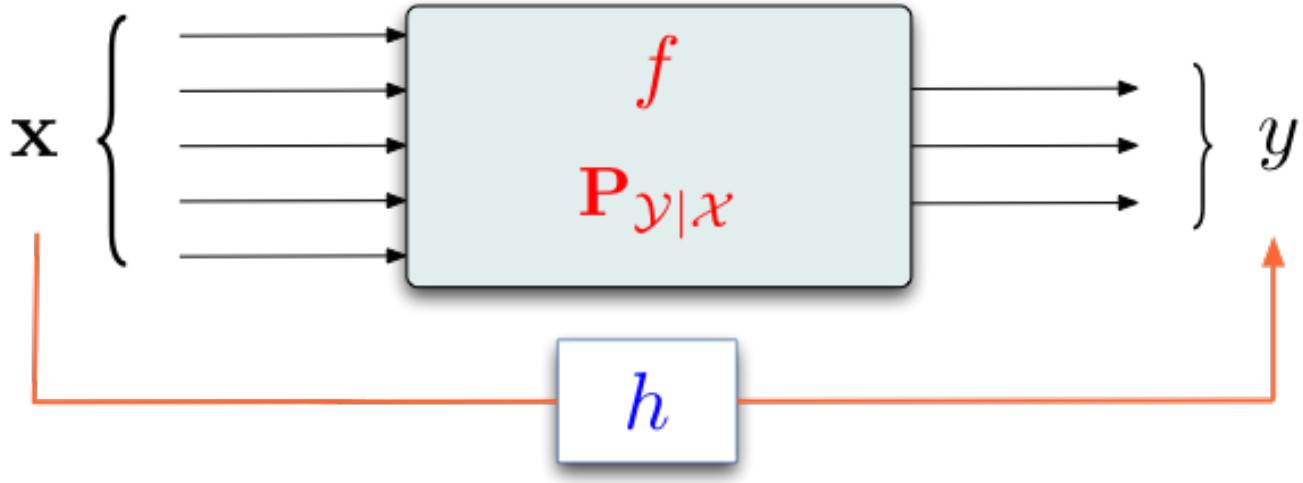
Learning
Decision Trees

Antoine Cornuéjols

AgroParisTech – INRAE MIA Paris-Saclay

antoine.cornuejols@agroparistech.fr

Apprentissage supervisé



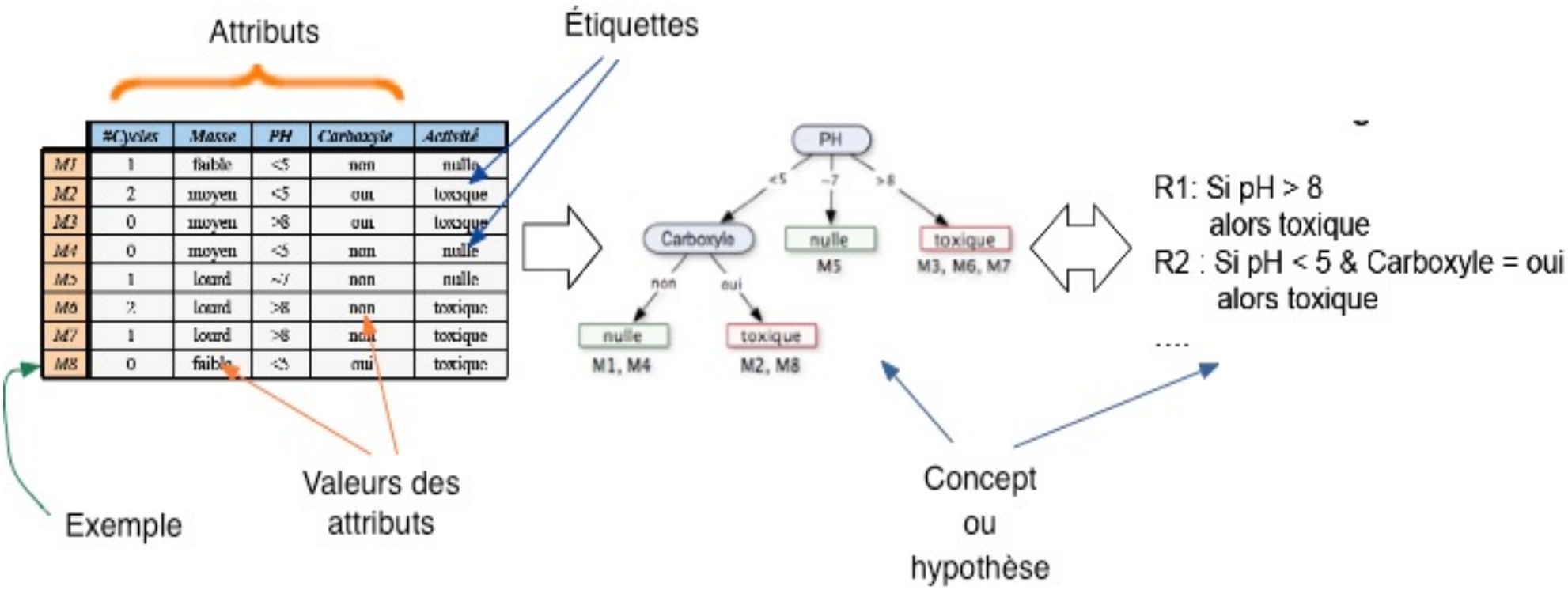
À partir :

- d'un échantillon d'apprentissage $S_m = \langle (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m) \rangle$
- de connaissances préalables sur le type de dépendances sur $\mathcal{X} \times \mathcal{Y}$

Trouver :

- une fonction h
- permettant la prédiction de y pour une nouvelle entrée x $h(\mathbf{x}) \approx y \ (= f(\mathbf{x}))$

Induction supervisée



Les données : organisation et types

Identifieur	Genre	Age	Niveau études	Marié ?	Nb enfants	Revenu	Profession	A prospecter ?
I_21	M	43	Bac+5	Oui	3	55 000	Architecte	OUI
I_34	M	25	Bac+2	Non	0	21 000	Infirmier	NON
I_38	F	34	Bac+8	Oui	2	35 000	Chercheuse	OUI
I_39	F	67	Bac	Oui	5	20 000	Retraitée	NON
I_58	F	56	CAP	Oui	4	27 000	Ouvrière	NON
I_73	M	40	Bac+3	Non	2	31 000	Commercial	OUI
I_81	F	51	Bac+5	Oui	3	75 000	Chef d'entreprise	OUI

Les données : organisation et types

Identifieur	Genre	Age	Niveau études	Marié ?	Nb enfants	Revenu	Profession	A prospecter ?
I_21	M	43	Bac+5	Oui	3	55 000	Architecte	OUI
I_34	M	25	Bac+2	Non	0	21 000	Infirmier	NON
I_38	F	34	Bac+8	Oui	2	35 000	Chercheuse	OUI
I_39	F	67	Bac	Oui	5	20 000	Retraitée	NON
I_58	F	56	CAP	Oui	4	27 000	Ouvrière	NON
I_73	M	40	Bac+3	Non	2	31 000	Commercial	OUI
I_81	F	51	Bac+5	Oui	3	75 000	Chef d'entreprise	OUI

Exemple
(*example, instance*)

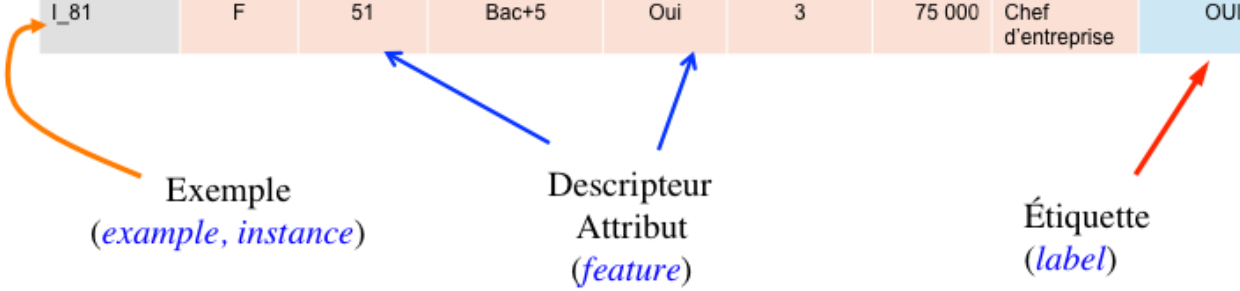
Descripteur
Attribut
(*feature*)

Étiquette
(*label*)

Les données

- **Vectorielles**
- Séquences
- Structurés
- Temporelles
- Spatiales

Identifieur	Genre	Age	Niveau études	Marié ?	Nb enfants	Revenu	Profession	A prospecter ?
I_21	M	43	Bac+5	Oui	3	55 000	Architecte	OUI
I_34	M	25	Bac+2	Non	0	21 000	Infirmier	NON
I_38	F	34	Bac+8	Oui	2	35 000	Chercheuse	OUI
I_39	F	67	Bac	Oui	5	20 000	Retraitée	NON
I_58	F	56	CAP	Oui	4	27 000	Ouvrière	NON
I_73	M	40	Bac+3	Non	2	31 000	Commercial	OUI
I_81	F	51	Bac+5	Oui	3	75 000	Chef d'entreprise	OUI



Outline

1. Decision trees

2. Learning decision trees

3. Various problems and their solutions

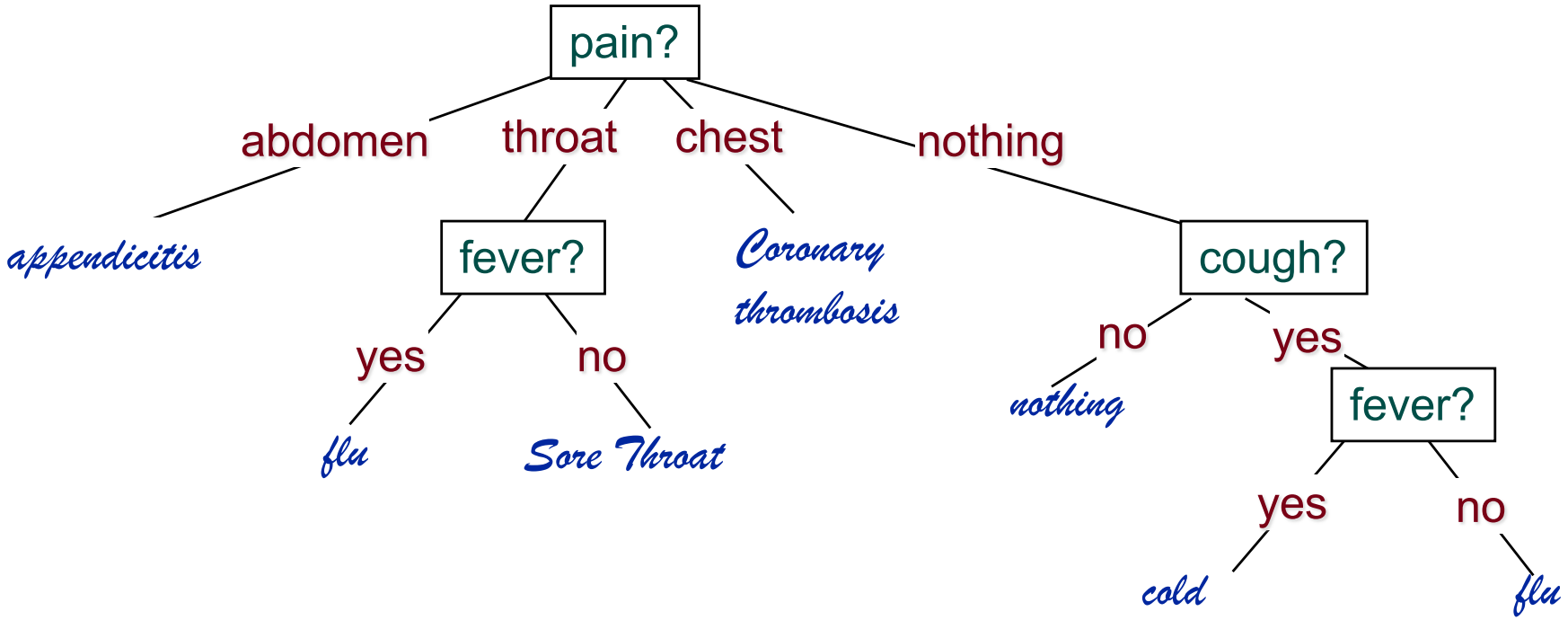
The learning task

- Task
  - Learning a classification function
- Protocole
  - Supervised learning using greedy iterative approximation
- Performance measure
  - Prediction error rate
- Inputs
  - Vectorial
- Hypothesis space
  - Decision trees

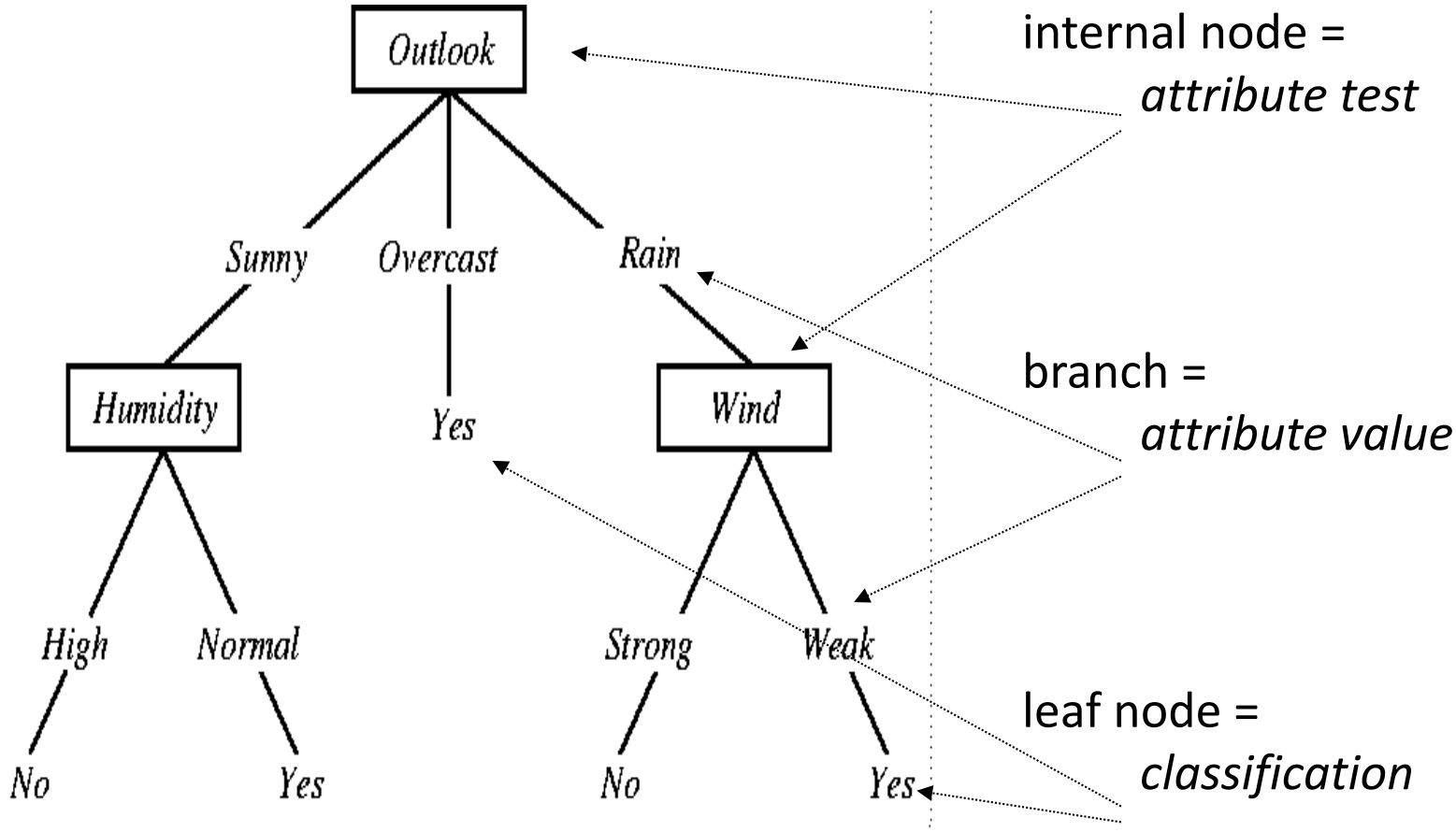
Decision trees

The structure of decision trees

- The internal nodes test attribute values
- A branch for each possible value of the tested attribute
- Leaves correspond to classes (labels)



