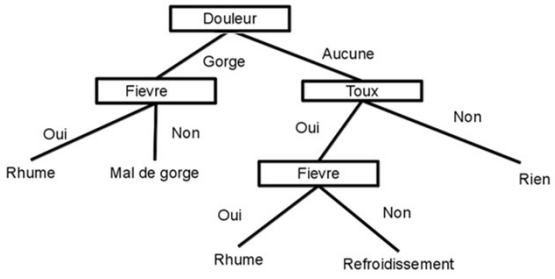
Apprentissage automatique

Lecture 4: Arbres de décision

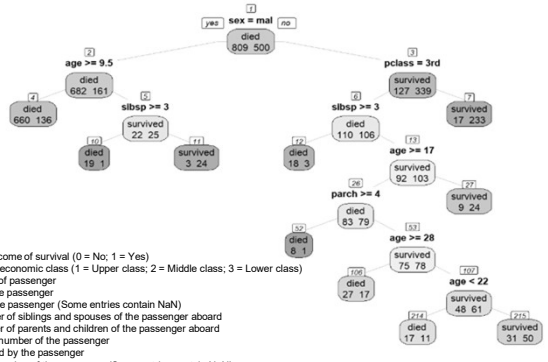
Département Génie Informatique, FST de Tanger

M. AIT KBIR

Arbres de décision : Exemples (Diagnostic d'une maladie)



Arbres de décision : Exemples (Prédiction des survivants du Titanic)



Survived: Outcome of survival (0 = No; 1 = Yes)
Pclass: Socio-economic class (1 = Upper class; 2 = Middle class; 3 = Lower class)
Name: Name of passenger
Sex: Sex of the passenger
Age: Age of the passenger (Some entries contain NaN)
SibSp: Number of siblings and spouses of the passenger aboard
Parch: Number of parents and children of the passenger aboard
Ticket: Ticket number of the passenger
Fare: Fare paid by the passenger
Cabin Cabin number of the passenger (Some entries contain NaN)
Embarked: Port of embarkation of the passenger (C = Cherbourg; Q = Queenstown; S = Southampton)

Entropie: Définition

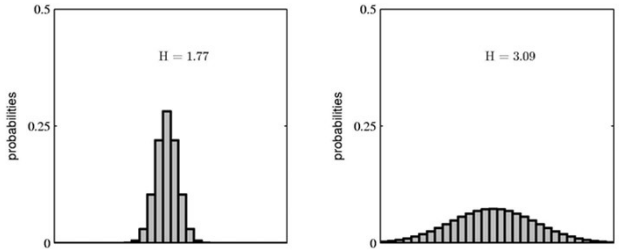
Le terme entropie caractérise le degré de désorganisation, ou d'imprédictibilité du contenu en information d'un système.

$$\mathrm{H}[x] = -\sum_{x} p(x) \log_2 p(x)$$

Utile dans :

- Codage de l'information
- Physique statistique
- Apprentissage automatique

Entropie



- Entropie élevée : variable aléatoire est issu d'une distribution plate
- Entropie basse : variable aléatoire issu d'une distribution non plate (avec des pics)

Entropie : codage de l'information

- Soit x une variable aléatoire décrite avec 8 états possibles; de combien de bits on a besoin pour transmettre l'état de x ?
- La longueur optimal du code est donnée par le théorème de Shannon : $-log_2(p(x))$ pour chaque état de x.

- Si tous les états sont équiprobables:
  - $-log_2\left(\frac{1}{8}\right) = 3$ bits pour chaque état
  - *Entropie* H = -8 x $\frac{1}{8} log_2\left(\frac{1}{8}\right)$

Entropie : codage de l'information

Les états ne sont pas équiprobables:

x	a	b	c	d	e	f	g	h
$p(x)$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{1}{64}$

Longueur moyenne du code:

$$\begin{aligned} \mathrm{H}[x] &= -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{8}\log_2\frac{1}{8} - \frac{1}{16}\log_2\frac{1}{16} - \frac{4}{64}\log_2\frac{1}{64} \\ &= 2 \text{ bits} \end{aligned}$$

Exemple:

