

Apprentissage Artificiel

Les arbres de décisions et leur apprentissage

Ewa Kijak

ESIR/Univ. Rennes

Sommaire

Principe

Construction

Elagage

Les arbres de décision : présentation

- ▶ Avez-vous de la fièvre ?
- ▶ Toussez-vous ?
- ▶ Avez-vous des maux de tête ?
- ▶ Avez-vous des douleurs musculaires ?
- ▶ Avez-vous des douleurs articulaires ?
- ▶ Avez-vous des maux de gorge ?
- ▶ Vomissez-vous ?
- ▶ Vomissez-vous du sang ?
- ▶ Avez-vous fréquenté des oiseaux ?
- ▶ ...

Décision : Vous avez la grippe aviaire...

Sommaire

Principe

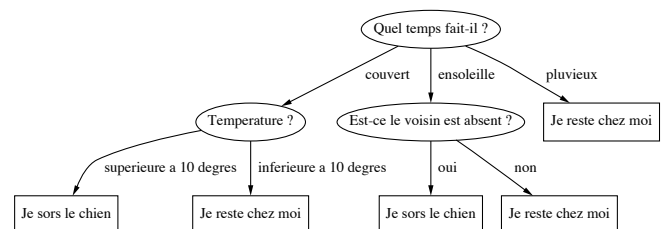
Construction

Elagage

Qu'est ce qu'un arbre de décision ?

- ▶ Soit un ensemble d'apprentissage \mathcal{S} de m exemples (\mathbf{x}, ω) décrits par d attributs $\{x_i, i = 1, d\}$ et une classe $\omega \in \mathcal{C} = \{\omega_1, \dots, \omega_C\}$.
- ▶ Arbre de décision tel que :
 - ▶ Noeud = test sur la valeur d'un ou plusieurs attributs
 - ▶ Branche = une ou plusieurs valeurs de ce test
 - ▶ Feuille = classe
- ▶ Exploitation de l'arbre de décision :
 - ▶ Classer de nouvelles données
 - ▶ Extraire un jeu de règles de classification
 - ▶ Interpréter la pertinence des attributs

L'arbre de décision pour la promenade du chien



Quelques avantages

- ▶ Le nombre moyen de tests à effectuer sur une donnée peut être réduit (si les d attributs sont tous binaires, ce nombre est limité par d).
- ▶ Structure de décision est globale à toutes les classes \Rightarrow pas de problème pour traiter directement C classes ;
- ▶ Pas besoin de tester tous les attributs de chaque donnée à chaque noeud ; dans la plupart des cas pratiques, on se limite même à un seul (sélecteur).

Problème : savoir construire un arbre de décision à partir d'un ensemble d'apprentissage et non d'une expertise.

Principe d'apprentissage d'un arbre de décision : construction descendante

Construction d'un arbre par le choix d'un attribut et d'un test sur celui-ci (sélecteur).

1. chaque noeud = test selon un critère qui a pour but de **séparer les classes** les unes des autres
2. itération jusqu'à arbre T_{max} dont chaque feuille ne couvre plus que des données **pures** (ce qui traduit un apprentissage exact des exemples).
3. **élagage** de cet arbre avec un **ensemble de validation** (pour contrebalancer le surapprentissage).

Sommaire

Principe

Construction

Principe général
Cas des attributs binaires
Autres attributs

Elagage

Algorithme de base

procedure CONSTRUIRE-ARBRE(X)

début

si tous les points de X sont de la même classe **alors**

créer une feuille de cette classe

sinon

choisir le meilleur sélecteur pour créer un noeud : il sépare X en X_d et

X_g ;

CONSTRUIRE-ARBRE(X_d)

CONSTRUIRE-ARBRE(X_g)

fin si

fin

Algorithme (suite)

Quand l'arbre est partiellement construit, chaque noeud couvre un sous-ensemble des points d'apprentissage : ceux qui satisfont tous les tests menant à ce noeud.