Les arbres de décision : représentation



©Tom Mitchell, McGraw Hill, 1997

Expressiveness of decision trees

- **Any boolean function** can be represented by a decision tree
  - Reminder: with 6 boolean attributes, there exists approximately 6 . 10^9 boolean functions
- Some functions can require very large decision trees
  - E.g. The "parity" function and the "majority" function may require exponentially large trees
  - Other functions can be represented with one node
- Limited to **propositional logic**. No relational representation
- A tree corresponds to a disjunction of rules

DT =
- (**if** feather = no **then** label = **not bird**)
- or (**if** feather = yes & color = brown **then** label = **not bird**)
- or (**if** feather = yes & color = B&W **then** label = **bird**)
- or (**if** feather = yes & color = yellow **then** label = **bird**)

Arbre de décision : exemple

Détection du spam

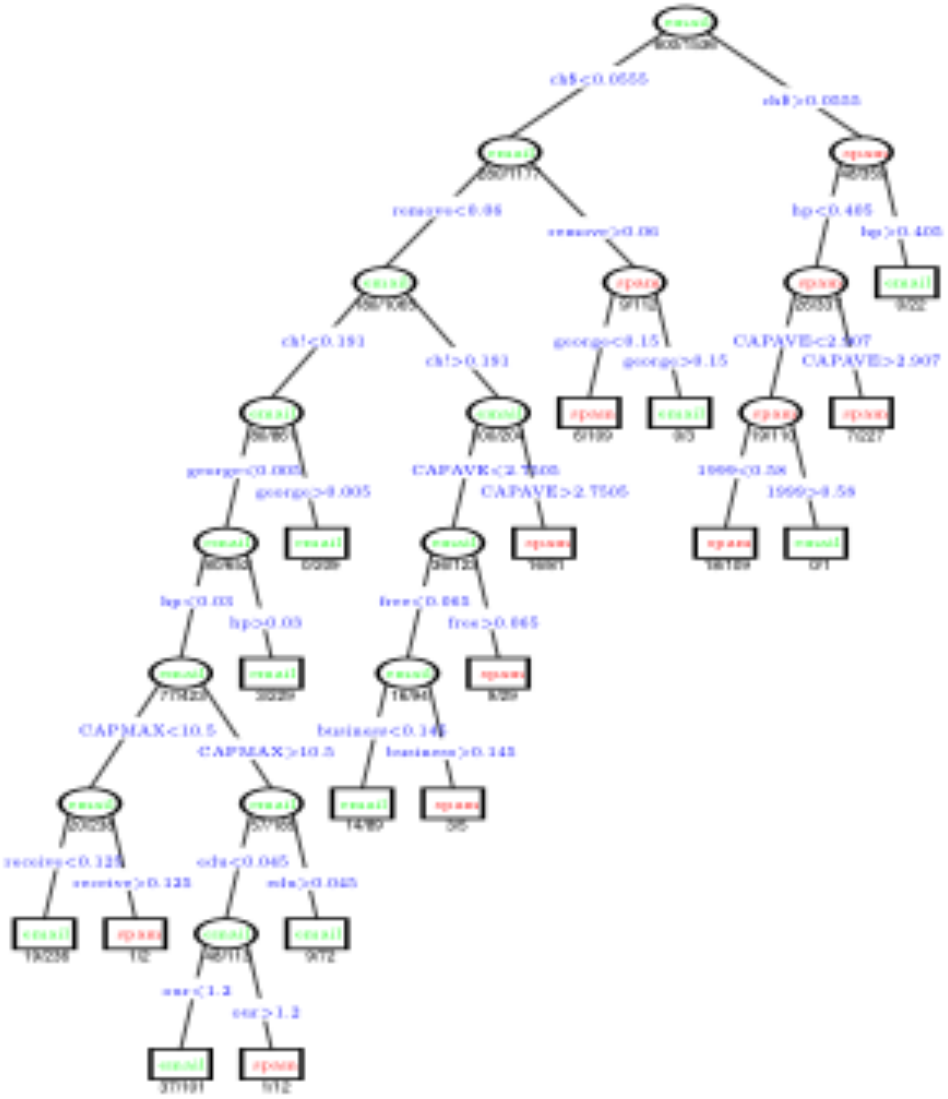
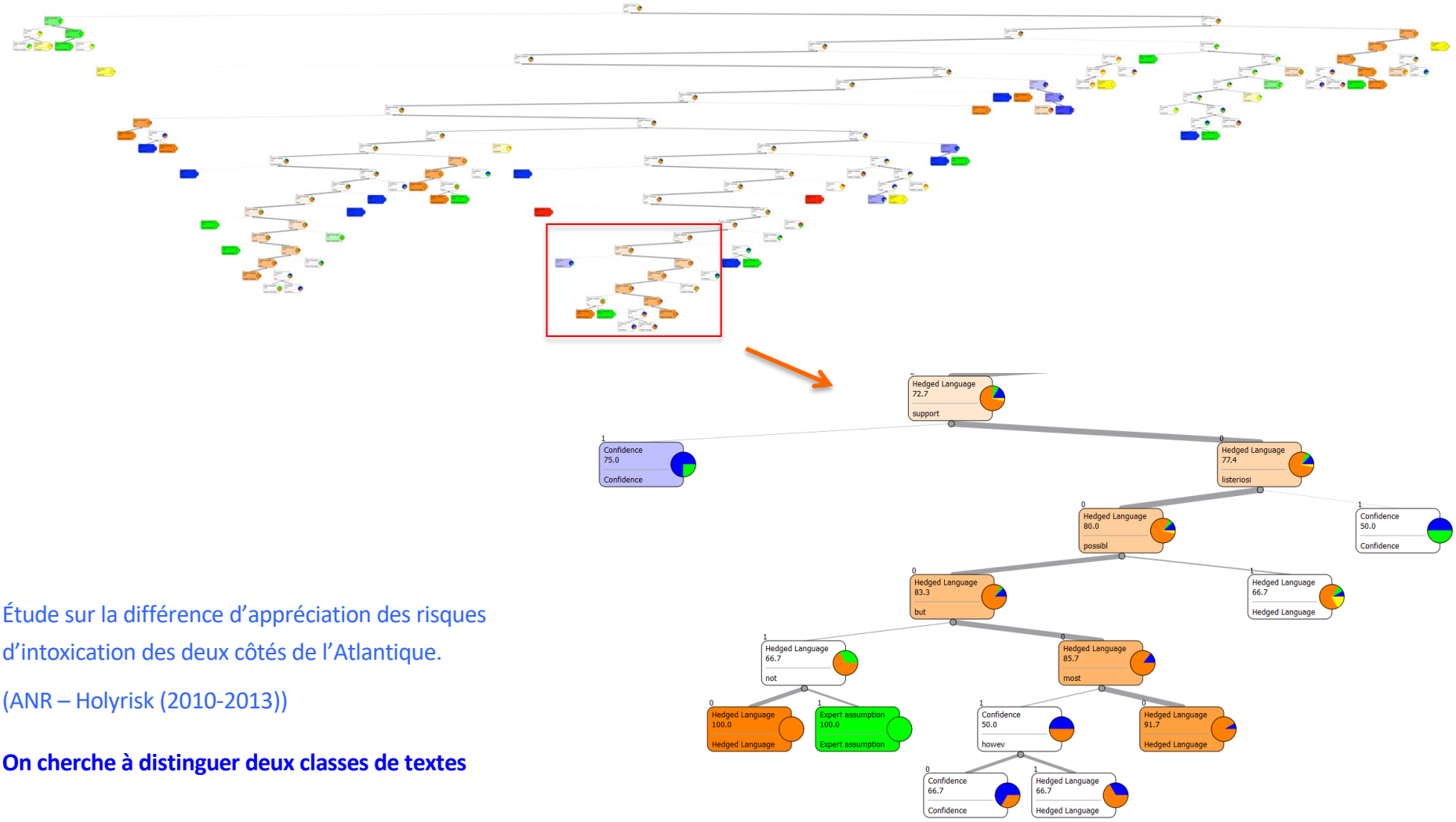


FIGURE 9.5. *The pruned tree for the* spam *example. The split variables are shown in blue on the branches, and the classification is shown in every node. The numbers under the terminal nodes indicate misclassification rates on the test data.*

Exemple : arbre de décision



Étude sur la différence d'appréciation des risques d'intoxication des deux côtés de l'Atlantique.

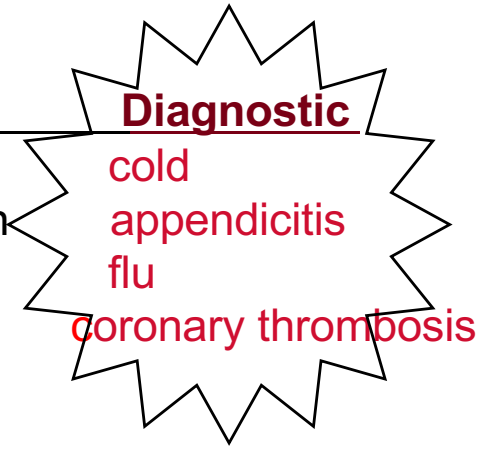
(ANR – Holyrisk (2010-2013))

On cherche à distinguer deux classes de textes

Data sets

- A training data set

	Cough	Fever	Weight	Pain	Diagnostic
Marie	no	yes	normal	throat	cold
Fred	no	yes	normal	abdomen	appendicitis
Julie	yes	yes	skinny	no	flu
Elvis	yes	no	overweight	chest	coronary thrombosis

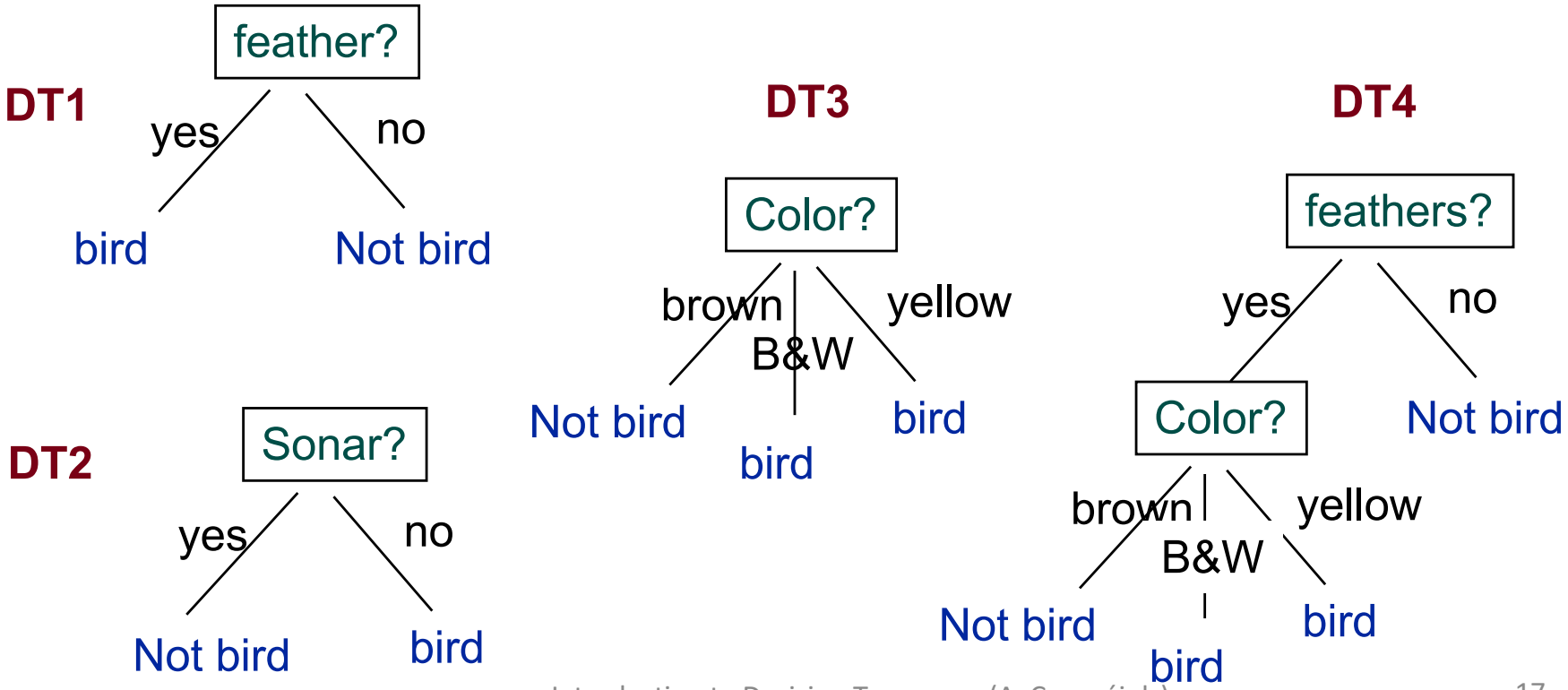


- How to learn a decision tree?

Which tree to select from *H*?

	Color	Wings	Feathers	Sonar	**Concept**
x23	yellow	yes	yes	no	bird
x24	B&W	yes	yes	no	bird
x25	brown	yes	no	yes	Not bird

There exist **four trees** consistant with the data set



L'espace de recherche

- Toutes les séquences possibles de tous les tests (éventuellement répétés)
- Arbre de recherche GIGANTESQUE
  - Nombre de Catalan (*n* nœuds d'au plus deux descendants)

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

n = 10 => 16 796 arbres binaires

n = 20 => 6.56 x 10^9 arbres binaires

The search space

- Number of trees = Catalan's number

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

n attributes of
branching factor = 2

Huge search space!

How to explore it?

