LE DATA MINING

La fouille de données

Dr. Nedioui Med Abdelhamid
Université Echahid Hama Lakhder Eloued - Algérie
nedioui3904@gmail.com

Chapitre 3

Classification par arbres de décision

1. Introduction

Parmi les algorithmes de classification, l'un des plus simples d'utilisation et d'interprétation, tout en gardant des performances très respectables, est **l'arbre de décision** . Existant sous plusieurs formes, l'arbre de décision est reconnu par le résultat de l'algorithme qui produit un modèle constitué d’un ensemble de règles de classification qu'il est possible de représenter sous forme d'arbre.

2. Définition

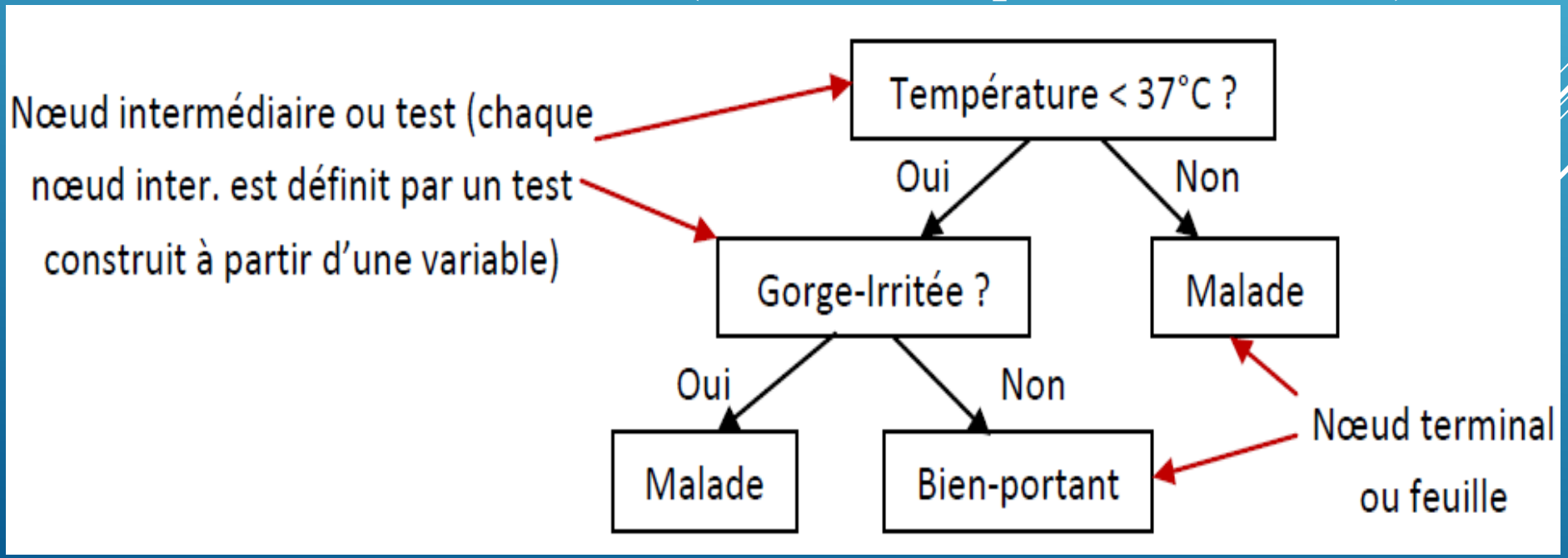
Un arbre de décision est une représentation graphique d'une procédure de classification. Il s'agit de partitionner un ensemble de données en des groupes les plus homogènes possible du point de vue de la variable à prédire. On prend en entrée un ensemble de données classées, et on fournit en sortie un arbre qui ressemble beaucoup à un diagramme d'orientation où chaque nœud final (**feuille**) représente une **décision** (**une classe**) et chaque nœud non final (**interne**) représente un **test**. Chaque feuille représente la décision d'appartenance à une classe des données vérifiant tous les tests du chemin menant de la racine à cette feuille.

2. Définition

L'intérêt des arbres de décision est en premier lieu leur **lisibilité**. En effet, il est très simple de comprendre les décisions de l'arbre une fois celui-ci créé, ce qui n'est pas toujours le cas pour les autres classifieurs que nous verrons. D'autre part, l'algorithme de création des arbres de décision fait automatiquement **la sélection d'attributs jugés pertinents**, et même sur des volumes de données importants.