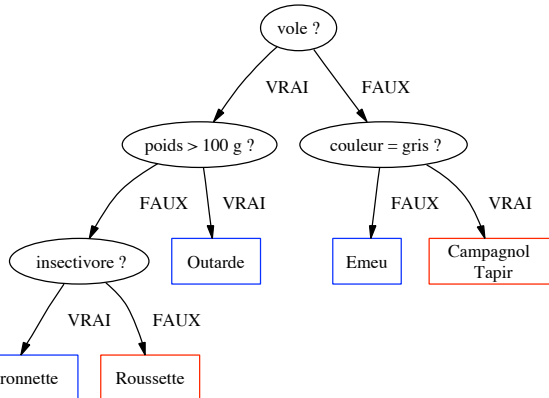
Si ce sous-ensemble n'est pas constitué de points appartenant tous à la même classe, la construction doit se poursuivre. Il faut choisir le meilleur attribut à tester.

Un exemple : oiseaux et mammifères

2 classes : **oiseaux** - **mammifères**

vole	poids	couleur	alimentation	peau	animal
OUI	1 kg	roux	granivore	plumes	Outarde
OUI	20 g	gris et jaune	insectivore	plumes	Bergeronnette
NON	100 kg	noir et blanc	omnivore	plumes	Emeu
NON	5 g	gris	granivore	poils	Campagnol
NON	40 kg	gris	herbivore	poils	Tapir
OUI	60 g	noir	frugivore	poils	Roussette

Bon ou mauvais arbre ?



Construction de l'arbre (1)

- Comment choisir les noeuds ?
 - Idée : choix de l'attribut le plus discriminant de l'ensemble des exemples considérés
- ⇒ Les attributs situés à la racine de l'arbre renferment le plus d'informations
- ⇒ L'arbre est construit de façon récursive

Construction de l'arbre (2)

- Comment détermine-t-on l'attribut le plus discriminant ?
- On cherche l'attribut (**sélecteur**) qui apporte le plus d'"**informations**" sur une classe à déterminer
- Comment qualifier numériquement ce gain d'information ?
 - Mesure de corrélation (χ^2)
 - Mesure d'hétérogénéité (Entropie, Indice de Gini)

Modélisation

Soient W et A , 2 variables aléatoires discrètes pouvant prendre les valeurs $\omega \in \mathcal{D}_W = \{\omega_1, \omega_2, \dots, \omega_n\}$ et $a \in \mathcal{D}_A$.

Remarque :

\mathcal{S} est un ensemble d'apprentissage de m exemples (\mathbf{x}, ω) décrits par d attributs $\{x_i\}_{i=1,d}$ et une classe $\omega \in \mathcal{C} = \{\omega_1, \dots, \omega_C\}$
 ⇔ m réalisations de W
 ⇔ échantillon.

⇒ sert à estimer $P(\omega)$.

Entropie conditionnelle

La variable aléatoire W possède une *entropie* $H(W)$ qui se définit par :

$$H(W) = - \sum_{\omega \in \mathcal{D}_W} P(\omega) \log P(\omega)$$

L'entropie de W *conditionnée par* A se définit comme :

$$\begin{aligned}
 H(W | A) &= \mathbb{E}_A[H(W | a)] = \sum_{a \in \mathcal{D}_A} H(W | a) P(a) \\
 &= \sum_{a \in \mathcal{D}_A} P(a) \left[\sum_{\omega \in \mathcal{D}_W} -P(\omega | a) \log(P(\omega | a)) \right] \\
 &= - \sum_{\omega, a \in \mathcal{D}_W \times \mathcal{D}_A} P(\omega, a) \log(P(\omega | a))
 \end{aligned}$$

Information mutuelle

$$I(W; A) = \sum_{\omega, a \in \mathcal{D}_W \times \mathcal{D}_A} P(\omega, a) \log \frac{P(\omega, a)}{P(\omega)P(a)}$$

D'après un résultat classique de théorie de l'information :

$$I(W; A) = H(W) - H(W | A)$$

$I(W; A)$ est :

- minimale à 0 quand $P(\omega, a) = P(\omega)P(a)$ sur tout le domaine $\mathcal{D}_W \times \mathcal{D}_A$, c'est-à-dire quand les deux distributions sont indépendantes.
- maximale quand les distributions sont parfaitement corrélées, c'est à dire quand pour tout élément ω_i du domaine \mathcal{D}_W , il existe un et un seul élément a_j du domaine \mathcal{D}_A avec : $P(\omega_i) = P(a_j)$.