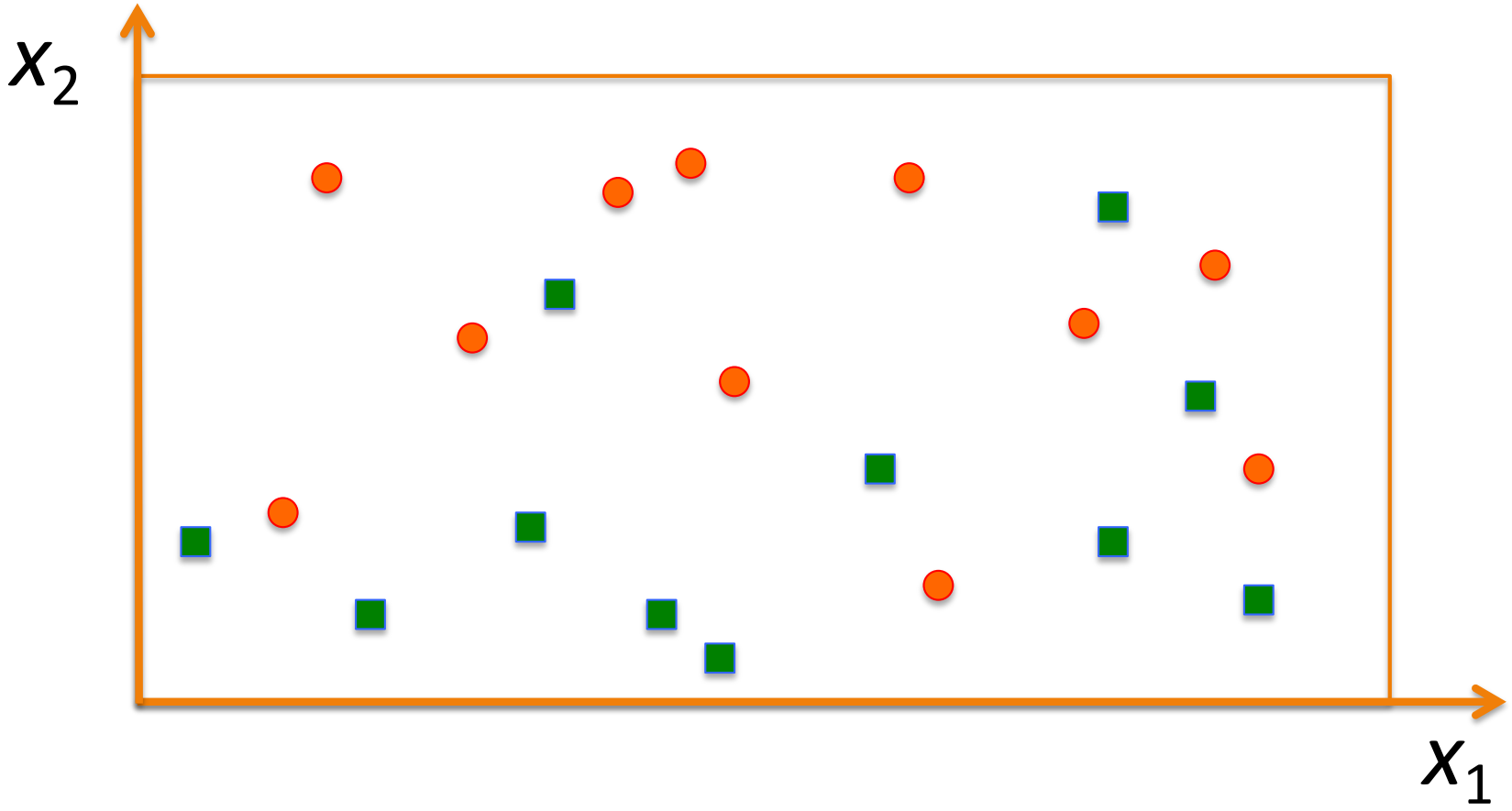
Learning decision trees

A greedy iterative top-down strategy

- Principle

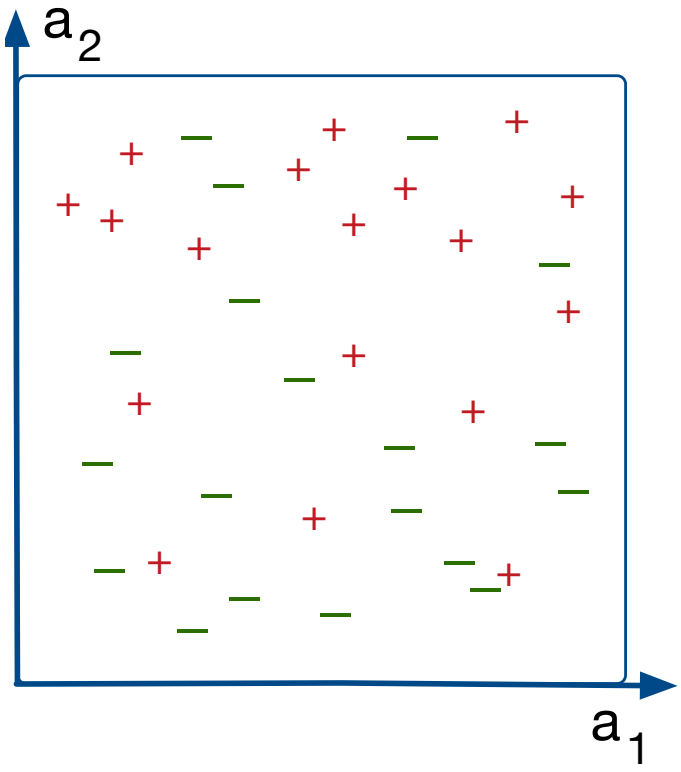
1. Select the **best attribute**
2. **Grow the tree** according to the choice
3. Now there are subsets of the data set at the leaves
4. Return to 1 **until stopping criterion**

Entropie



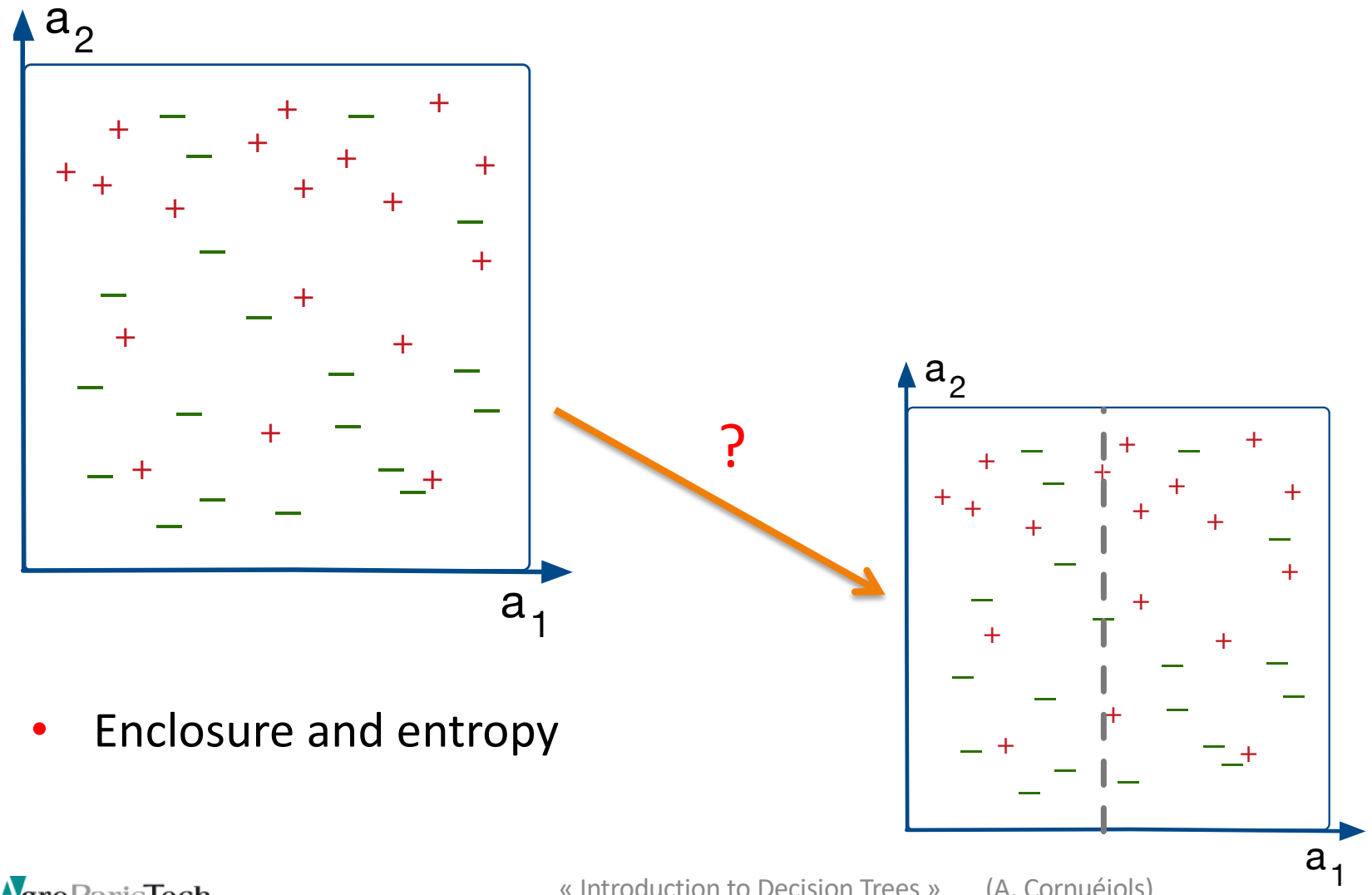
...

How to chose the best separator?



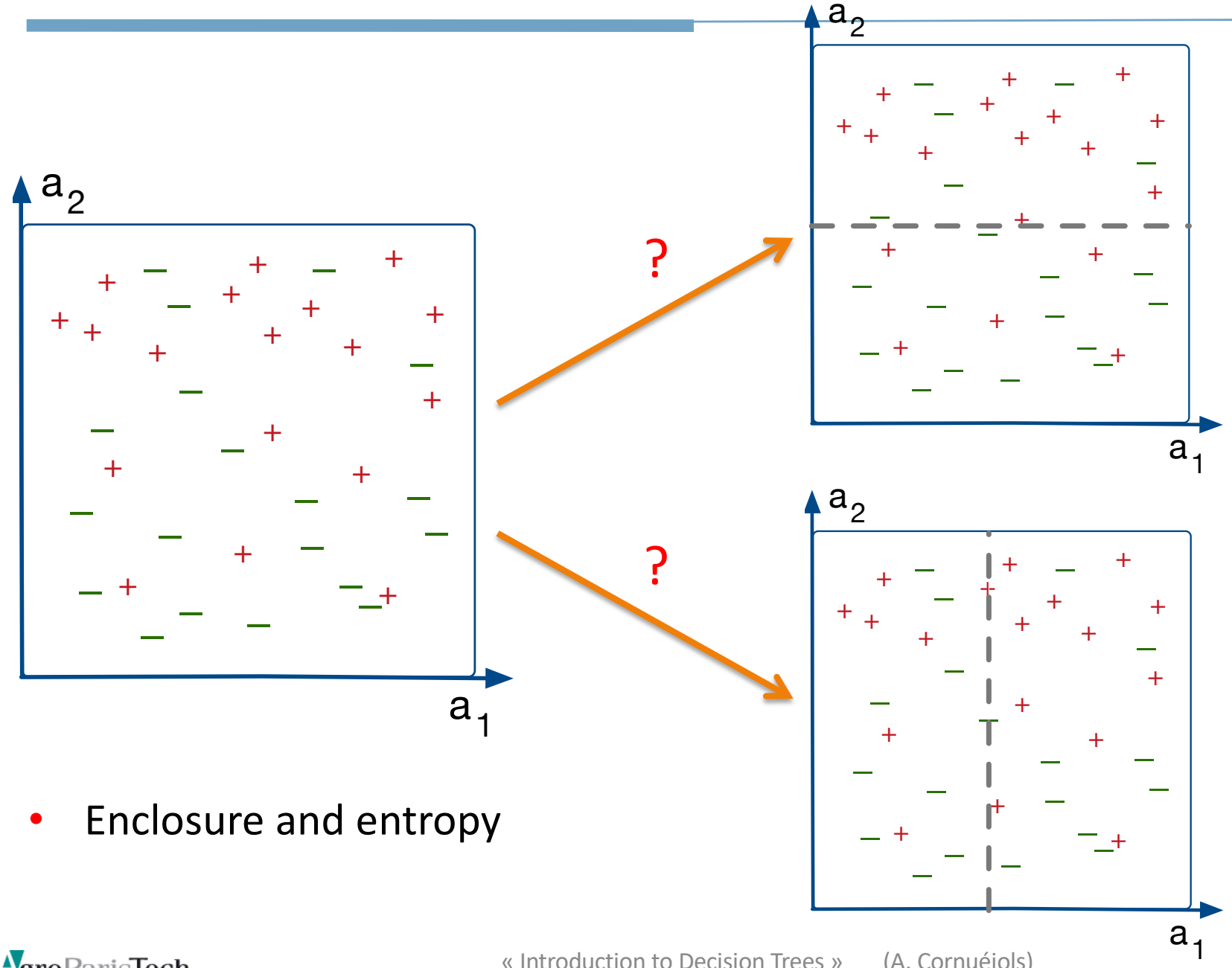
- Enclosure and entropy

How to chose the best separator?

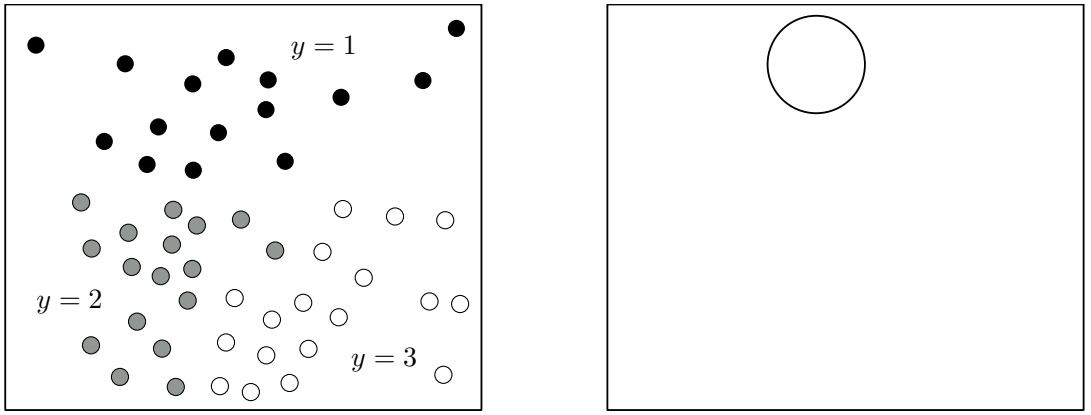


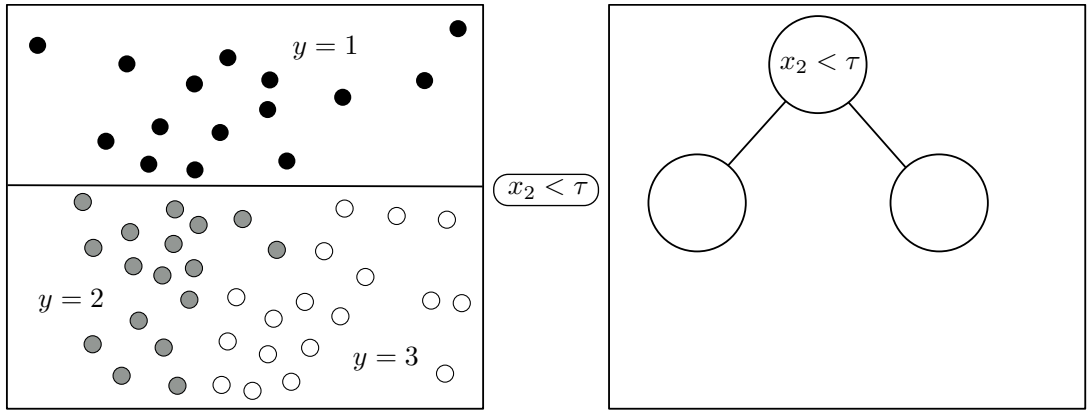
- Enclosure and entropy

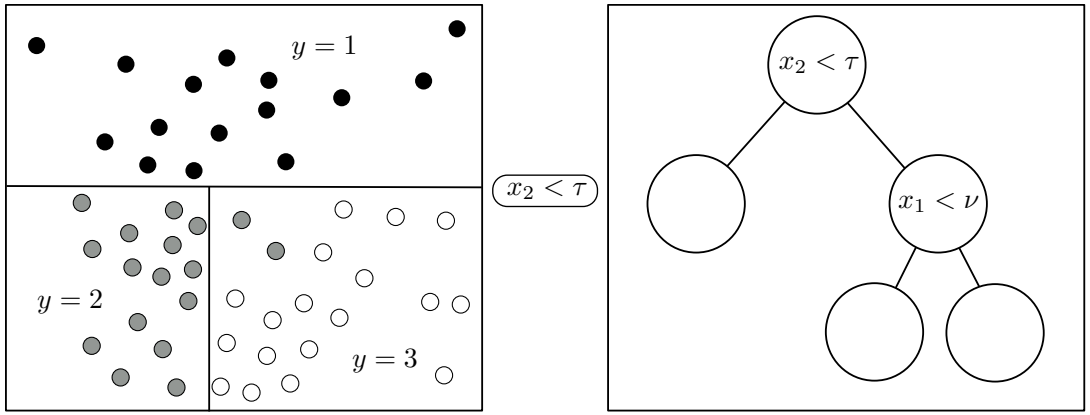
How to chose the best separator?