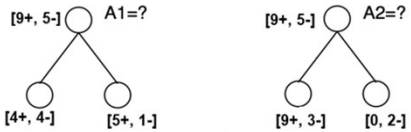
x	a	b	c	d	e	f	g	h
code	0	10	110	1110	111100	111101	111110	111111

Arbre décision: diviser pour régner

Pour créer un arbre de décision, on doit prendre une décision sur l'ensemble des données pour savoir la caractéristique à utiliser pour fractionner les données. Ensuite, diviser le jeu de données en sous-ensembles. On parcourt ensuite les branches à partir du nœud créé. Si les données sur une sous-branche appartiennent à la même classe, on a pas besoin de continuer à les diviser. Si les données ne sont pas identiques, on doit répéter le processus de division sur ce sous-ensemble.

Mesure de d'information

Le changement d'information avant et après la division est connu sous le nom de gain d'information. On fractionne les données en choisissant chaque fois la division qui donne le gain d'information le plus élevé.



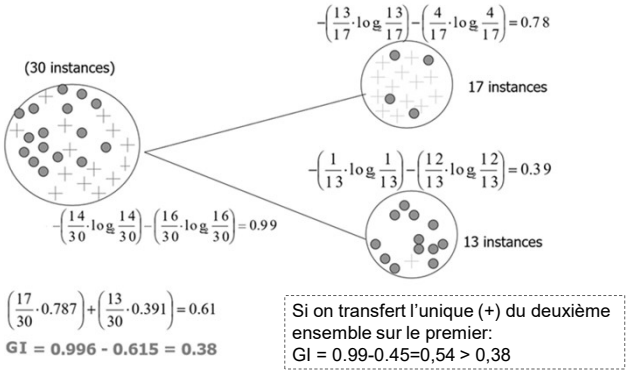
Gain d'information

Le gain d'information qui résulte du fractionnement de de l'ensemble des données (S) par rapport à la caractéristique x_i, qui prend des valeurs dans $\{v_{i1}, v_{i2}, \dots, v_{iM_i}\}$, est la différence entre la quantité d'information calculée avant et après le fractionnement en sous-ensembles. Lorsque l'information est mesurée par entropie :

$$Gain(S, x_i) = H(S) - H(S|x_i)$$
$$= H(S) - \sum_{m=1}^{M} \frac{|S_{im}|}{|S|} H(S_{im})$$

$|\ \ | = Taille\ du\ sous - ensemble$

Entropie : Division



$\left(\frac{17}{30}\cdot 0.787\right)+\left(\frac{13}{30}\cdot 0.391\right)=0.61$

GI = 0.996 - 0.615 = 0.38

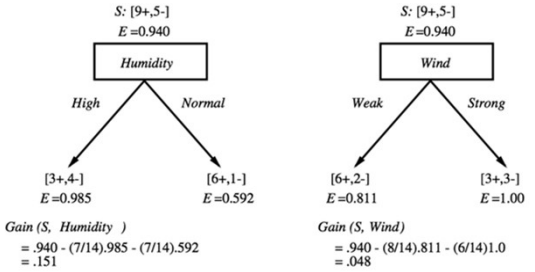
Si on transfert l'unique (+) du deuxième ensemble sur le premier:
GI = 0.99-0.45=0,54 > 0,38

Gain d'information : Base d'exemples "Weather" (Quinlan, 1993)

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

Gain d'information (Exemple : jouer au tennis)

On cherche à diviser par rapport à la caractéristique x_i qui apporte la plus grande réduction l'entropie. L'entropie après fractionnement est une moyenne pondérée des entropies des sous-ensembles qui correspondent aux différentes valeurs de x_i.

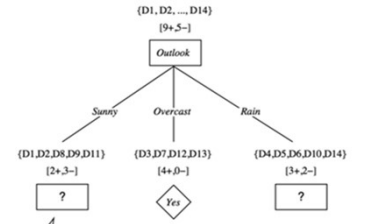


Algorithme : ID3 (Iterative Dichotomiser 3)

Développé en 1986 par Ross Quinlan

a. Commencer avec le nœud racine qui correspond à l'ensemble des exemples de la base d'apprentissage
b. Si la condition d'arrêt n'est pas satisfaite faire:
   c. Calculer le meilleur attribut x_i pour fractionner l'ensemble correspondant au nœud courant
   d. Pour chaque valeur de x_i dans $\{v_{i1}, v_{i2}, \dots, v_{iM_i}\}$, créer un nouveau nœud descendant et le sous-ensemble correspondant (initialement vide)
   e. Attribuer les exemples d'apprentissage, de l'ensemble du nœud courant, aux sous-ensembles des nœuds fils
f. Répéter récursivement les étapes de b à f pour tous les descendants du nœud courant.