2. Définition

Exemple : sur la figure suivante, la première *variable discriminante* est la température corporelle. Elle divise la population en deux classes : les personnes dont la température est supérieure à 37°C et les autres. Le processus est ensuite réitéré au deuxième niveau de l'arbre, ou les sous-populations sont segmentées à leur tour en fonction d'une autre *valeur discriminante* (dans l'exemple, c'est la toux).



2. Définition

Lors de la création de l'arbre, la première question qui vient à l'esprit est le choix de la variable de segmentation sur un sommet. Pourquoi par exemple avons-nous choisi la variable "température" à la racine de l'arbre ? Il nous *faut donc une* ***mesure*** *afin* ***d'évaluer la qualité d'une segmentation*** et *sélectionner la meilleure variable* sur chaque sommet. Ces algorithmes s'appuient notamment sur les techniques issues de la ***théorie de l'information***, et notamment la théorie de Shannon.

3. Construction d'un Arbre de Décision

Pour chaque échantillon considéré, il y a plusieurs arbres de décision qui peuvent le représenter. En général, l'arbre ayant La taille la plus petite possible est choisie parmi l'ensemble.

Le mécanisme de base de ces algorithmes consiste à choisir un attribut comme **racine** et à développer l'arbre selon les variables les plus significatives, une règle (de la forme *si condition alors conclusion*) est créé pour chaque chemin partant de la racine de l'arbre et parcourant les tests jusqu'à la feuille qui est l'**étiquette** de la classe.

3. Construction d'un Arbre de Décision

Algorithm 1 Construire_Arbre(D, T)

ENTRÉES: *D, D est l'ensemble d'apprentissage avec ses attributs et l'ensemble des classes prédéfinies.*

SORTIES: *T, l'arbre construit initialement vide*

1: Déterminer le meilleur attribut de coupure
2: Créer un noeud fils de T étiqueté par cet attribut
3: Créer autant de fils que l'attribut a de valeurs (attributs nominaux) et étiqueter les arcs par le prédicat ainsi formé
4: **pour** chaque fils *F* **faire**
5: Associer à ce fils les objets du noeud père vérifiants le prédicat
6: **si** le noeud vérifie le critère de terminaison **alors**
7: Transformer ce noeud en feuille et lui attribuer une classe
8: **sinon**
9: Construire_Arbre (D, F)
10: **finsi**
11: **fin pour**

4. Algorithme de Construction d'un AD

4.1 Algorithme ID3

ID3 construit l'arbre de décision récursivement. A chaque étape de la récursion, il calcule parmi les attributs restant pour la branche en cours, celui qui maximisera le gain d'information. C'est-à-dire l'attribut qui permettra le plus facilement de classer les exemples à ce niveau de cette branche de l'arbre. Le calcul ce fait à base de l'entropie de **Shanon** déjà présentée. L'algorithme suppose que tous les attributs sont catégoriels ; si des attributs sont numériques, ils doivent être discrétisés pour pouvoir l'appliquer

4. Algorithme de Construction d'un AD

4.2 Algorithme C4.5

C'est une amélioration de l'algorithme ID3, il prend en compte les attributs numérique ainsi que les valeurs manquantes. L'algorithme utilise la fonction du gain d'entropie combiné avec une fonction SplitInfo pour évaluer les attributs à chaque itération.

4. Algorithme de Construction d'un AD

4.3 Algorithme CART

L'algorithme CART « Classification And Regression Trees », construit un arbre de décision d'une manière analogue à l'algorithme ID3. À l'inverse à ce dernier, l'arbre de décision généré par CART est binaire et le critère de segmentation est l'indice de Gini.

À un attribut binaire correspond un test binaire, et un attribut qualitatif ayant n modalités, on peut associer autant de tests qu'il y a de partitions en deux classes, soit 2^{n-1} tests binaires possibles.

4. Algorithme de Construction d'un AD

Enfin, dans le cas d'attributs continus, il y a une infinité de tests envisageables. Dans ce cas, on découpe l'ensemble des valeurs possibles en segments.

4.4 Forêts aléatoires

Les forêts aléatoires ont été inventées par **Breiman** en 2001. Elles sont en général plus efficaces que les simples arbres de décision mais possède l'inconvénient d'être plus difficilement interprétables. Leur construction se base sur le bootstrap (ou le bagging). On subdivise l'ensemble de données en plusieurs parties par le bagging puis on apprend un arbre de décision à pou chaque partie.

5. Exemple de Construction d'un AD

Entropie

Soit un ensemble X d'exemples dont une proportion p_+ sont positifs et une proportion p_- sont négatifs.