- Enclosure and entropy


$y = 1$
$y = 2$
$y = 3$


$y = 1$
$y = 2$
$y = 3$
$x_2 < \tau$
$x_2 < \tau$


$y = 1$
$y = 2$
$y = 3$
$x_1 < \nu$
$x_2 < \tau$
$x_2 < \tau$
$x_1 < \nu$

- ...

Impurity measure: the entropy criterion

- Boltzmann entropy
- … used by Shannon
  - In 1949 Shannon proposed an entropy measure valid for discrete probability.
  - It expresses the quantity of information, that is the number of bits required to specify the distribution
  - The information entropy is:

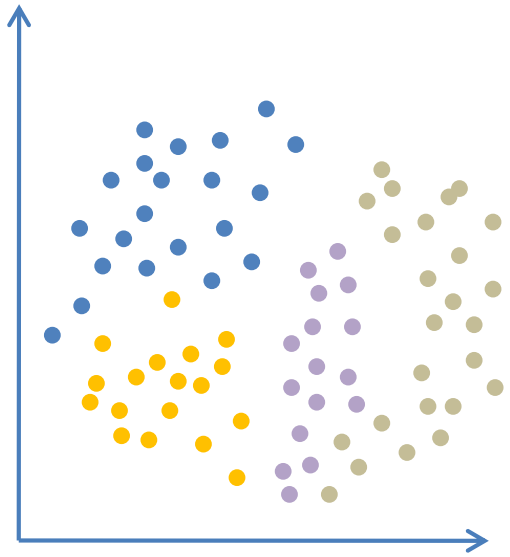
$$I = -\sum_{i=1..k} p_i \times \log_2(p_i)$$

where p_i is the probability of class C_i.

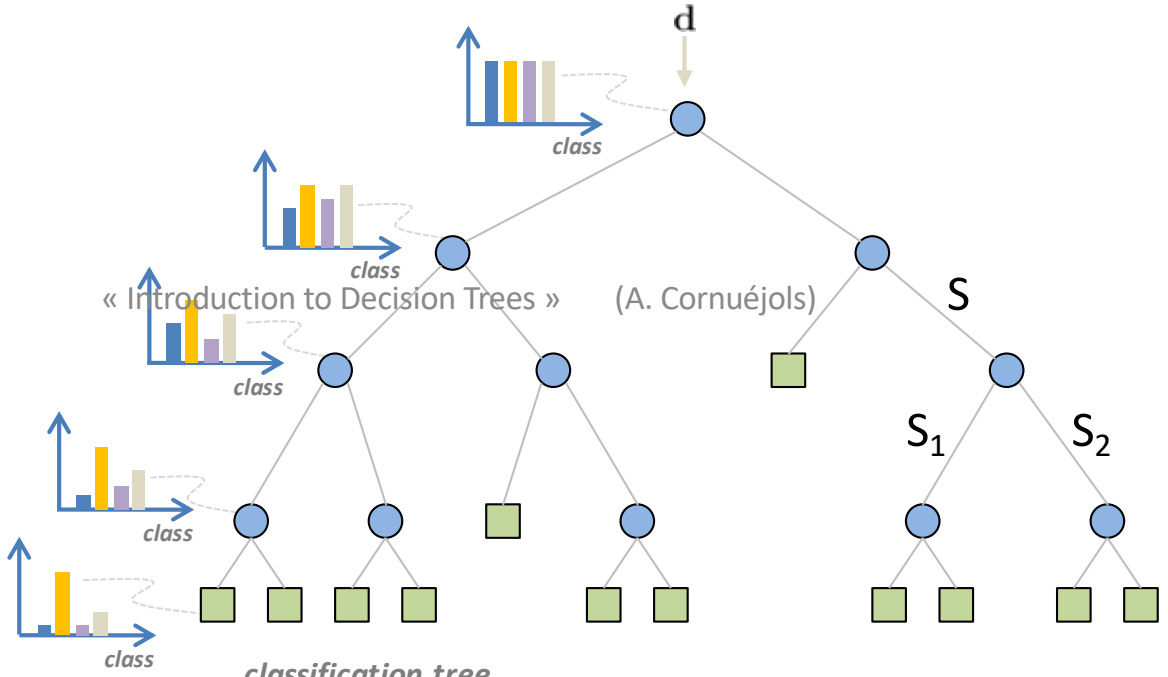
Tree outputs and objective functions

- Trees can be trained for
  - classification, regression, or clustering

- Change the object function
  - information gain for classification: $I = H(S) - \sum_{i=1}^{2} \frac{|S_i|}{|S|} H(S_i)$ measure of distribution purity



data



classification tree

$P_{l_t(\mathbf{d})}(Y(\mathbf{d}) = c)$

Impurity measure: the entropy criterion

Information entropy of S (with C classes) :

$$I(S) = -\sum_{i=1}^{C} p(c_i) \cdot \log\ p(c_i)$$

$p(c_i)$: probability of the class c_i

- Zero if only one class
- Increasing as the classes are more equi likely
- Equals $\log_2(k)$ when the k classes are equiprobables
- Unit: bit of information

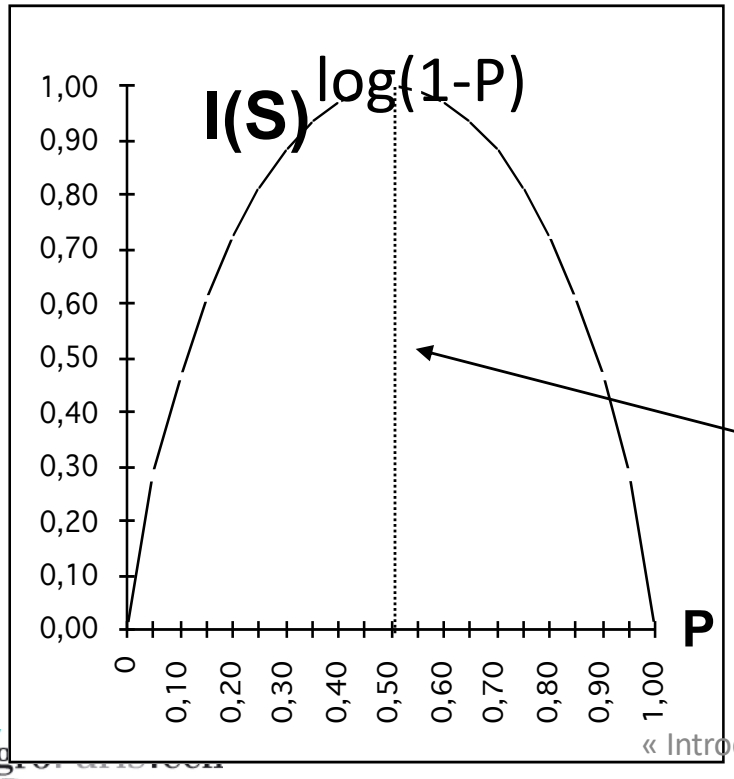
The entropy criterion for 2 classes

- For C=2 : $I(S) = -p_+ \times \log_2(p_+) - p_- \times \log_2(p_-)$

$p_+ = p/(p+n)$ and $p_- = n/(p+n)$

d'où $I(S) = -\frac{p}{(p+n)} \log \left(\frac{p}{(p+n)} \right) - \frac{n}{(p+n)} \log \left(\frac{n}{(p+n)} \right)$

and $I(S) = -P \log P - (1-P)$



P=p/(p+n)=n/(n+p)=0.5
equiprobable

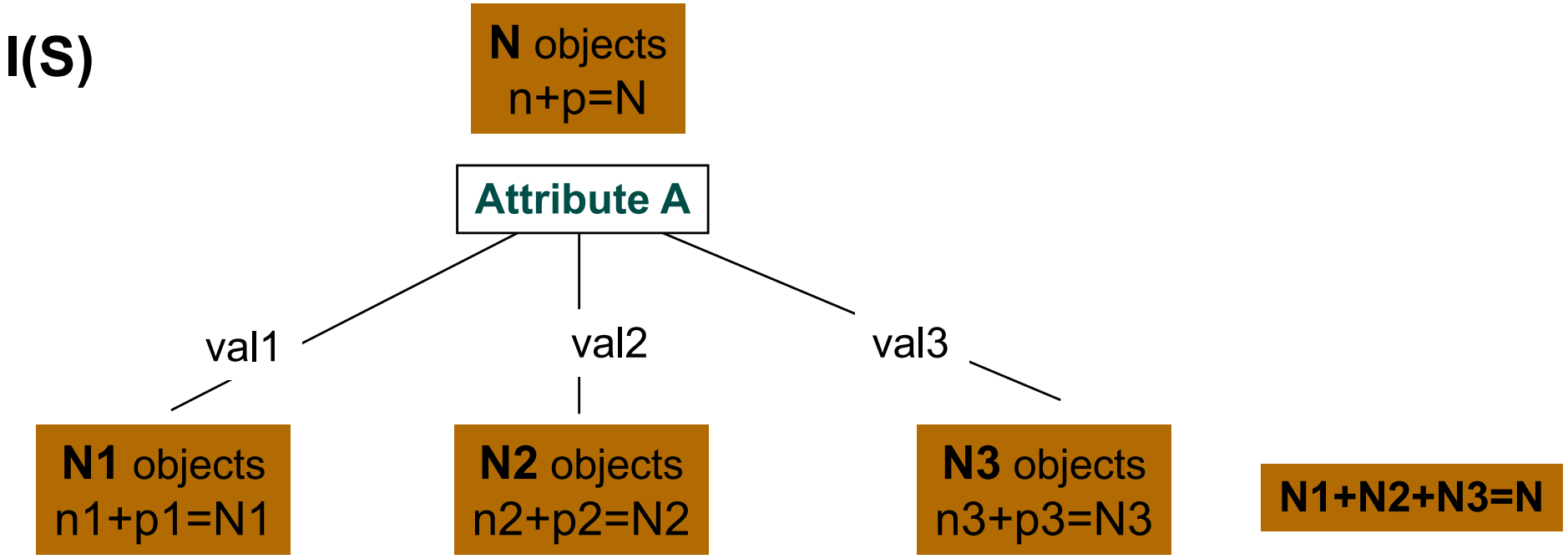
Entropy gain for one attribute

$$Gain(S,A) = I(S) - \sum_{v \in valeurs(A)} \frac{|S_v|}{|S|} \cdot I(S_v)$$

$|S_v|$: size of the sub-population in the branch *v* of *A*

Measures to what extent the knowledge of the value of attribute A brings information about the class of an example

Illustration



$$\mathbf{E(N,A)} = N1/N \times I(p1,n1) + N2/N \times I(p2,n2) + N3/N \times I(p3,n3)$$

The entropy gain for attribute A is:

GAIN(A)= I(S)-E(N,A)

Illustration

ID	color	root	sound	texture	umbilicus	surface	ripe
1	green	curly	muffled	clear	hollow	hard	true
2	dark	curly	dull	clear	hollow	hard	true
3	dark	curly	muffled	clear	hollow	hard	true
4	green	curly	dull	clear	hollow	hard	true
5	light	curly	muffled	clear	hollow	hard	true
6	green	slightly curly	muffled	clear	slightly hollow	soft	true
7	dark	slightly curly	muffled	slightly blurry	slightly hollow	soft	true
8	dark	slightly curly	muffled	clear	slightly hollow	hard	true
9	dark	slightly curly	dull	slightly blurry	slightly hollow	hard	false
10	green	straight	crisp	clear	flat	soft	false
11	light	straight	crisp	blurry	flat	hard	false
12	light	curly	muffled	blurry	flat	soft	false
13	green	slightly curly	muffled	slightly blurry	hollow	hard	false
14	light	slightly curly	dull	slightly blurry	hollow	hard	false
15	dark	slightly curly	muffled	clear	slightly hollow	soft	false
16	light	curly	muffled	blurry	flat	hard	false
17	green	curly	dull	slightly blurry	slightly hollow	hard	false

...

Illustration

ID	color	root	sound	texture	umbilicus	surface	ripe
1	green	curly	muffled	clear	hollow	hard	true
2	dark	curly	dull	clear	hollow	hard	true
3	dark	curly	muffled	clear	hollow	hard	true
4	green	curly	dull	clear	hollow	hard	true
5	light	curly	muffled	clear	hollow	hard	true
6	green	slightly curly	muffled	clear	slightly hollow	soft	true
7	dark	slightly curly	muffled	slightly blurry	slightly hollow	soft	true
8	dark	slightly curly	muffled	clear	slightly hollow	hard	true
9	dark	slightly curly	dull	slightly blurry	slightly hollow	hard	false
10	green	straight	crisp	clear	flat	soft	false
11	light	straight	crisp	blurry	flat	hard	false
12	light	curly	muffled	blurry	flat	soft	false
13	green	slightly curly	muffled	slightly blurry	hollow	hard	false
14	light	slightly curly	dull	slightly blurry	hollow	hard	false
15	dark	slightly curly	muffled	clear	slightly hollow	soft	false
16	light	curly	muffled	blurry	flat	hard	false
17	green	curly	dull	slightly blurry	slightly hollow	hard	false

$$\text{Ent}(D) = -\sum_{k=1}^{2} p_k \log_2 p_k = -\left(\frac{8}{17}\log_2\frac{8}{17} + \frac{9}{17}\log_2\frac{9}{17}\right) = 0.998$$

...

