , puis la probabilité critique associée (p-valeur).

CHAID – 1.4 L'algorithme CHAID

2. On choisit la variable explicative X_j associée à la probabilité critique la plus faible (i.e. la variable la plus liée à Y) ; si cette probabilité est inférieure à un seuil α fixé, **on divise le nœud en autant de segments que de modalités de la variable choisie.**
- **Règle d'arrêt** : L'algorithme s'arrête si, pour chaque nœud, il n'existe aucune variable explicative qui permette de créer une nouvelle partition significative au sens du khi-deux.

CHAID – 1.5 Correction de Bonferroni

- Pour chaque variable X_j , on remplace la probabilité critique p par **une probabilité "corrigée"** $p' = \theta_j \cdot p$
- Cette correction **tient compte des regroupements de modalités** qui ont été (éventuellement) réalisés sur la variable X_j :
 θ_j est égal au nombre de possibilités de regrouper les M_j modalités en M_j^* groupes :

$$\theta_j = \sum_{i=0}^{M_j^*-1} (-1)^i \frac{(M_j^* - i)^{M_j^*}}{i! (M_j^* - i)!}$$

CHAID – 1.5 Correction de Bonferroni

- **Cas de variables explicatives ordinales ou continues**

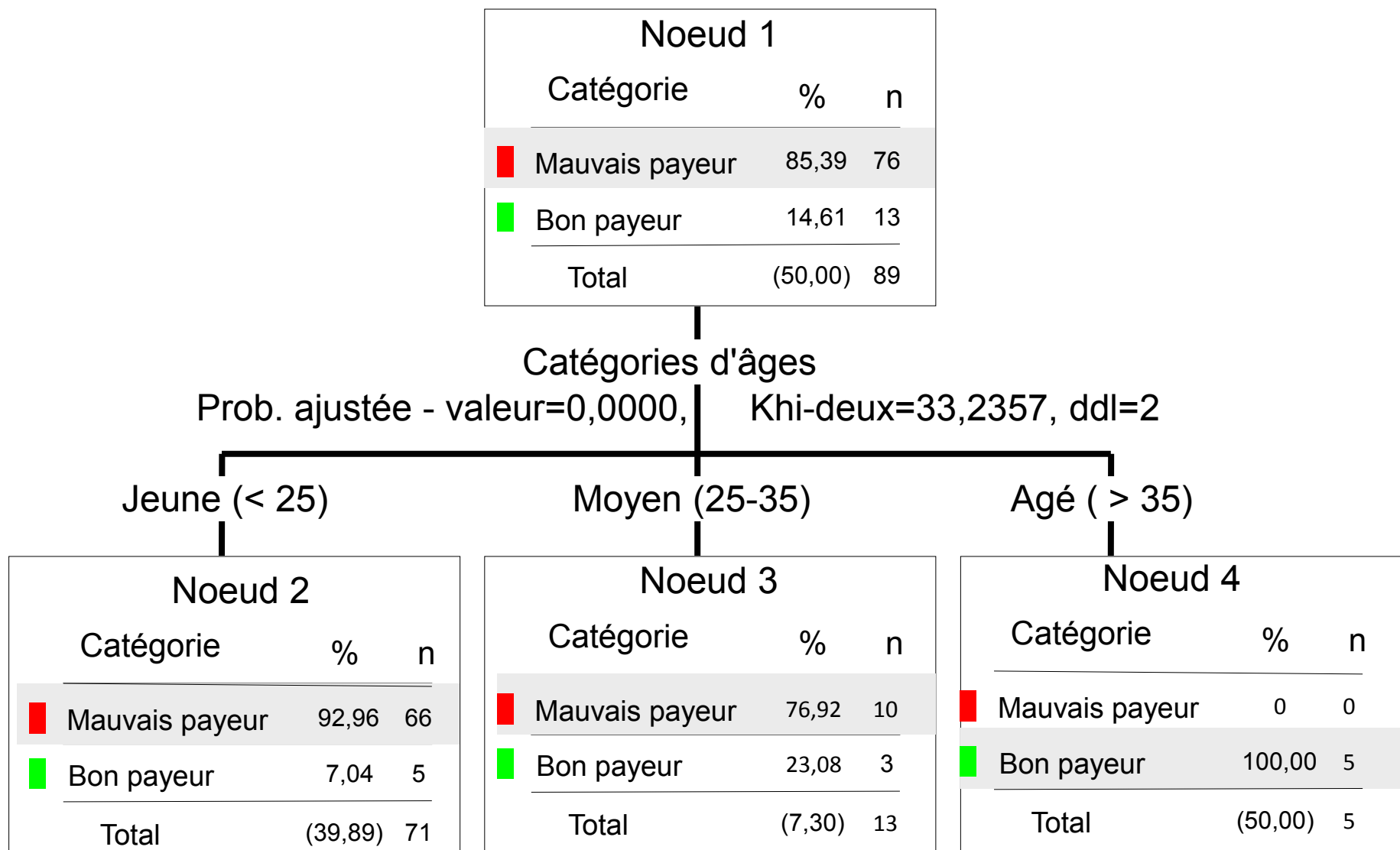
Une variable continue sera traitée comme une variable ordinale.