Gain d'information (Exemple : jouer au tennis)



Quel est l'attribut suivant ?

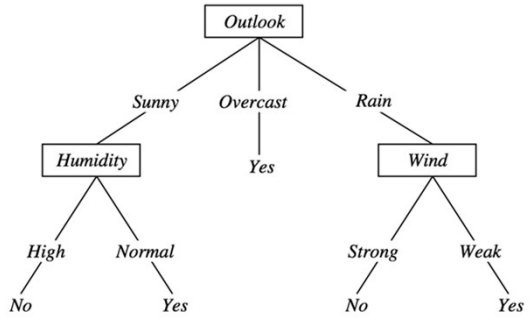
S_{sunny} = {D1,D2,D8,D9,D11}

Gain (S_{sunny} , *Humidity*) = .970 − (3/5) 0.0 − (2/5) 0.0 = .970

Gain (S_{sunny} , *Temperature*) = .970 − (2/5) 0.0 − (2/5) 1.0 − (1/5) 0.0 = .570

Gain (S_{sunny} , *Wind*) = .970 − (2/5) 1.0 − (3/5) .918 = .019

Gain d'information (Exemple : jouer au tennis)



Overcast Or (Sunny And (Normal humidity)) Or (Rain And (Weak wind))

Algorithme ID3: Stratégie de croissance de l'arbre

- Règle de fractionnement
- Condition d'arrêt: qui détermine la fin de la récursivité. C'est la règle qui détermine si un nœud est feuille ou non. Par exemple: - Tous les exemples appartiennent à la même classe - Il ne reste aucun attribut pour plus de fractionnement – Il n'y a plus d'exemples
- Règle d'étiquetage: qui attribue une étiquette de classe à chaque nœud feuille, le vote à la majorité est utilisé pour classer la feuille.

Indice Gini : mesure d'impureté

Le coefficient de **Gini** est une mesure statistique qui permet de mesurer des disparités dans une population. Si S contient des exemples issus de C classes :

$$Gini(S) = 1 - \sum_{j=1}^{C} p(w_j)^2$$

Lors de la construction d'un arbre de décision, il s'agit de fractionner par rapport à la caractéristique avec la valeur minimal de l'indice.

$$Gini(x_i) = 1 - \frac{1}{|S|}\sum_{j=1}^{C}\sum_{m=1}^{M_i}\frac{|(S_{im}^j)|^2}{|S_{im}|}$$

$|S|$ *nombre des exemples du noeuds initial*

$\left|S_{im}^{j}\right|$ *nombre des exemples de la classe j, qui correspondent à la valeur* v_{im}

$|S_{im}|$ *nombre des exemples qui correspondent à la valeur* v_{im}

Arbres de décision

Avantages:

- Les arbres peuvent gérer les espaces de grande dimensionnalité aisément.
- Vu la nature hiérarchique de l'algorithme, le calcul des probabilité est extrêmement rapide.
- Les arbres peuvent aussi traiter des bases données avec des caractéristiques continues.

Inconvénients:

- ID3 n'est pas adapté aux attributs continus
- Souffre de quelques problèmes comme le sur-apprentissage, il peut donner comme résultat un optimum local et non pas la solution globale.
- L'arbre peut nécessiter l'élagage

Autres algorithmes adaptés aux caractéristiques continues: C4.5 et C5.0 (successeurs de ID3) et CART (Classification and Regression Trees).

Arbres de décision: problèmes

- Choix d'une mesure qui permet d'évaluer objectivement la qualité d'un fractionnement et ainsi de sélectionner la meilleure caractéristique à utiliser par rapport à un nœud.
- Choix d'un ou plusieurs seuil pour les attributs continus et la mise en concurrence de ces derniers et les attributs discrets.
- Utilisation des règles efficaces pour définir la taille adéquate de l'arbre de décision lorsqu'un partitionnement pur des observations de la base n'est pas possible.
- Utilisation des règles de décision optimales lorsqu'une feuille contient des exemples avec des classes différentes.