Illustration

ID	color	root	sound	texture	umbilicus	surface	ripe
1	green	curly	muffled	clear	hollow	hard	true
2	dark	curly	dull	clear	hollow	hard	true
3	dark	curly	muffled	clear	hollow	hard	true
4	green	curly	dull	clear	hollow	hard	true
5	light	curly	muffled	clear	hollow	hard	true
6	green	slightly curly	muffled	clear	slightly hollow	soft	true
7	dark	slightly curly	muffled	slightly blurry	slightly hollow	soft	true
8	dark	slightly curly	muffled	clear	slightly hollow	hard	true
9	dark	slightly curly	dull	slightly blurry	slightly hollow	hard	false
10	green	straight	crisp	clear	flat	soft	false
11	light	straight	crisp	blurry	flat	hard	false
12	light	curly	muffled	blurry	flat	soft	false
13	green	slightly curly	muffled	slightly blurry	hollow	hard	false
14	light	slightly curly	dull	slightly blurry	hollow	hard	false
15	dark	slightly curly	muffled	clear	slightly hollow	soft	false
16	light	curly	muffled	blurry	flat	hard	false
17	green	curly	dull	slightly blurry	slightly hollow	hard	false

For the "color" attribute

:

3 subsets D^1 (color = green)

D^2 (color = dark)

D^3 (color = light)

$$\text{Ent}(D) = -\sum_{k=1}^{2} p_k \log_2 p_k = -\left(\frac{8}{17}\log_2\frac{8}{17} + \frac{9}{17}\log_2\frac{9}{17}\right) = 0.998$$

$$\text{Ent}(D^1) = -\left(\frac{3}{6}\log_2\frac{3}{6} + \frac{3}{6}\log_2\frac{3}{6}\right) = 1.000$$

$$\text{Ent}(D^2) = -\left(\frac{4}{6}\log_2\frac{4}{6} + \frac{2}{6}\log_2\frac{2}{6}\right) = 0.918$$

$$\text{Ent}(D^3) = -\left(\frac{1}{5}\log_2\frac{1}{5} + \frac{4}{5}\log_2\frac{4}{5}\right) = 0.722$$

$$\begin{aligned}\text{Gain}(D, \text{color}) &= \text{Ent}(D) - \sum_{v=1}^{3}\frac{|D^v|}{|D|}\text{Ent}(D^v)\\ &= 0.998 - \left(\frac{6}{17}\times 1.000 + \frac{6}{17}\times 0.918 + \frac{5}{17}\times 0.722\right)\\ &= 0.109.\end{aligned}$$

Illustration

ID	color	root	sound	texture	umbilicus	surface	ripe
1	green	curly	muffled	clear	hollow	hard	true
2	dark	curly	dull	clear	hollow	hard	true
3	dark	curly	muffled	clear	hollow	hard	true
4	green	curly	dull	clear	hollow	hard	true
5	light	curly	muffled	clear	hollow	hard	true
6	green	slightly curly	muffled	clear	slightly hollow	soft	true
7	dark	slightly curly	muffled	slightly blurry	slightly hollow	soft	true
8	dark	slightly curly	muffled	clear	slightly hollow	hard	true
9	dark	slightly curly	dull	slightly blurry	slightly hollow	hard	false
10	green	straight	crisp	clear	flat	soft	false
11	light	straight	crisp	blurry	flat	hard	false
12	light	curly	muffled	blurry	flat	soft	false
13	green	slightly curly	muffled	slightly blurry	hollow	hard	false
14	light	slightly curly	dull	slightly blurry	hollow	hard	false
15	dark	slightly curly	muffled	clear	slightly hollow	soft	false
16	light	curly	muffled	blurry	flat	hard	false
17	green	curly	dull	slightly blurry	slightly hollow	hard	false

Information gain for the **other attributes**:

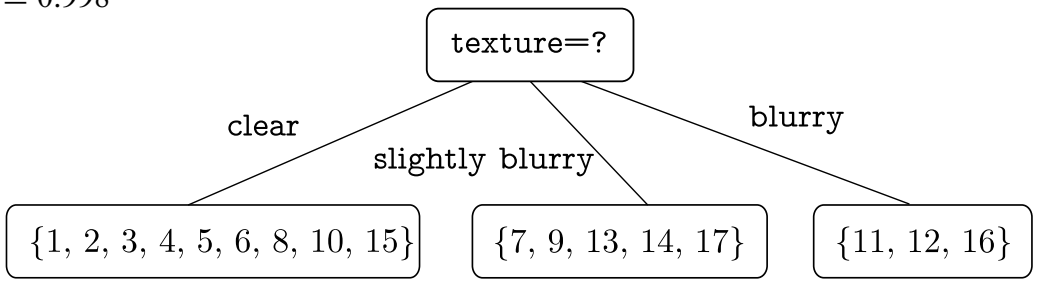
$\text{Gain}(D, \text{root}) = 0.143;$ $\quad \text{Gain}(D, \text{sound}) = 0.141;$

$\text{Gain}(D, \text{texture}) = 0.381;$ $\quad \text{Gain}(D, \text{umbilicus}) = 0.289;$

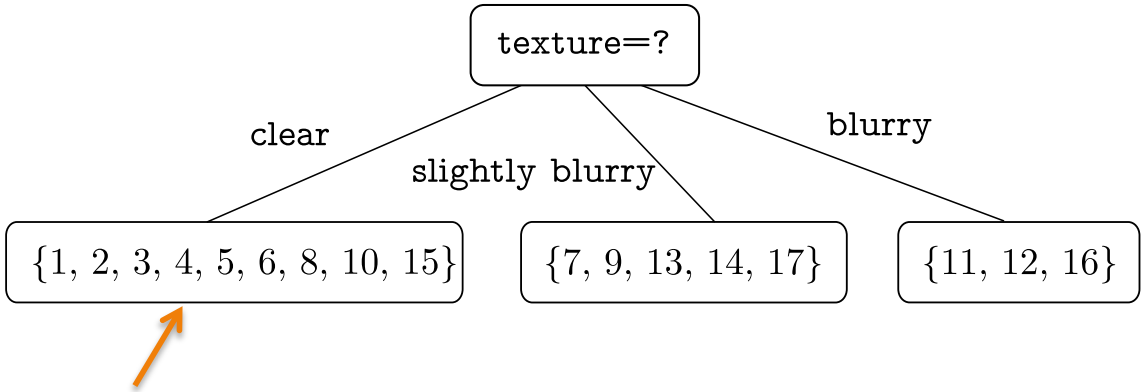
$\text{Gain}(D, \text{surface}) = 0.006.$

Texture is the best attribute

$$\text{Ent}(D) = -\sum_{k=1}^{2} p_k \log_2 p_k = -\left(\frac{8}{17}\log_2\frac{8}{17} + \frac{9}{17}\log_2\frac{9}{17}\right) = 0.998$$



Illustration



$$D^1 = D^{\texttt{texture=clear}} = \{(1,+),(2,+),(3,+),(4,+),(5,+),(6,+),(8,-),(10,-),(15,-)\}$$

$\text{Gain}(D^1, \text{color}) = 0.043;$ $\quad \text{Gain}(D^1, \text{root}) = 0.458;$

$\text{Gain}(D^1, \text{sound}) = 0.331;$ $\quad \text{Gain}(D^1, \text{umbilicus}) = 0.458;$

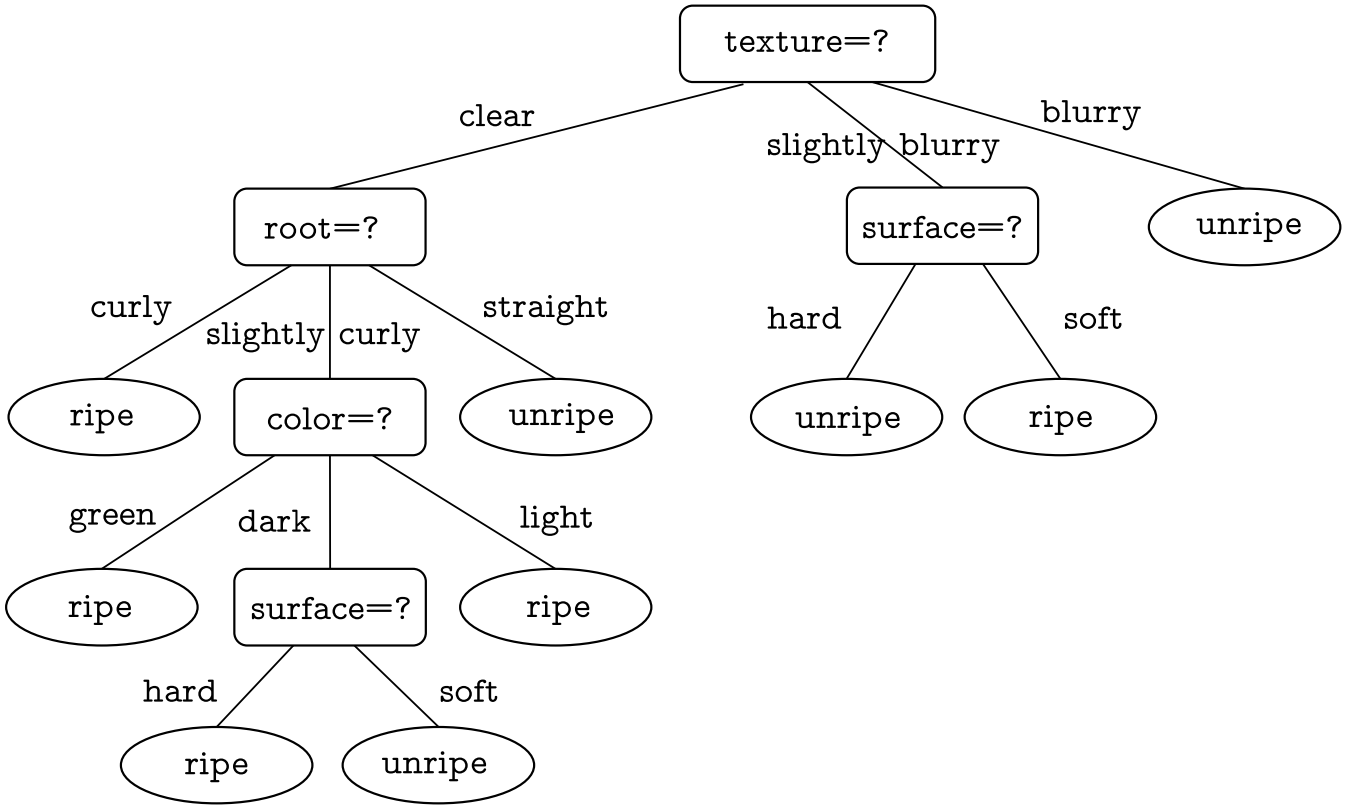
$\text{Gain}(D^1, \text{surface}) = 0.458.$

Root, surface and umbilicus are the best attributes

Any one of them can be chosen

...

Illustration



A possible final decision tree for the database

...

Impurity measure: the Gini criterion

- Ideally:
  - The measure should be zero if the sub-populations are homogeneous (only one class)
  - The measure should be maximal if the classes are maximally mixed in the sub-populations

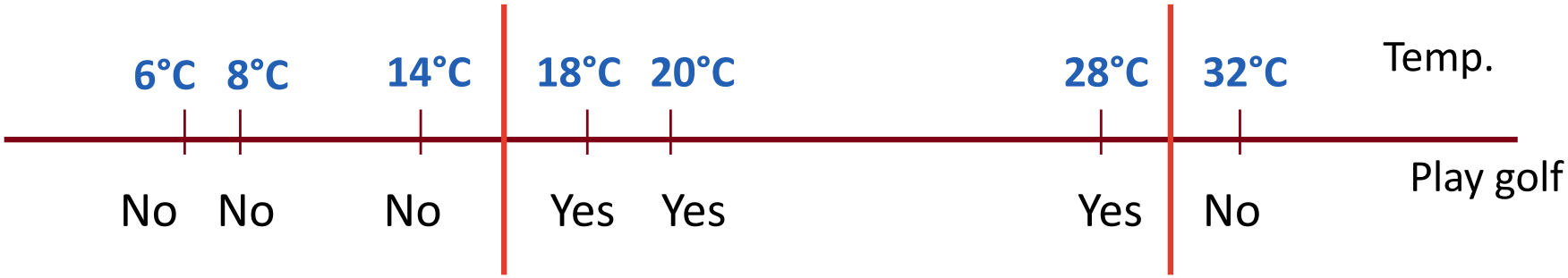
- Index Gini [Breiman et al.,84]

$$Gini(D) = 1 - \sum_{j=1}^{k} (p_j)^2$$

Some problems

And their solutions

Discretizing continuous attributes



Here, two candidate thrersholds: 16°C and 30°C

The attribute $Temp_{>16°C}$ is the most informative, it is chosen

Different branching factors

- Problem:

  Entropy gain unduly favors the attributes with high branching factors

- Two solutions:
  - *Binarize all attributes*
    - *But the resulting tree lose interpretability*
  - *Introduce a normalizing factor to correct the bias*

$$Gain\_norm(S,A) = \frac{Gain(S,A)}{\sum_{i=1}^{nb\ valeurs\ de\ A} \frac{|S_i|}{|S|} \cdot \log \frac{|S_i|}{|S|}}$$

Missing values

- Given example $\langle x , c(x) \rangle$ with missing values for attribute A
- How can we compute $gain(S,A)$?

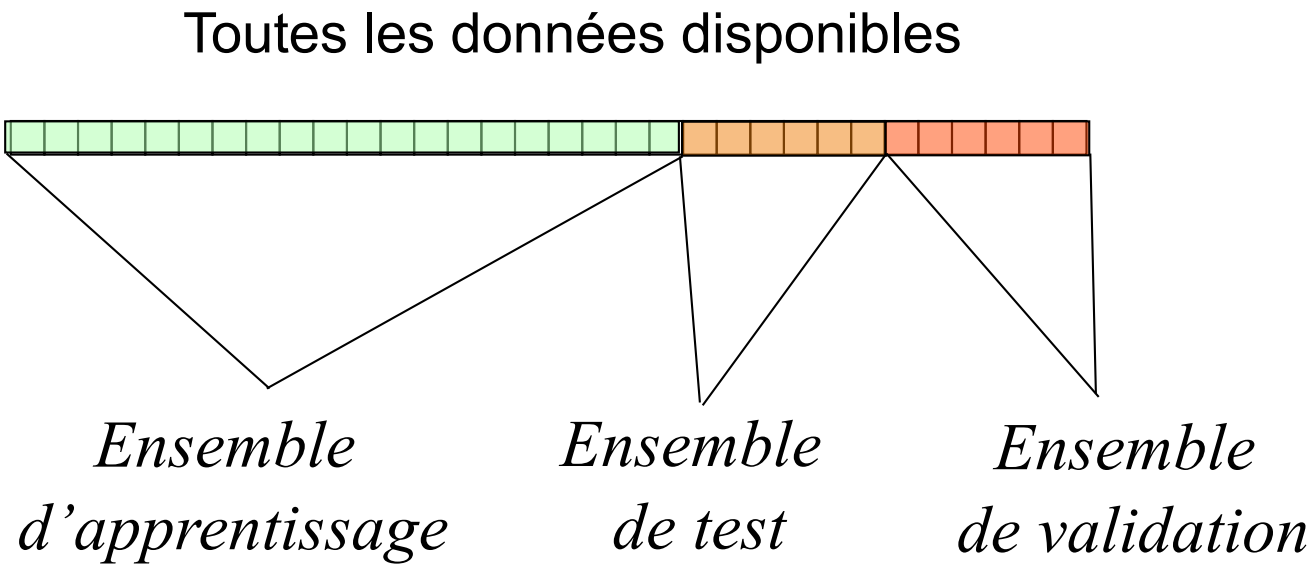
1. Take the **most frequent value** for A **in S**
2. Take the **most frequent value** for A in **the node**
3. Distribute the example into **fictive example**s with the possible values of A weighted by their respective frequencies
   - E.g. if 6 examples in this node take the value A=a_1 and 4 examples the value A=a_2
     A(x) = a_1 with prob=0.6 and A(x) = a_2 with prob=0.4
   - **When predicting**, give the label corresponding to the most likely leave

Le problème de la généralisation

A-t-on appris un bon arbre de décision ?