Une mesure pour choisir l'attribut

On cherche parmi les d attributs (v.a. $\{A_i\}_{i=1\dots d}$), celui qui possède la plus grande corrélation avec la répartition en classes, ie qui maximise $I(W; A)$.

$$I(W; A) = H(W) - H(W | A)$$

$I(W; A)$ est une mesure du **gain** d'information.

$$\text{Maximiser } I(W; A) \Leftrightarrow \text{Minimiser } H(W | A)$$

Donc on choisit l'attribut A_i tel que :

$$i^* = \arg \min_{i=1,\dots,d} H(W | A_i)$$

Choix de l'attribut à tester - en pratique

Exemple : Cas des attributs binaires

Plaçons-nous à un noeud auquel sont attachés M points : M_1 points de la classe ω_1 , ... M_C points de la classe ω_C .

Soit un attribut binaire A qui partage chaque M_j en deux : G_j points pour $A = \text{VRAI}$ et D_j points pour $A = \text{FAUX}$.

A	VRAI	FAUX
M_1	G_1	D_1
\dots	\dots	\dots
M_C	G_C	D_C
M	G	D

→ Ici, on utilise une version simplifiée puisque $\mathcal{D}_A = \{\text{VRAI}, \text{FAUX}\}$

Choix de l'attribut à tester (suite)

Notons : $G = \sum_{j=1}^C G_j$ et $D = \sum_{j=1}^C D_j$ avec : $D + G = M$

Les quantités G_j/G et D_j/D sont des estimations des probabilités $P(W = j | A = \text{VRAI})$ et $P(W = j | A = \text{FAUX})$.

De même, G/M et D/M sont des estimations de $P(A = \text{VRAI})$ et de $P(A = \text{FAUX})$, et M_j/M de $P(W = j)$.

L'entropie de la distribution des M points sur les C classes est :

$$H(W) = - \sum_{j=1}^C \frac{M_j}{M} \log \frac{M_j}{M}$$

Choix de l'attribut à tester (suite)

Construction d'un noeud dans l'arbre = recherche parmi les d attributs de celui qui maximise l'information mutuelle avec la distribution des points d'apprentissage sur les classes.

Quand on divise les M points selon l'attribut A , la quantité d'information apportée par le fait " A vaut VRAI " se mesure par :

$$\begin{aligned} J(A = \text{VRAI}) &= H(W | A = \text{VRAI}) \\ &= - \sum_{\omega \in \mathcal{D}_W} P(\omega | A = \text{VRAI}) \log P(\omega | A = \text{VRAI}) \\ &= - \sum_{j=1}^C \frac{G_j}{G} \log \frac{G_j}{G} \end{aligned}$$

Choix de l'attribut à tester (suite)

La quantité d'information apportée par A à valeurs dans $\mathcal{D}_A = \{\text{VRAI}, \text{FAUX}\}$ vaut :

$$\begin{aligned} H(W | A) &= \sum_{a \in \mathcal{D}_A} P(a) H(W | a) \\ &= \frac{G}{M} J(A = \text{VRAI}) + \frac{D}{M} J(A = \text{FAUX}) \\ &= - \sum_{j=1}^C \frac{G_j}{M} \frac{G_j}{G} \log \frac{G_j}{G} + \frac{D_j}{M} \frac{D_j}{D} \log \frac{D_j}{D} \end{aligned}$$

Un exemple

- ▶ Dans l'exemple qui suit, le problème d'apprentissage consiste à trouver une règle de décision binaire à partir de huit exemples sur quatre paramètres binaires.
- ▶ Le problème qui se pose à un enfant qui revient de l'école est le suivant : *peut-il aller jouer chez son voisin ou pas ?*
- ▶ L'expérience, qu'il a acquise par punition-récompense sur les huit jours d'école précédents, est résumée dans le tableau des huit exemples d'apprentissage.

Un exemple

	Mes Devoirs sont-ils Finis ? (DF)	Maman est-elle de Bonne Humeur ? (MBH)	Est-ce qu'il Fait Beau ? (FB)	Mon Goûter est-il Pris ? (GP)	DECISION
1	VRAI	FAUX	VRAI	FAUX	OUI
2	FAUX	VRAI	FAUX	VRAI	OUI
3	VRAI	VRAI	VRAI	FAUX	OUI
4	VRAI	FAUX	VRAI	VRAI	OUI
5	FAUX	VRAI	VRAI	VRAI	NON
6	FAUX	VRAI	FAUX	FAUX	NON
7	VRAI	FAUX	FAUX	VRAI	NON
8	VRAI	VRAI	FAUX	FAUX	NON

Exercice

- ▶ Que vaut $H(W)$?
- ▶ Quel est le test à la racine de l'arbre ?
 - ▶ Construire les distributions conjointes (tableaux de contingence) entre la distribution des classes et chacun des attributs
 - ▶ Calculer les entropies conditionnelles $H(W | DF)$, $H(W | FB)$, $H(W | MBH)$ et $H(W | GP)$
- ▶ Donner la nouvelle répartition des exemples entre les sous-arbres gauche et droit.