Arbres de décision (C 4.5), R. Quinlan, 1993

C4.5 est une amélioration de ID3 qui permet de traiter les attributs numériques continus, par le partitionnement de ces derniers en un ensemble d'intervalles. L'arbre générée est formulée sous forme d'un nombre de règles SI-ALORS. Cette technique fractionne par rapport à x_i qui maximise le rapport de gain suivant:

$$\frac{Gain(S, x_i)}{-\sum_{m=1}^{M_i} \frac{|S_{im}|}{|S|} log_2(\frac{|S_{im}|}{|S|})}$$

Arbres de décision (C 4.5), R. Quinlan, 1993

Humidity	85	90	86	96	80	70	65	95	70	80	70	90	75	91
Play	no	no	yes	yes	yes	no	yes	no	yes	yes	yes	yes	yes	no

Il s'agit de trier les valeurs de l'attribut, puis itérer à travers les seuils. Prendre chaque fois la moyenne des deux valeurs qui correspondent au changement de classe et séparer le jeu de données en deux ensembles. Puis calculer le rapport de gain pour chaque valeur du seuil, pour garder celle qui correspond au maximum.

Humidity	65	70	70	70	75	80	80	85	86	90	90	91	95	96
Play	yes	no	yes	yes	yes	yes	yes	no	yes	no	yes	no	no	yes

Arbres de décision : CART

(Classification and Regression Trees), L. Breiman et al., 1984

CART supporte des valeurs numériques continues pour l'attribut cible (Régression), au lieu d'avoir comme valeurs possibles un ensemble d'étiquettes.

CART impose une construction d'arbres binaires, les valeurs des attributs sont regroupées en deux sous-ensembles, le critère de Gini respectivement le critère de réduction de la variance sont utilisés pour le fractionnement dans le cas de la classification respectivement régression .

Temps de construction de l'arbre est élevé, surtout lorsque la base des exemples est de grande taille. Mais, on obtient un arbre avec des bonnes performances.

Arbres de décision : CART

$$RSS = \frac{1}{n} \sum_{j=1}^{J} n_j V_j$$

ou $V_j = \frac{1}{n_j} \sum_{i \in R_j} (y_i - \bar{y}_j)^2$ est la **variance intra-groupe**, n_j le nombre d'obervation dans chaque feuilles de l'arbre.

Arbres de décision (CHAID)

(Chi-squared Automatic Interaction Detection) – G. Kass, 1980

Technique valable pour les attributs discrets. Utilise l'écart à l'indépendance appelé encore test χ^2 (Khi-2, varie de 0 à +∞), donnée par la formule suivante (pour l'attribut x_i) :

$$\chi^2 = \sum_{c}^{C} \sum_{m}^{M_i} \frac{(n_{cm} - \frac{n_c x n_m}{n})^2}{\frac{n_c x n_m}{n}}$$

n_{cm} : nombre d'exemples de la classe C pour lesquels l'attribut x_i est égal à v_{im}

n_m : nombre d'exemples pour lesquels l'attribut x_i est égal à v_{im}

n_c : nombre d'exemples de la classe C

Arbres de décision (CHAID)

La formule précédente favorise les attributs avec plus de valeurs. Il préférable donc de normaliser avec le nombre de degrés de libertés, en prenant: $\frac{\chi^2}{n\sqrt{(C-1)(M_i-1)}} \epsilon [0,1]$

Classe/x_i	v_{i1}	...	v_{im}	...	v_{iMi}	Σ
$Classe_1$			...			
...						
$Classe_c$		...	n_{cm}	...		n_c
...			...			
$Classe_C$						
Σ			n_m			n

Arbres de décision (CHAID)

Données "Weather" (Quinlan, 1993)

Class/Outlook	Overcast	Rain	Sunny	Total
No	0	2	3	5
Yes	4	3	2	9
Total	4	5	5	14

Outlook	0.3559
Wind	0.2582

Arbres de décision

Voir le livre de Peter Harrington,
chapitre 3 "Splitting datasets one feature at a time: decision trees", page 37.