- Ensemble d'**apprentissage**. Ensemble **test**.
- **Courbe** d'apprentissage
- **Méthodes** d'évaluation de la généralisation
  - Sur un **ensemble test**
  - Validation **croisée**

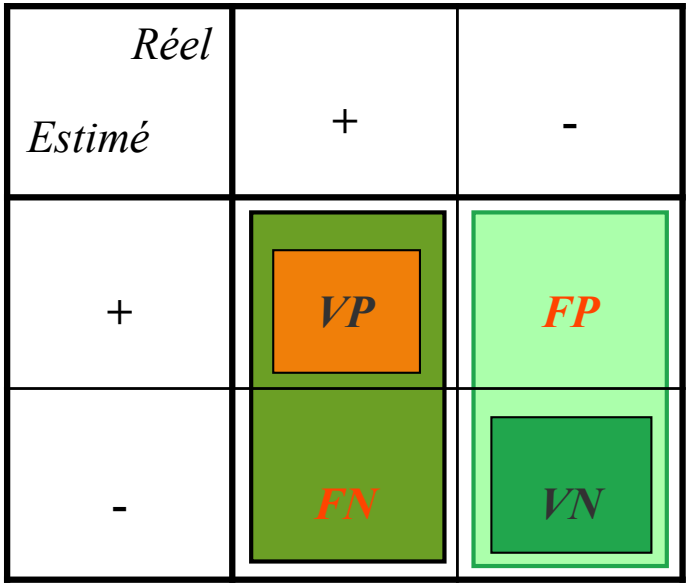
Ensembles de données (collections)



Indicateurs de performances

- **Sensibilité** $\frac{VP}{FN + VP}$
- **Spécificité** $\frac{VN}{VN + FP}$

- *Rappel* $\frac{VP}{VP + FN}$
- *Précision* $\frac{VP}{VP + FP}$



Indicateurs de performances

- **FN-rate** $\frac{FN}{VP + FN}$ ■ *FP-rate* $\frac{FP}{FP + VN}$

- **F-measure** $\frac{2 \times rappel \times précision}{rappel + précision} = \frac{2\,VP}{2\,VP + FP + FN}$

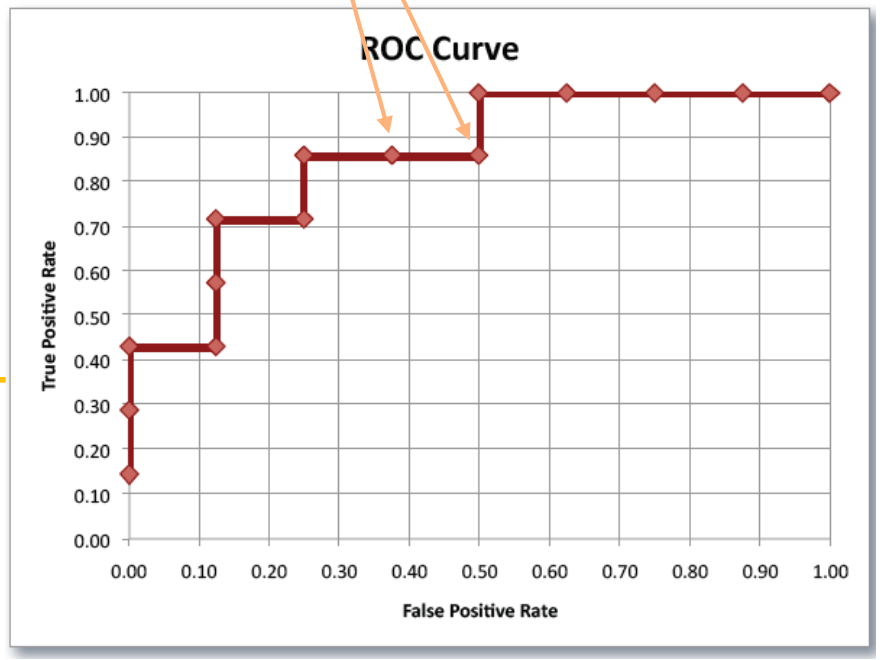
Réel / Estimé	+	-
+	VP	FP
-	FN	VN

Courbe ROC

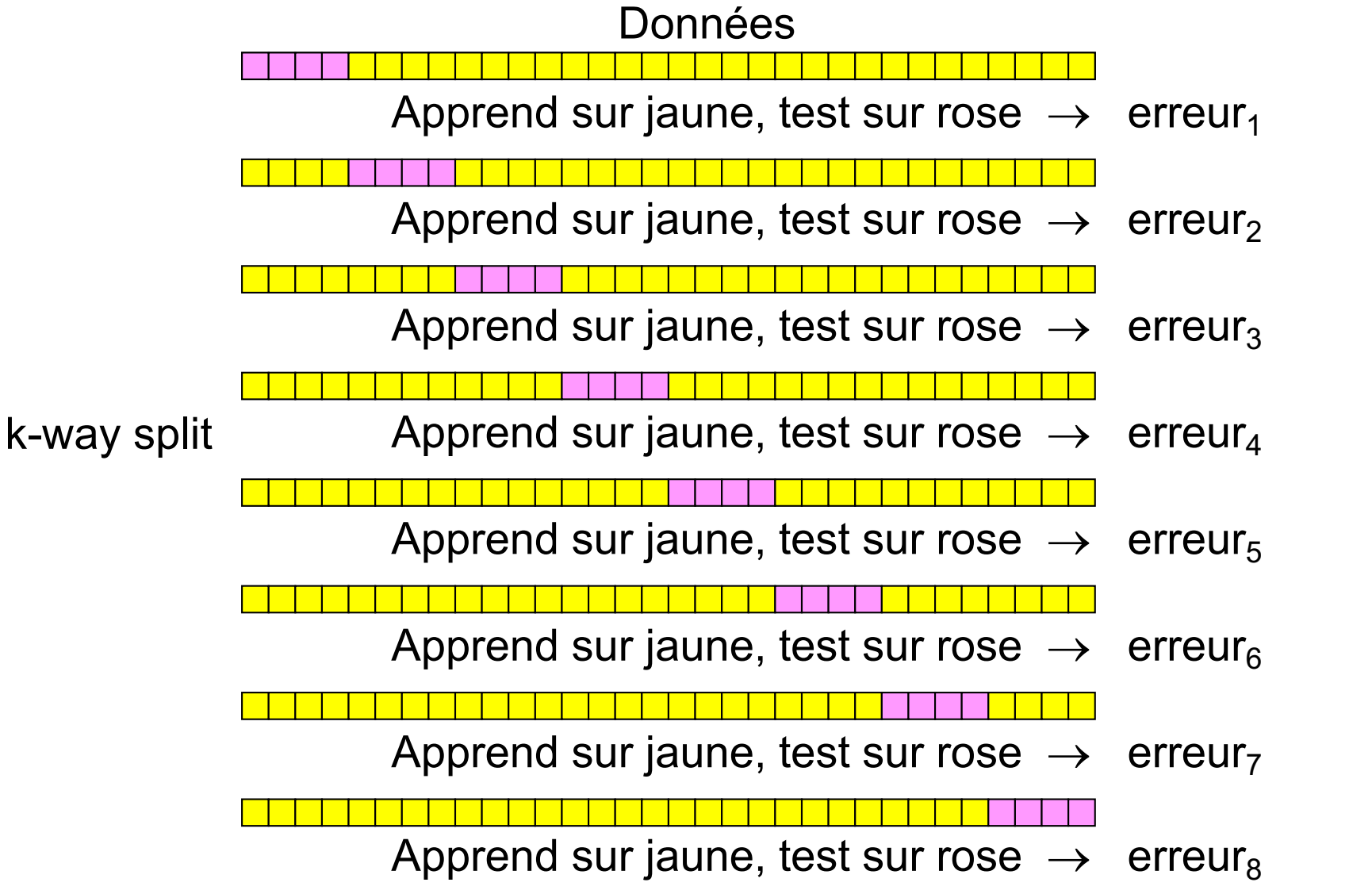
On fait évoluer T de 0 à 1

Score	T=0.5	Label	FP	TP	FPR	TPR
0.99	1	1	0	1	0.00	0.14
0.9	1	1	0	2	0.00	0.29
0.8	1	1	0	3	0.00	0.43
0.85	**1**	0	1	3	0.13	0.43
0.7	1	1	1	4	0.13	0.57
0.7	1	1	1	5	0.13	0.71
0.65	**1**	0	2	5	0.25	0.71
0.6	1	1	2	6	0.25	0.86
0.45	0	0	3	6	0.38	0.86
0.45	0	0	4	6	0.50	0.86
0.4	**0**	1	4	7	0.50	1.00
0.3	0	0	5	7	0.63	1.00
0.2	0	0	6	7	0.75	1.00
0.2	0	0	7	7	0.88	1.00
0.2	0	0	8	7	1.00	1.00

T = 0.5



Validation croisée à *k* plis (*k*-fold)



Matrice de confusion

14% des papillons sont pris pour des poissons

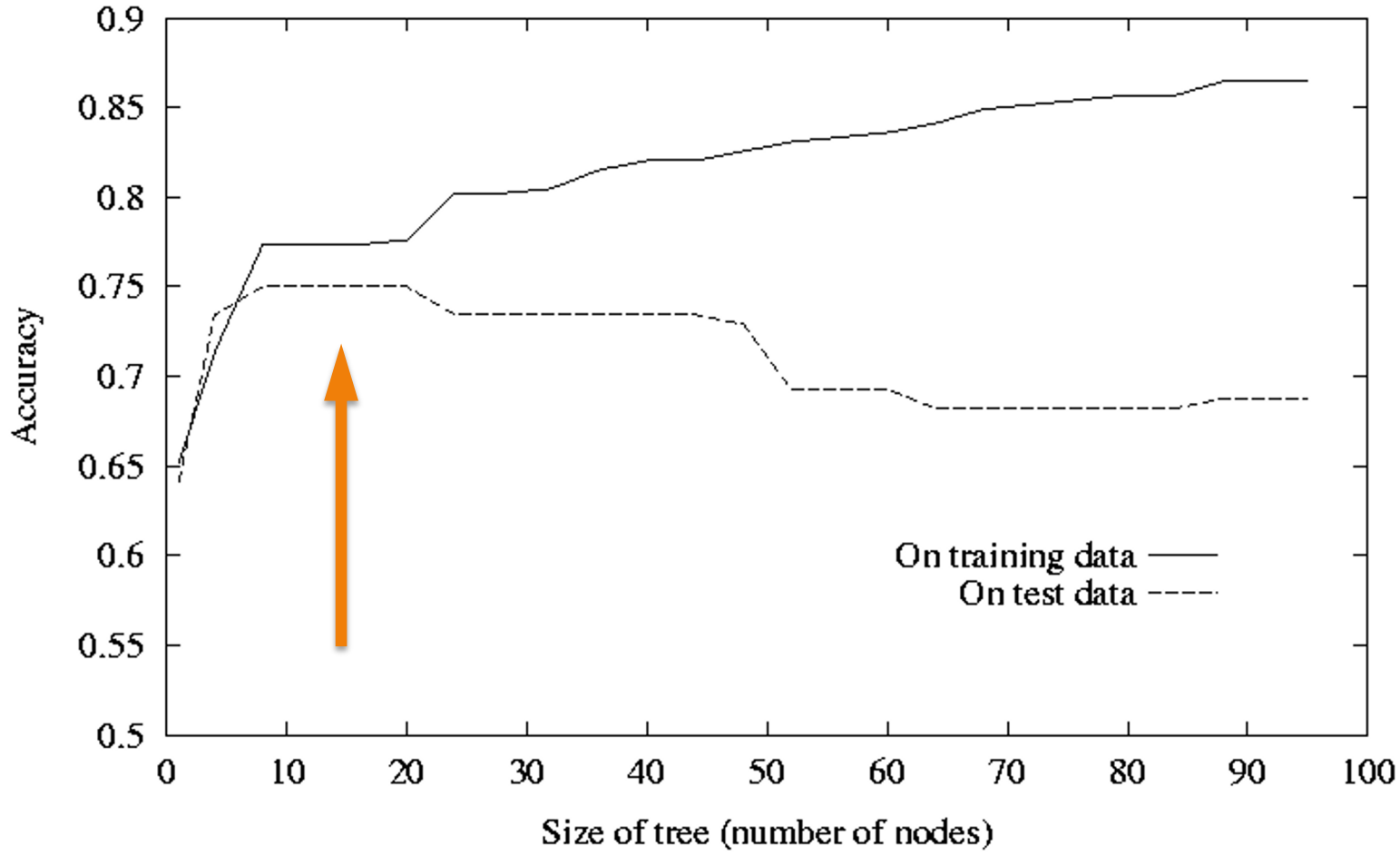
	Av	Pl	Ut	Mi	Ch	Po	Ve	Pa	Por	Fi	Vo	Fl
Avi	**67%**	2%	-	-	2%	2%	10%	10%	4%	2%	-	-
Pla	-	**21%**	-	2%	7%	19%	10%	12%	5%	-	19%	5%
Uta	17%	-	**33%**	-	7%	-	-	3%	10%	10%	13%	7%
Min	-	-	-	**100%**	-	-	-	-	-	-	-	-
Chi	26%	5%	7%	-	**14%**	9%	12%	9%	5%	2%	12%	-
Poi	5%	13%	3%	8%	-	**13%**	18%	21%	-	3%	10%	8%
Ver	2%	2%	-	-	10%	7%	**43%**	-	21%	5%	7%	2%
Pap	6%	6%	-	-	2%	14%	14%	**35%**	6%	-	12%	4%
Por	2%	2%	-	-	-	2%	-	12%	**70%**	10%	-	2%
Fig	-	-	-	-	-	-	6%	-	24%	**70%**	-	-
Voi	21%	6%	-	-	4%	4%	8%	4%	4%	29%	**19%**	-
Fle	2%	9%	-	-	-	9%	21%	14%	-	-	16%	**28%**

Sur-apprentissage

- Types de bruits
  - Erreurs de **description**
  - Erreurs de **classification**
  - “**clashes**”
  - valeurs **manquantes**