L'entropie de X est : $\mathbf{H(X) = - p_+ \log2 p_+ - p_- \log2 p_-}$

1. $0 \leq H(X) \leq 1$; $(p_+ + p_- = 1)$

2. si $p_+ = 0$ ou $p_- = 0$, alors $H(X) = 0$: ainsi, si tous exemples sont soit tous positifs, soit tous négatifs, l'entropie de la population est nulle ;

3. si $p_+ = p_- = 0.5$, alors $H(X) = 1$: ainsi, s'il y a autant de positifs que de négatifs, l'entropie est maximale.

5. Exemple de Construction d'un AD

Gain d'information

- Le gain d'information de X par rapport a un attribut aj donne est la réduction d'entropie causée par la partition de X selon aj :

$$Gain(X, a_j) = H(X) - \sum_{v \in valeurs(a_j)} \frac{|X_{a_j=v}|}{|X|} H(X_{a_j=v})$$

Où X_{aj}=v, est l'ensemble des exemples dont l'attribut considéré aj prend la valeur v, et la notation |X| indique le cardinal de l'ensemble X.

5. Exemple de Construction d'un AD

Exemple : Soit le Jeu de données suivant :

Jour	Ciel	Température	Humidité	Vent	Jouer au tennis ?
1	Ensoleillé	Chaude	Élevée	Faible	Non
2	Ensoleillé	Chaude	Élevée	Fort	Non
3	Couvert	Chaude	Élevée	Faible	Oui
4	Pluie	Tiède	Élevée	Faible	Oui
5	Pluie	Fraîche	Normale	Faible	Oui
6	Pluie	Fraîche	Normale	Fort	Non
7	Couvert	Fraîche	Normale	Fort	Oui
8	Ensoleillé	Tiède	Élevée	Faible	Non
9	Ensoleillé	Fraîche	Normale	Faible	Oui
10	Pluie	Tiède	Normale	Faible	Oui
11	Ensoleillé	Tiède	Normale	Fort	Oui
12	Couvert	Tiède	Élevée	Fort	Oui
13	Couvert	Chaud	Normale	Faible	Oui
14	Pluie	Tiède	Élevée	Fort	Non

Sur cet exemple, on montre la construction d'un AD par ID3. Le principe de l'algorithme ID3 est de déterminer l'attribut à placer à la racine de l'AD on recherchant l'attribut qui possède le gain d'information maximum, le placer en racine, et itérer pour chaque valeur de l'attribut.

1. Construction

1- création d'une racine : les exemples n' étant ni tous positifs, ni tous négatifs, l'ensemble des attributs n' étant pas vide, on calcule les gains d'information pour chaque attribut :

Le Gain du champs "Vent" de la table est calculé comme suit:

$$Gain(X, vent) = H(X) - \frac{9}{14}H(X_{a=oui}) - \frac{5}{14}H(X_{a=non})$$

$On\ a :$

$$H(X) = -\frac{5}{14}ln_2\frac{5}{14} - \frac{9}{14}ln_2\frac{9}{14} = 0.940$$

$$H(X_{a=non}) = -(\frac{6}{8}ln_2\frac{6}{8} + \frac{2}{8}ln_2\frac{2}{8}) = 0.811$$

Et

$$H(X_{a=oui}) = -(\frac{3}{6}ln_2\frac{3}{6} + \frac{3}{6}ln_2\frac{3}{6}) = 1.0$$

$D'où :$

$$Gain(X, vent) = 0.940 - \frac{9}{14} * 0.811 - \frac{5}{14} * 1.0$$

$$= 0.048$$

Attribut	Gain
Ciel	0.246
Humidité	0.151
Vent	0.048
Température	0.029

Donc, la racine de l'AD testera l'attribut « Ciel »

