

ARBRES DE DECISION

Principe: Classifier un objet par une suite de tests sur ses attributs.

↳ sera forme d'arbres

↳ un nœud est un test sur des attributs dont la réponse désigne le fils sur lequel on doit aller.

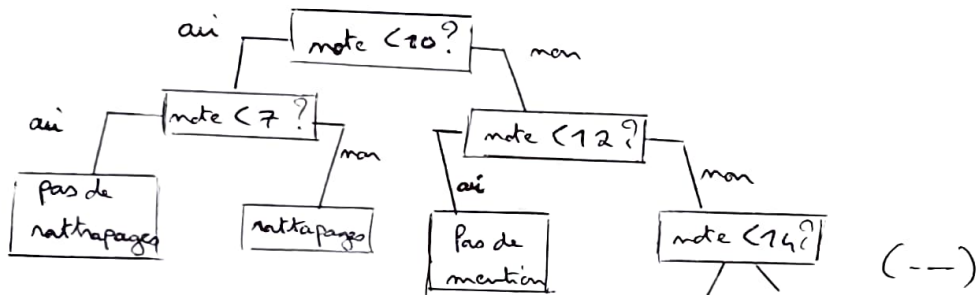
Objectifs:

- nombre moyen de tests la
- ⊗ faible possible
- interprétation facile
- utilisation fiable en prédiction

Donc un bon arbre est :

- Compact
- lisible
- d'erreur faible
- de complexité acceptable

Ex:



I] Construction (Apprentissage):

$\left\{ \begin{array}{l} S : \text{échantillon} \\ \text{classes } \{1, \dots, c\} \\ \text{arbre } T \\ p : \text{position dans } T \end{array} \right.$

$\left\{ \begin{array}{l} N(p) : \text{card de } \{ \text{exemples à } p \} \\ N(k|p) : | \{ \text{exemples à } p \text{ de classe } k \} | \\ P(k|p) = N(k|p) / N(p) \text{ proportion} \\ \text{d'éléments de classe } k \text{ à } p. \end{array} \right.$

Mélange: C_1 et C_2 des classes.

↳ si $n/2$ exemples de classe C_1 : mélange maximal

↳ si $n-1$ ou 1 exemple(s) de classe C_1 : ensemble presque pur.

Fonction mesurant le degré de mélange: - max quand équirépartition
- min quand tout le monde de même cl.

↳ classiquement: Entropie $(p) = - \sum_{k=1}^c P(k|p) \log_2(P(k|p))$

$$\text{Gini}(p) = 1 - \sum_{k=1}^c P(k|p)^2$$

Pour choisir le test à appliquer : avec $i()$ pct° mesurant le mélange et P_j proportion d'élément de p qui iraient sur la j -ième filière p_j ,

On utilise $\text{Gain}(p, \text{test}) = i(p) - \sum_{j=1}^n P_j i(p_j)$

↳ On veut minimiser l'entropie, donc on prend le test qui maximise le gain!

Ex de construction:

ID	Sexe	Age	Etat civil	Revenu	Achat
1	Homme	18-35	marie	moyen	non
2	Homme	<18	Célibataire	faible	non
3	Homme	>35	marie	élevé	oui
4	Femme	<18	Célib	moyen	non
5	Homme	18-35	Célib	moyen	non
6	Femme	18-35	Célib	élevé	oui
7	Femme	18-35	marie	faible	non
8	Homme	18-35	marie	élevé	oui
9	Homme	>35	Célib	faible	oui
10	Femme	<18	Célib	moyen	non
11	Femme	>35	Célib	moyen	oui
12	Femme	>35	marie	élevé	oui
13	Homme	18-35	Célib	faible	non
14	Femme	18-35	marie	moyen	oui

avec l'entropie E .