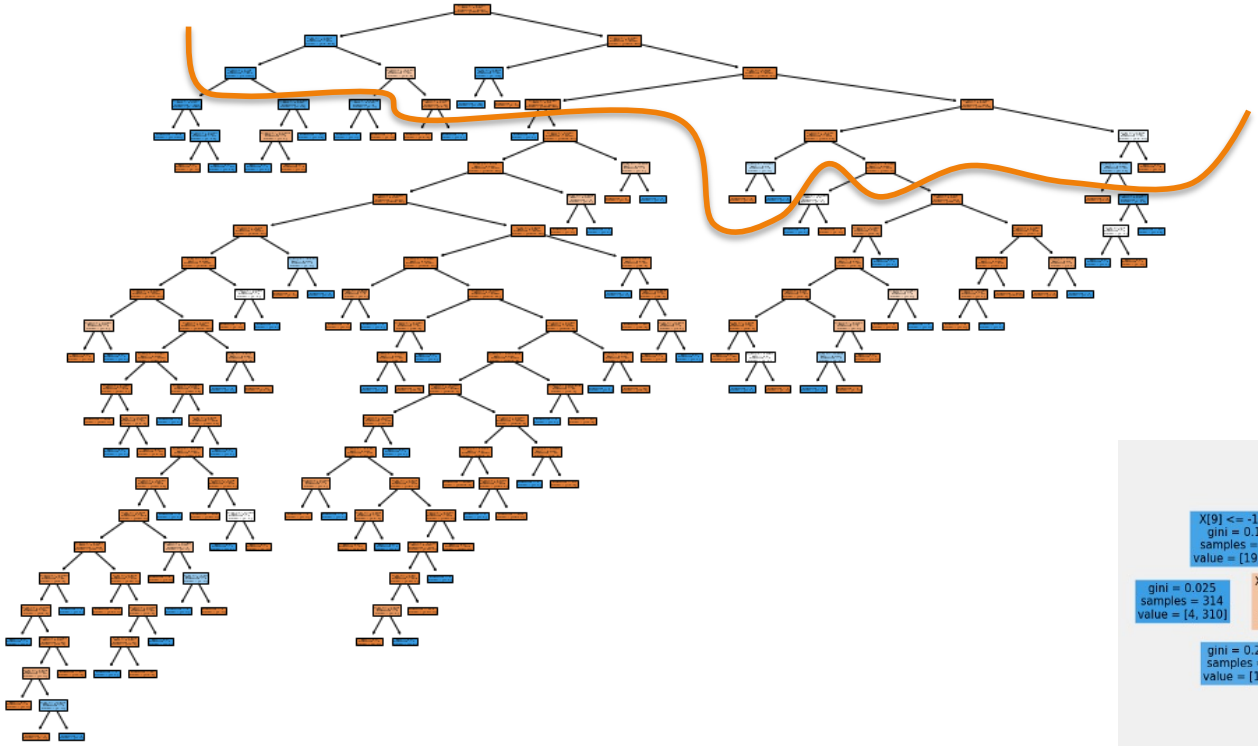
- Effet
  - **Arbre trop développé** : « touffus », trop profond

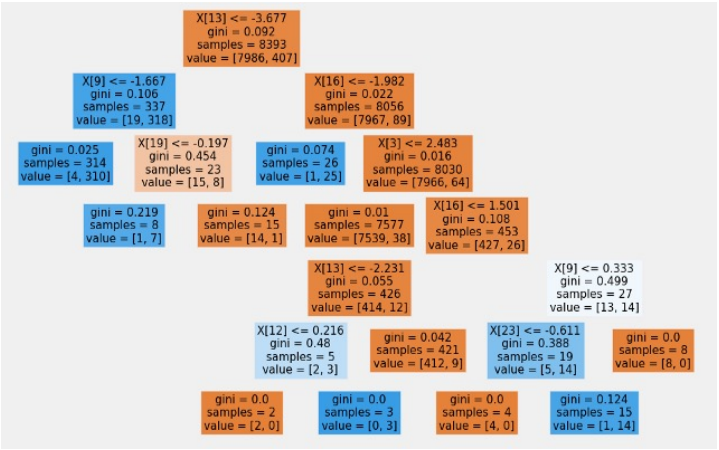
Overfitting



©Tom Mitchell, McGraw Hill, 1997

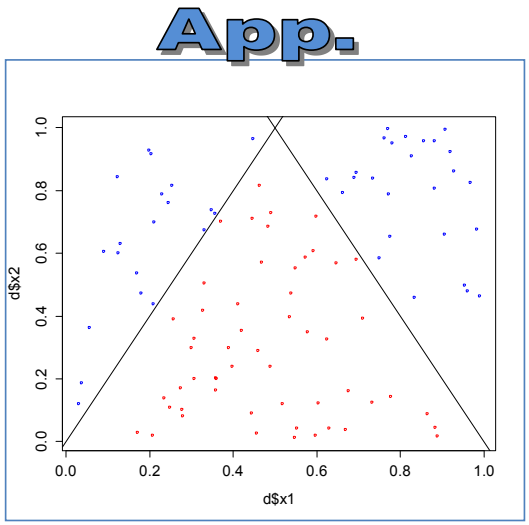
Overfitting in decision trees

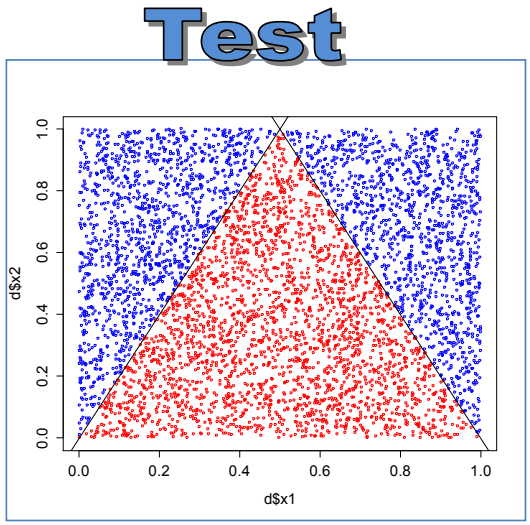




...

Arbre de décision

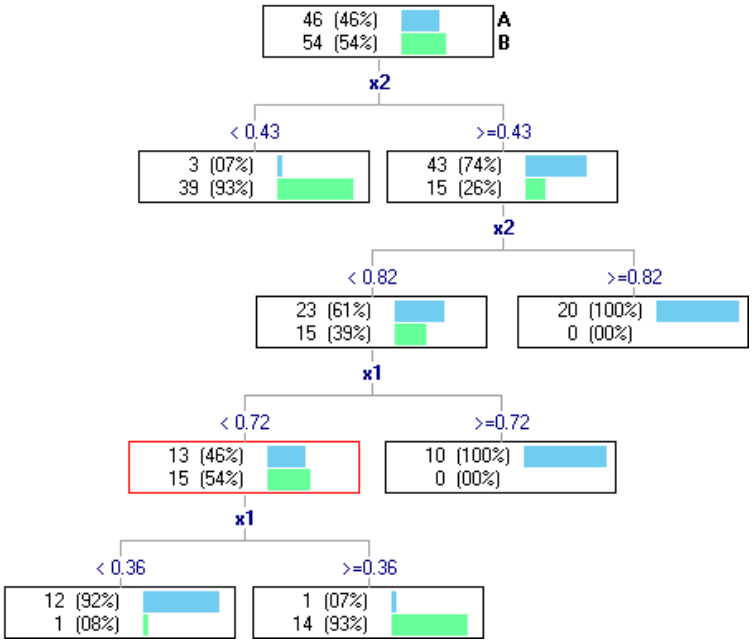


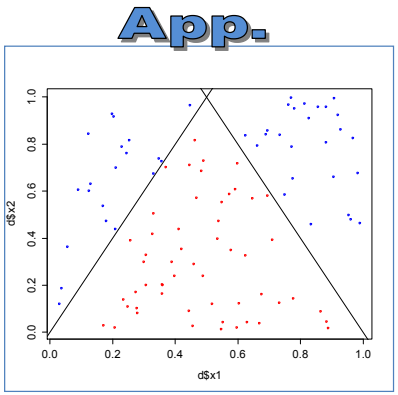


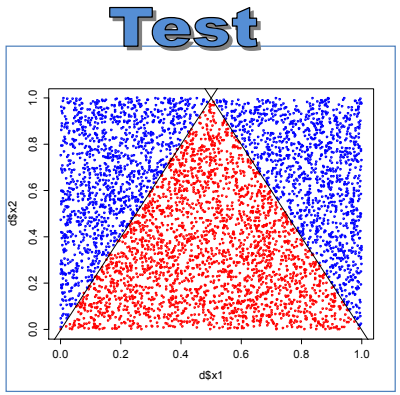
Y = « bleu »
si (1) x2 > 2 * x1 pour x1 < 0.5
ou (2) x2 > (2 – 2 * x1) pour x1 ≥ 0.5
« rouge » autrement

- toto

Arbre de décision







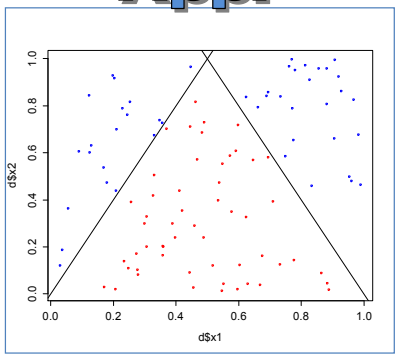
Y = « bleu »
si (1) x2 > 2 * x1 pour x1 < 0.5
ou (2) x2 > (2 – 2 * x1) pour x1 ≥ 0.5
« rouge » autrement

Arbre profond : biais faible, variance forte

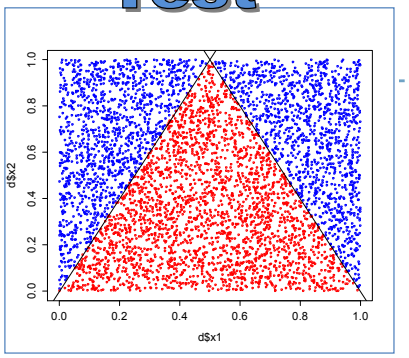
Arbre court : biais fort, variance faible

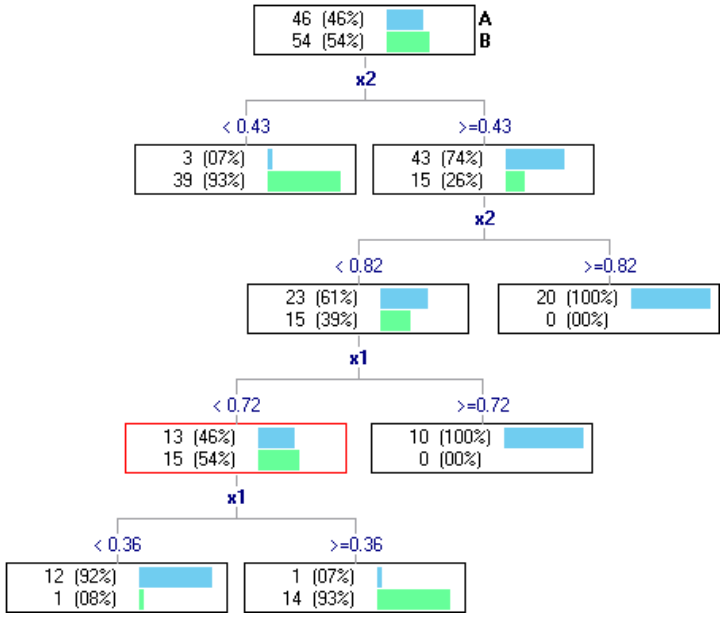
- toto

App.



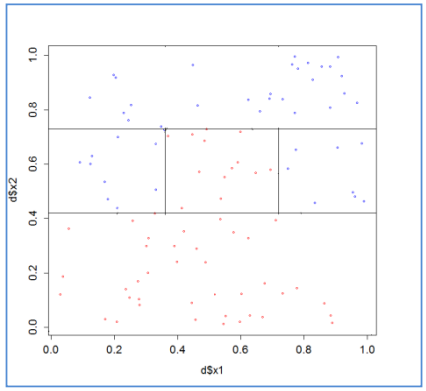
Test





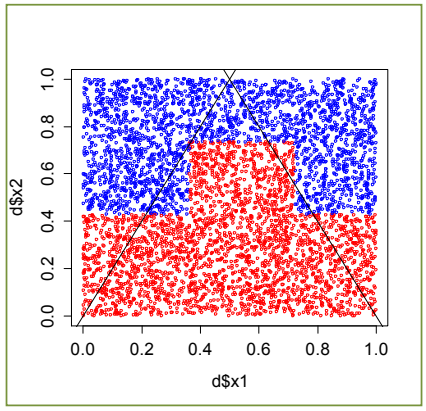
Arbre profond : biais faible, variance forte

Arbre court : biais fort, variance faible



TRAIN

(5 feuilles dans l'arbre = 5 zones sont définies)



TEST

$\varepsilon = 0.1632$

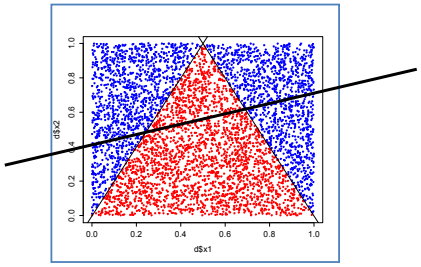
- toto

Composantes de l'erreur

L'erreur résulte de deux composantes

Biais

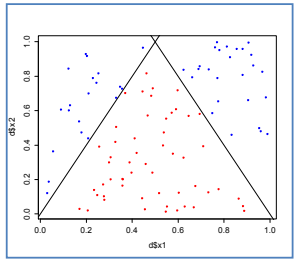
Traduit l'incapacité du modèle à traduire le concept (la « vraie » fonction) reliant Y aux X.



Un classifieur linéaire ne peut pas fonctionner ici. Impossible de trouver une droite permettant de séparer les points bleus des rouges.

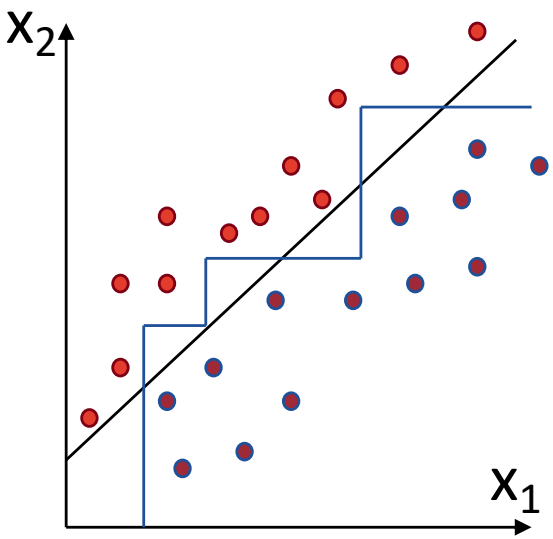
Variance

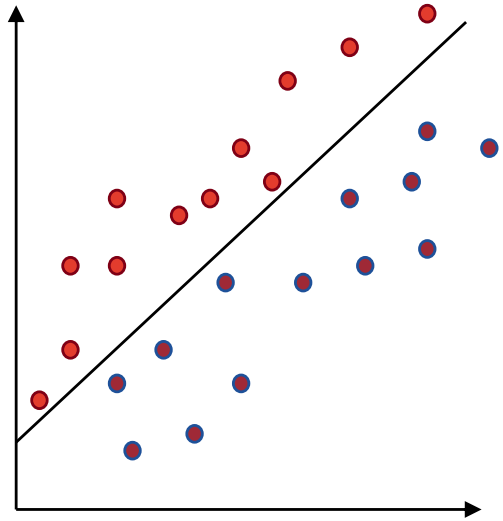
Sensibilité (variabilité par rapport) aux fluctuations d'échantillonnage.