Distributions conjointes

DF	VRAI	FAUX	GP	VRAI	FAUX
OUI	3	1	OUI	2	2
NON	2	2	NON	2	2
8	5	3	8	4	4

MBH	VRAI	FAUX	FB	VRAI	FAUX
OUI	2	2	OUI	3	1
NON	3	1	NON	1	3
8	5	3	8	4	4

Cas d'un attribut nominal (1)

- ▶ Le test que l'on construit peut consister à opposer une valeur à toutes les autres : on se ramène au cas binaire.
 - ▶ Par exemple, soit un attribut couleur prenant ses valeurs dans l'ensemble $\{bleu, rouge, vert, jaune\}$
→ solution simple = l'éclater en 4 attributs binaires, du type Couleur-rouge, qui est VRAI ou FAUX sur chaque donnée d'apprentissage.
- ⇒ transformation d'un attribut nominal à k valeurs possibles en k attributs binaires que l'on traite indépendamment.
- ▶ Cette technique a l'inconvénient d'oublier la signification globale de l'attribut.

Cas d'un attribut nominal (2)

Autre solution :

- ▶ calculer directement l'information mutuelle entre cet attribut et l'ensemble des classes (les deux variables à valeurs discrètes)
- ▶ Si celle-ci est la meilleure pour tous les attributs, on crée alors un noeud **non binaire** dans l'arbre de décision.

Cas d'un attribut nominal (3)

Exemple :

- ▶ considérons un attribut A_i with $\mathcal{D}_{A_i} = \{blue, white, red\}$
- ▶ et sa distribution :

A_i	$\omega = \text{true}$	$\omega = \text{false}$	total
blue	2	3	5
white	4	0	4
red	3	2	5
total	9	5	14

- ▶ alors :

$$H(W|A_i) = \frac{5}{14} \left(-\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} \right) + \frac{4}{14} \left(-\frac{4}{4} \log_2 \frac{4}{4} - \frac{0}{4} \log_2 \frac{0}{4} \right) + \frac{5}{14} \left(-\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} \right)$$

Cas d'un attribut continu

- ▶ Le nombre de données d'apprentissage étant fini, le nombre de valeurs que peut prendre cet attribut est en pratique fini.
- ▶ De plus, ses valeurs sont ordonnées. Le sélecteur consistera donc à comparer les valeurs à un seuil, pour construire un noeud binaire.
- ▶ Pour un attribut A_i continu, on procède alors ainsi : on trie les points d'apprentissage selon la valeur de cet attribut, puis on cherche le seuil $s(A_i)$ qui minimise l'un des critères précédents.
- ▶ Ceci nécessite, pour les $p + n$ données, l'examen de $p + n - 1$ seuils : par exemple les valeurs médianes entre deux points d'apprentissage dans leur liste triée.

Gain Ratio : Attributs nominaux/continus

- ▶ Dans le cas d'attributs nominaux ou continus avec une forte arisé (i.e. un grand nombre de valeurs possibles), ceux-ci sont mécaniquement favorisés par le calcul du gain d'information.
- ▶ Une façon de prévenir cet effet négatif est de choisir le sélecteur qui maximise le *rapport de gain* et non le gain :

$$\text{Gain Ratio}(W,A) = \frac{\text{Gain}(W,A)}{\text{SplitInfo}(W,A)} = \frac{H(W) - H(W|A)}{-\sum_{a \in \mathcal{D}_A} \frac{|S_a|}{|S|} \log_2 \frac{|S_a|}{|S|}}$$

- ▶ $\text{Gain}(W,A)$ = information mutuelle
- ▶ $|S_a|$ est le sous-ensemble des exemples ayant l'attribut $A = a$.
- ▶ $\text{SplitInfo}(W,A) = H(A)$ est aussi appelé information intrinsèque *intrinsic information*.
- ▶ Cette mesure pénalise la sélection d'attributs dont les valeurs sont uniformément distribuées.