Dans l'algorithme de regroupement de modalités, seules des modalités "contiguës" peuvent être regroupées.

Le coefficient de Bonferroni est alors égal à :

$$\theta_j = \left(\frac{M_j - 1}{M_j^* - 1} \right)$$

CHAID – 1.5 Correction de Bonferroni



CHAID – 2. Arbre de régression

- Y = variable à expliquer *continue*
- $X_1 \dots X_j \dots X_J$ = variables explicatives *nominales*
- On note $X_j^1, X_j^2, \dots, X_j^{M_j}$ les M_j modalités de X_j .
- **Deux algorithmes utilisés** : un pour regrouper les modalités des variables explicatives et un pour construire l'arbre.

CHAID – 2.1 Statistique de Fisher

On utilise la statistique F de Fisher pour choisir la meilleure variable explicative.

On note $X_j^1, X_j^2, \dots, X_j^{M_j}$ les M_j modalités de X_j

Notations	Dans le nœud t	Dans la modalité X_j^m
Effectif	$n(t)$	$n(X_j^m)$
Moyenne de Y	$\bar{y}(t)$	$\bar{y}(X_j^m)$
Variance de Y	$s^2(t)$	$s^2(X_j^m)$

CHAID – 2.1 Statistique de Fisher

On utilise la décomposition de la variance :

Variance totale (t) = var. intra + var.inter

$$s^2(t) = \sum_{m=1}^{M_j} \frac{n(X_j^m)}{n(t)} s^2(X_j^m) + \sum_{m=1}^{M_j} \frac{n(X_j^m)}{n(t)} \left(\bar{Y}(X_j^m) - \bar{Y}(t) \right)^2$$

On utilise la statistique de Fisher pour tester l'hypothèse H_0 que les moyennes de Y par modalité sont égales.

$$F = \frac{n(t) - M_j}{M_j - 1} \frac{V_{\text{inter}}}{V_{\text{intra}}}$$

CHAID – 2.1 Statistique de Fisher

Sous l'hypothèse H_0 , F suit une loi de Fisher à $M_j - 1$, $n(t) - M_j$ degrés de liberté.

Une valeur élevée de F conduit au rejet de H_0 : elle traduit l'existence d'une relation forte entre la variable continue Y et la variable qualitative X_j .

On calcule la probabilité critique associée au test et on la compare à un seuil α fixé.

CHAID – 2.2 L'algorithme CHAID

1. Pour un nœud t donné, on examine successivement chaque variable explicative X_j : on calcule le F de Fisher indiquant l'intensité de la liaison entre Y et X_j , puis la probabilité critique associée p .

Si des regroupements de modalités de X_j ont été réalisés alors c'est la probabilité corrigée p' qu'on utilise : obtenue avec l'ajustement de Bonferroni.

CHAID – 2.2 L'algorithme CHAID

2. On retient la variable explicative X_j associée à la probabilité critique la plus faible = la variable la plus liée à Y .

Si cette probabilité est inférieure à un seuil α fixé, on divise le nœud en autant de segments que la variable X_j choisie a de modalités.

CHAID – 2.2 L'algorithme CHAID

Règle d'arrêt :

L'algorithme s'arrête lorsque, pour chaque nœud, on ne trouve aucune variable explicative qui permette de créer une nouvelle partition significative au sens du F de Fisher.

D – Avantages, inconvénients des arbres de décision

Avantages

- **Les résultats sont explicites :**
 - Les conditions sont formulées sur les variables d'origine
 - Aspect visuel = graphiques (arbres)
- **La technique est non paramétrique :**
 - pas d'hypothèse sur les lois suivies par les variables explicatives
- **Des variables de différents types peuvent être traitées directement :** quantitatives, nominales, ordinales par CART par ex.

Avantages

- **La réponse de la cible peut être non linéaire** en fonction des variables explicatives.
- **Procédure naturelle de sélection** des variables explicatives en pas à pas
- **L'arbre peut détecter les interactions** entre plusieurs variables
- **L'arbre n'est pas modifié par une transformation monotone** des variables explicatives
⇒ Phase de préparation et sélection de variables simplifiée

Avantages

- Les individus hors norme perturbent peu les résultats : ils peuvent être isolés dans de petits nœuds.
- Les arbres peuvent gérer les valeurs manquantes :
 - Avec CHAID, les valeurs manquantes d'une variable = une modalité à part ou fusionnée
 - Avec CART, utilisation des variables équadivisantes ou suppléantes

Inconvénients

- **Aspect séquentiel** : la définition des nœuds au niveau $n+1$ dépend de celle au niveau n donc
 - **Détection d'optimum locaux** et non globaux car évaluation séquentielle et non simultanée des variables explicatives
 - **Manque de robustesse** : si modification d'une seule variable placée près du sommet alors modification possible de tout l'arbre \Rightarrow bagging & boosting

Inconvénients

- **De grands échantillons** sont nécessaires pour avoir une certaine fiabilité
- Les règles obtenues définissent des **régions rectangulaires qui ne correspondent pas forcément à la distribution des individus**
⇒ difficulté à classer les individus si la répartition n'est pas rectangulaire