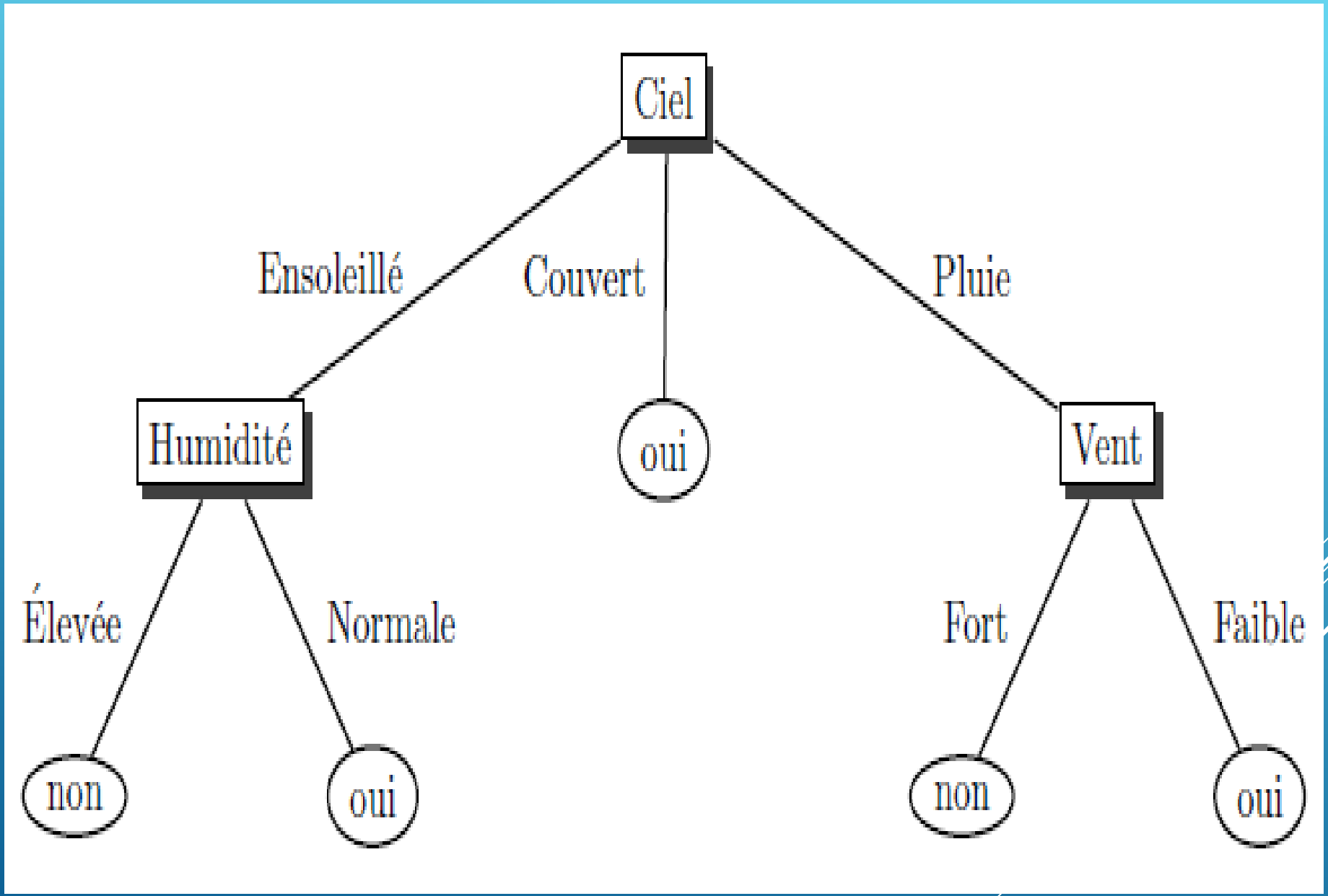
2- l'attribut «Ciel» peut prendre 3 valeurs. Pour «Ensoleille», ID3 est appelé récursivement avec 5 exemples : {x1; x2; x8; x9; x11}. Les gains d'information des 3 attributs restants sont alors :

Attribut	Gain
Humidité	0.970
Vent	0.570
Température	0.019

3. pour la branche « Pluie » partant de la racine, ID3 est appelé récursivement avec 5 exemples : {x4; x5; x6; x10; x14} on continue la construction de l'arbre de décision récursivement ;

4. pour la branche « Couvert » partant de la racine, ID3 est appelé récursivement avec 4 exemples : {x3; x7; x12; x13} ; dans ce dernier cas, tous les exemples sont positifs : on acte donc tout de suite la classe « oui » a cette feuille.

Donc, la racine de l'arbre de décision testera l'attribut « Ciel »



Arbre de décision obtenu pour l'exemple « jouer au tennis ? »

2. Interprétation

Remarquons que l'arbre de décision qui vient d'être construit nous donne des informations sur la pertinence des attributs vis-à-vis de la classe. Ainsi, l'attribut « Température » n’étant pas utilise dans l'arbre ; ceci indique que cet attribut n'est pas pertinent pour déterminer la classe. En outre, si l'attribut « Ciel » vaut « Ensoleille », l'attribut «Vent » n'est pas pertinent ; si l'attribut « Ciel » vaut « Pluie », c'est l'attribut « Humidité » qui ne l'est pas.