Sommaire

Principe

Construction

Elagage

L'élagage : généralités

- ▶ Poursuite de l'algorithme de construction jusqu'à son terme naturel \Rightarrow arbre T_{max} :
 - ▶ feuilles \Leftrightarrow classes parfaitement homogènes
 - ▶ apprentissage par coeur et donc risque de sous-estimation de la probabilité d'erreur par le taux d'erreur apparent (qui vaut 0).
- ▶ Nombre de noeuds de l'arbre de décision = critère de complexité simple et efficace.
- ▶ Technique pour contrôler la taille de l'arbre = chercher la valeur "optimale" k_0 du nombre de noeuds.

L'élagage : principe

L'élagage consiste à simplifier un arbre de décision en coupant des branches.

Il possède 2 objectifs :

1. simplifier l'arbre de décision
2. diminuer le sur-apprentissage, i.e. augmenter la capacité de généralisation, et par là-même, diminuer le taux d'erreur.

Deux possibilités :

- ▶ élagage lors de la construction
- ▶ élagage après la construction

Elagage lors de la construction : une mauvaise solution

- ▶ Une solution simple consiste à cesser de diviser un noeud quand l'homogénéité des points qu'il domine est, non pas parfaite, mais "suffisante".
- ▶ Une fois sélectionné le meilleur attribut, on regarde si la valeur du critère de la division est inférieure à un certain seuil.
- ▶ Prise en compte d'un critère local à la feuille

Elagage après la construction par un ensemble de validation

Technique meilleure en théorie et en pratique :

1. Laisser se terminer l'algorithme de construction (arbre T_{max})
 2. Construire une série d'arbres plus simples par regroupement des noeuds,
 3. Choisir le meilleur d'entre eux avec un ensemble de validation.
- ▶ L'ensemble d'apprentissage est alors coupé en deux parties :
 - ▶ une pour construire T_{max} (ensemble d'apprentissage proprement dit) et proposer des élagages,
 - ▶ l'autre (ensemble de validation) pour choisir le meilleur parmi les élagages proposés.
 - ▶ Le problème est donc de remplacer certains des noeuds de T_{max} par des feuilles.
 - ▶ La classe affectée à une feuille f ainsi créée sera ensuite choisie comme celle qui est majoritaire dans les feuilles de T_{max} dominées par le noeud que l'on vient de remplacer par f .

Elagage par un ensemble de validation (suite)

L'algorithme optimal consisterait à calculer le taux d'erreur de l'ensemble de validation sur tous les arbres qu'il est possible d'obtenir par élagage de T_{max} . Mais leur nombre croît très rapidement avec la taille de T_{max} , mesurée en nombre de noeuds.

⇒ utilisation de solutions sous-optimales, dont la plus classique consiste à construire une séquence d'arbres par élagages successifs :

$$S = \{T_{max}, T_1, \dots, T_k, \dots, T_m\},$$

où T_m est constitué d'une seule feuille comprenant les m points d'apprentissage.

Pour passer de T_k à T_{k+1} , il faut transformer en feuille(s) un (ou plusieurs) noeud(s) dans T_k .

Elagage par un ensemble de validation (suite)

La technique généralement utilisée est de choisir le(s) noeud(s) qui minimise(nt) sur l'ensemble des noeuds de T_k le critère suivant :

$$\varpi(T_k, d) = \frac{MC(d, k) - MCT(d, k)}{n(k) \cdot (nt(d, k) - 1)}$$

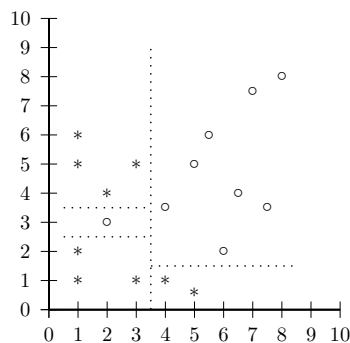
- ▶ $MC(d, k)$ est le nombre de points de l'ensemble d'apprentissage mal classés par le noeud d de T_k quand on fait l'hypothèse qu'il a été transformé en feuille
- ▶ $MCT(d, k)$ le nombre de points de l'ensemble d'apprentissage mal classés par les feuilles de T_k situées sous le noeud d
- ▶ $n(k)$ le nombre de feuilles de T_k
- ▶ $nt(d, k)$ le nombre de feuilles du sous-arbre de T_k situé sous le noeud d .