→ 14 attributs, 7 oui et 7 non: $E(\text{racine}) = -\frac{7}{14} \log_2\left(\frac{7}{14}\right) - \frac{7}{14} \log_2\left(\frac{7}{14}\right) = 1$

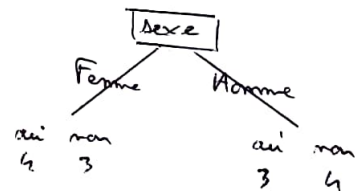
↳ on regarde les tests un par un:

↳ sexe: $E(\text{Homme}) = -\frac{3}{7} \log_2\left(\frac{3}{7}\right) - \frac{4}{7} \log_2\left(\frac{4}{7}\right)$
 $= 0,985$

$E(\text{Femme}) = 0,985$

donc $\text{Gain}(\text{racine}, \text{sexe}) =$

$E(\text{racine}) - \frac{7}{14} E(\text{Homme}) - \frac{7}{14} E(\text{Femme}) = 0,015$

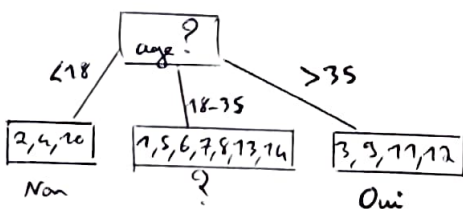


↳ Age: $\text{Gain}(\text{racine}, \text{age}) = 0,507$

↳ Etat civil: $\text{Gain}(\text{racine}, \text{etat civil}) = 0,061$

↳ Revenu: $\text{Gain}(\text{racine}, \text{revenu}) = 0,375$

) ⇒ On prends l'Age.
 Alors l'arbre devient



Ensuite, on étend l'arbre depuis un autre nœud...

II Texte :

La construction de l'arbre se termine lorsque les nœuds sont déclarés terminaux. Souvent lorsque toutes les feuilles sont pures, mais pas toujours. Alors on définit l'erreur apparente sur $S = \frac{|\{\text{exemples mal classés}\}|}{|S|}$

On peut tester l'arbre sur un échantillon S' tq $S \cap S' = \emptyset$ pour obtenir l'erreur réelle $= \frac{|\{\text{exemples mal classés de } S'\}|}{|S'|}$

↳ l'erreur apparente est une vision très optimiste de l'erreur réelle.

↳ Un arbre peut être bien adapté à l'échantillon (err. app. faible) mais avoir un pouvoir décisionnel faible (err. réelle forte)

Pour régler ça, on peut élaguer l'arbre.

méthode:

up sans arbre de racine p .

$\Delta_{app}(p)$ variation d'erreur apparente sur S lorsqu'on élague en p . $\Delta_{app} =$

$$\text{On définit } g(p) = \frac{\Delta_{app}}{|u_p| - 1}$$

On construit $t_0 = T, t_1, \dots, t_m$ tq $\begin{cases} t_m \text{ est une feuille} \\ \forall i, t_{i+1} = \text{élagué de } t_i \text{ à la position } t_q \\ g \text{ est minimal.} \end{cases}$

Enfin, on calcule l'erreur réelle sur tous les t_i et on garde celui où elle est la plus faible.

III Algorithmes Classiques:

CART: (Tests et attributs linéaires) $\begin{cases} \text{mesure du mélange: Gain} \\ \text{nœud terminal: Gain}(p) < i_0 \text{ ou } N(p) < N_0 \end{cases}$
puis élagage.

ID3: Arbre le plus simple possible $\begin{cases} \text{mesure du mélange: Entropie} \\ \text{nœud terminal: si le nœud est pur} \end{cases}$

C4.5: version de ID3 améliorée pour limiter le nombre de branches. $\begin{cases} \text{mesure du mélange: Entropie} \\ \text{nœud terminal: si le nœud est pur} \end{cases}$

$$SplitInfo(p, test) = - \sum_{j=1}^m P(j|p) \log_2(P(j|p)) \text{ et } GainRatio = \frac{Gain}{SplitInfo}$$