Elagage par un ensemble de validation (suite)

Ce critère permet donc d'élaguer dans T_k un ou plusieurs noeuds de façon à ce que l'arbre obtenu, noté T_{k+1} , possède le meilleur compromis entre taille et taux d'erreur.

Finalement, la suite $S = \{T_{max}, T_1, \dots, T_k, \dots, T_m\}$ que l'on construit ainsi possède un élément T_{k_0} pour lequel le nombre d'erreurs commises sur l'ensemble de validation est minimal.

C'est cet arbre-là qui sera finalement retenu par la procédure d'élagage.



L'arbre de décision géométrique T_{max} .

Exercice

- ▶ Quel est l'arbre de décision T_{max} associé à la représentation géométrique précédente ?

En appelant n_1 le noeud racine de T_{max} , n_2 et n_3 ses fils Gauche et Droit et n_4 son dernier noeud intérieur (le fils gauche de n_2) :

- ▶ Calculer $\varpi(T_{max}, n_1)$, $\varpi(T_{max}, n_2)$, $\varpi(T_{max}, n_3)$, $\varpi(T_{max}, n_4)$
- ▶ Donner l'arbre T_1 résultant de l'élagage de T_{max}

De la même façon :

- ▶ Donner l'arbre T_2 résultant de l'élagage de T_1

Elagage

Choix d'un arbre dans la suite construite $S = \{T_{max}, T_1, T_2, T_m\}$ (avec T_m composé d'une seule feuille et qui ne représente que la probabilité a priori des classes dans l'ensemble d'apprentissage) :

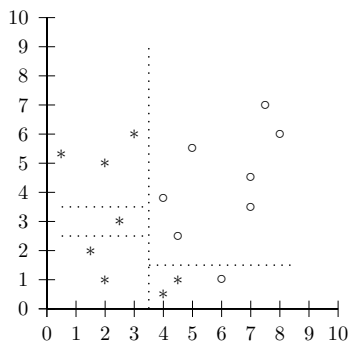
- Test de l'ensemble de validation sur tous les arbres,
- Choix de l'arbre qui possède la meilleure estimation de taux d'erreur de classification.

La procédure d'élagage sera alors terminée.

Exercice

On reprend l'arbre T_{max} précédent. Soit l'ensemble de validation représenté à la figure suivante.

1. Calculez l'erreur apparente et l'estimation de l'erreur réelle pour chacun des arbres T_{max} , T_1 et T_2 précédemment obtenus.
2. Tracez les courbes d'erreur apparente et d'estimation d'erreur réelle en fonction de la complexité des arbres.



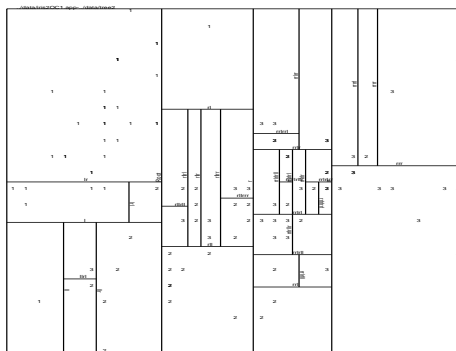
L'ensemble de validation sur l'arbre T_{max} .

Exemple

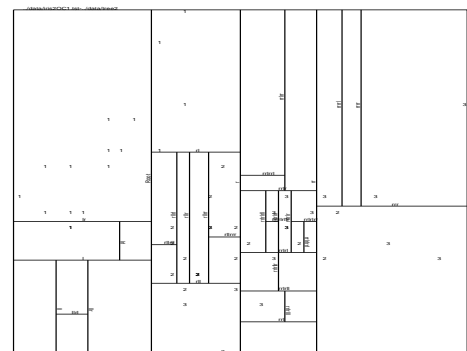
Soit l'ensemble des *Iris de Fisher*. Chaque exemple est décrit par 2 attributs : la longueur et la largeur des sépales. Il y a 3 classes possibles désignées dans les représentations qui suivent par $\{1, 2, 3\}$.

Le jeu de données contient 150 exemples :

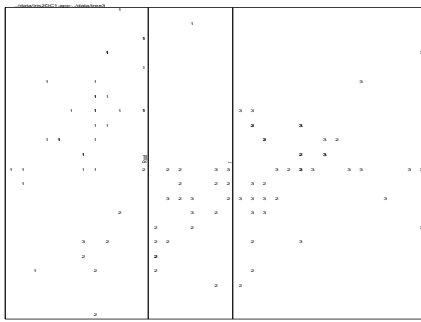
- 100 sont utilisés pour l'apprentissage
- les 50 restantes pour le test



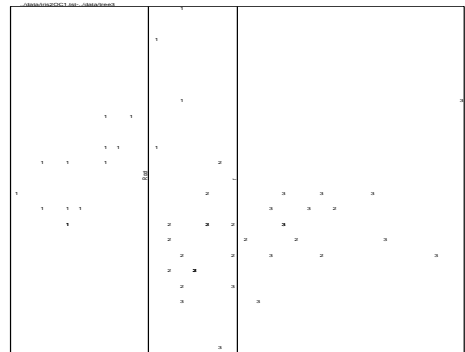
Les 100 données d'apprentissage sur l'arbre non élagué.
Taux d'erreur apparent : 2 %



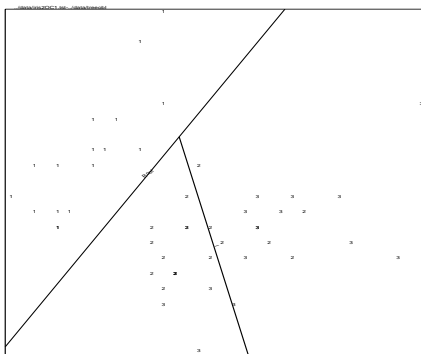
Les 50 données de test sur l'arbre non élagué.
Estimation de l'erreur réelle : 32 %



Les 90 données d'apprentissage restantes
sur l'arbre élagué par 10 données.
Taux d'erreur apparent : 25 %



Les 50 données de test sur l'arbre élagué.
Estimation de l'erreur réelle : 26 %



Les 50 données de test sur l'arbre oblique élagué.
Estimation de l'erreur réelle : 26 %

Exercice 1

On considère le jeu de données de la table ci-après : un ensemble de jours (un jour = un exemple), chacun caractérisé par un numéro et ses conditions météorologiques (température, humidité de l'air, force du vent, ciel), l'attribut cible étant "jouer au tennis?", dont les valeurs possibles sont oui et non.

1. Quelle est l'entropie de la population ?
2. Quel est l'attribut dont le gain d'information est maximal dans la population ?
3. Construire l'arbre de décision.

Exercice 1

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

Exercice 2

On prend maintenant en compte des attributs numériques, c'est-à-dire, des attributs dont l'arité est élevée (voire infinie). On considère donc le nouveau jeu de données du tableau ci-après dans lequel les attributs "Température" et "Humidité" ont été numérisés.

Dans ce cas, un noeud de l'arbre de décision peut contenir un test du fait que la valeur d'un attribut numérique est inférieure à un certain seuil : cela correspond donc à un nouveau pseudo-attribut binaire.

On ne considère que les exemples dont l'attribut "Ciel" vaut "Ensoleillé".

Exercice 2

	Ciel	Temp.	Humidité	Vent	Jouer au tennis ?
1	Ensoleillé	27.5	85	Faible	Non
2	Ensoleillé	25.0	90	Fort	Non
3	Couvert	26.5	86	Faible	Oui
4	Pluie	20.0	96	Faible	Oui
5	Pluie	19.0	80	Faible	Oui
6	Pluie	17.5	70	Fort	Non
7	Couvert	17.0	65	Fort	Oui
8	Ensoleillé	21.0	95	Faible	Non
9	Ensoleillé	19.5	70	Faible	Oui
10	Pluie	22.5	80	Faible	Oui
11	Ensoleillé	22.5	70	Fort	Oui
12	Couvert	21.0	90	Fort	Oui
13	Couvert	25.5	75	Faible	Oui
14	Pluie	20.5	91	Fort	Non

Exercice 2

- Déterminer le seuil s pour l'attribut "Température" pour partitionner cet ensemble d'exemples.
On utilise les règles suivantes :
 - Ne pas séparer deux exemples successifs ayant la même classe,
 - Si on coupe entre deux valeurs v et w ($v < w$) de l'attribut, le seuil s est fixé à w ,
 - Choisir s de telle manière que le gain d'information soit maximal.
- Construire l'arbre de décision pour les attributs numériques.