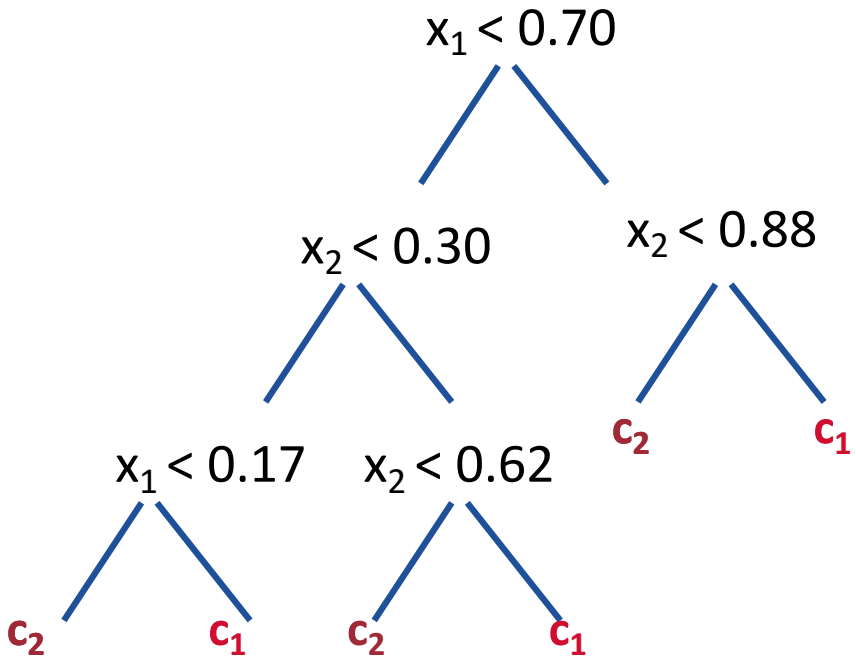


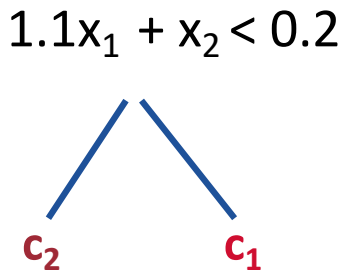
Le faible effectif de l'échantillon d'apprentissage ne permet pas de trouver avec exactitude les « bonnes » frontières.

Oblique trees



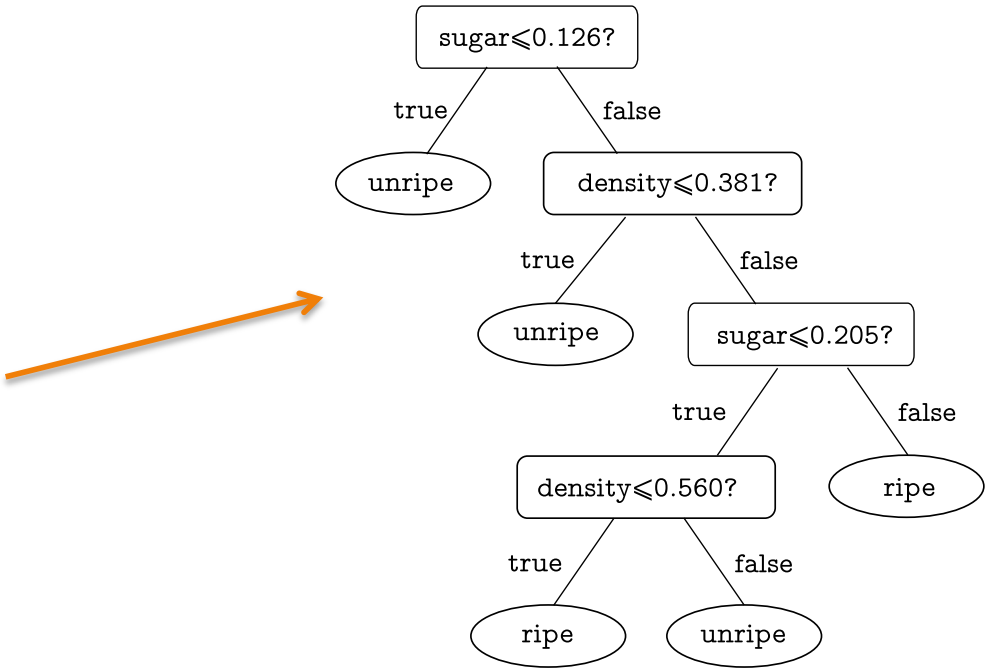






Oblique decision trees

ID	density	sugar	ripe
1	0.697	0.460	true
2	0.774	0.376	true
3	0.634	0.264	true
4	0.608	0.318	true
5	0.556	0.215	true
6	0.403	0.237	true
7	0.481	0.149	true
8	0.437	0.211	true
9	0.666	0.091	false
10	0.243	0.267	false
11	0.245	0.057	false
12	0.343	0.099	false
13	0.639	0.161	false
14	0.657	0.198	false
15	0.360	0.370	false
16	0.593	0.042	false
17	0.719	0.103	false

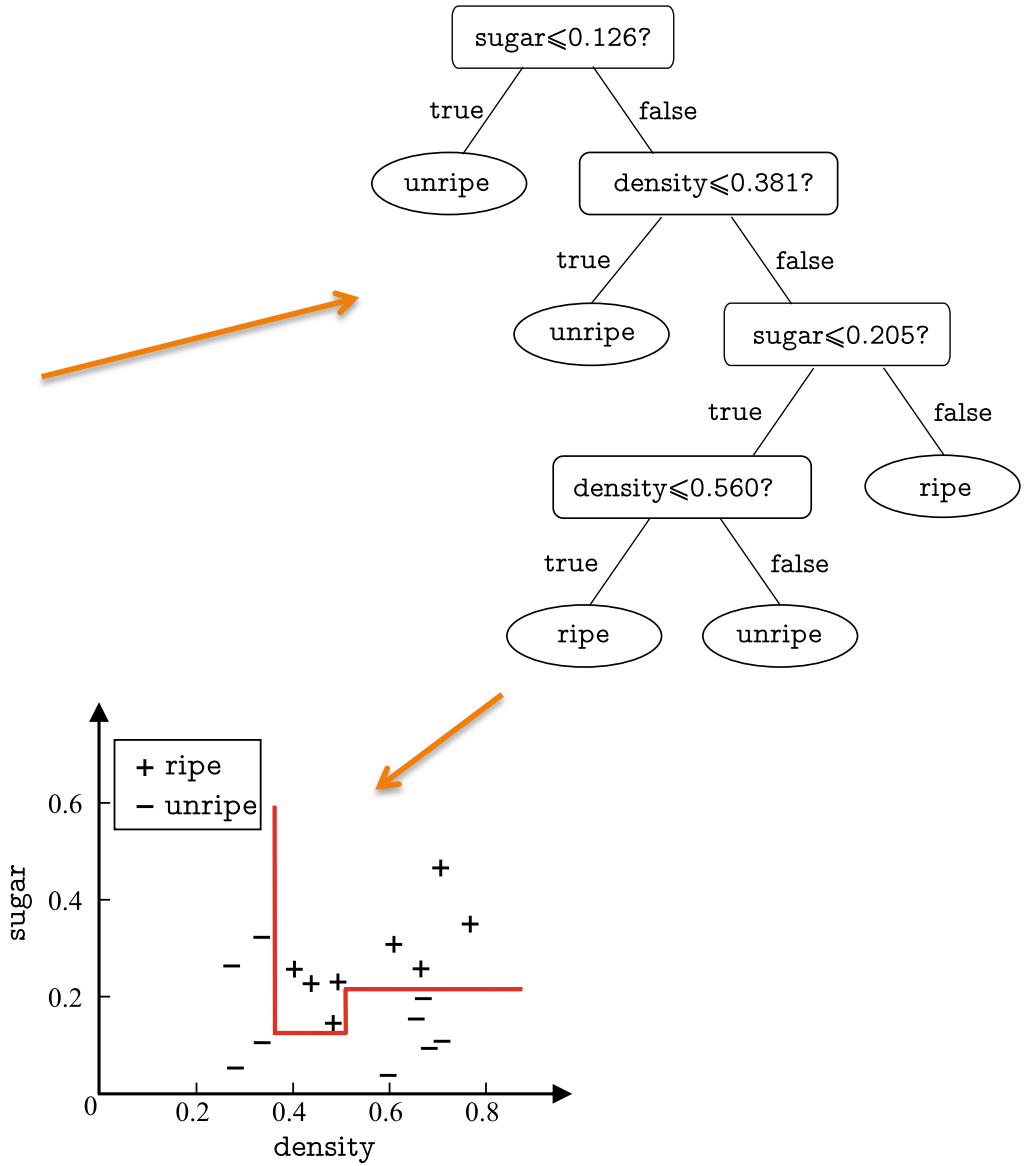


...

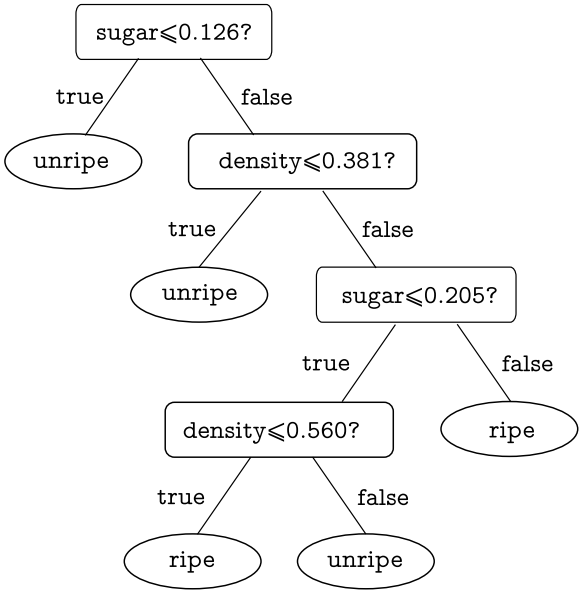
Oblique decision trees

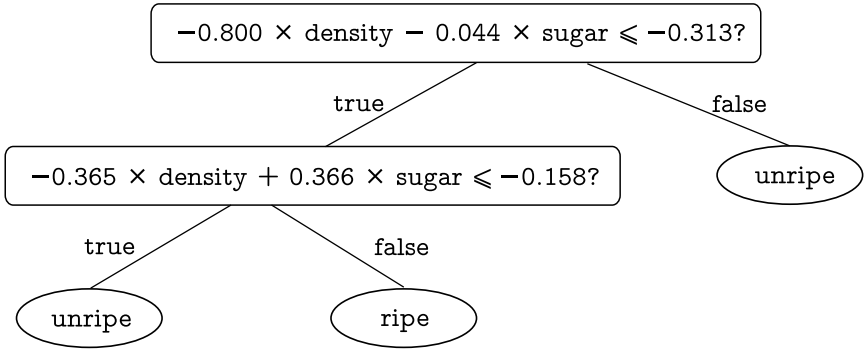
ID	density	sugar	ripe
1	0.697	0.460	true
2	0.774	0.376	true
3	0.634	0.264	true
4	0.608	0.318	true
5	0.556	0.215	true
6	0.403	0.237	true
7	0.481	0.149	true
8	0.437	0.211	true
9	0.666	0.091	false
10	0.243	0.267	false
11	0.245	0.057	false
12	0.343	0.099	false
13	0.639	0.161	false
14	0.657	0.198	false
15	0.360	0.370	false
16	0.593	0.042	false
17	0.719	0.103	false

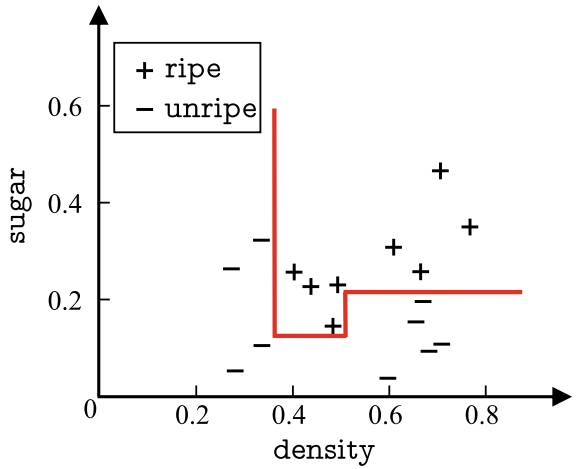
...

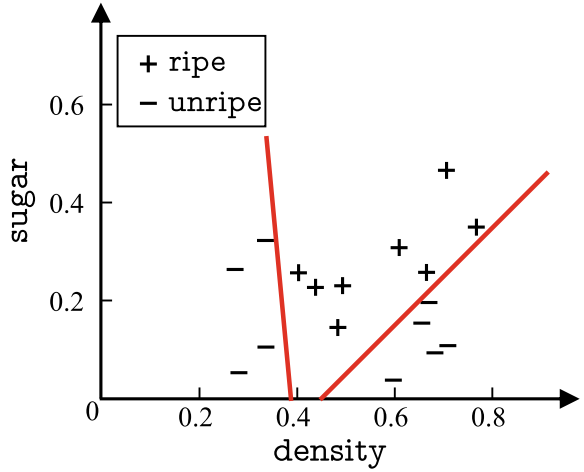


Oblique decision trees



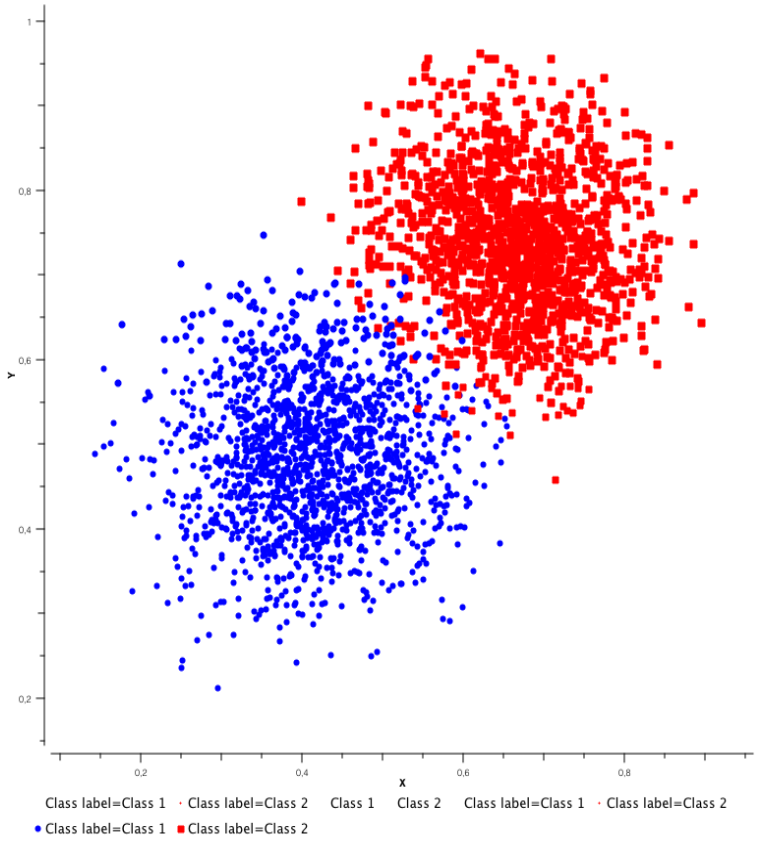


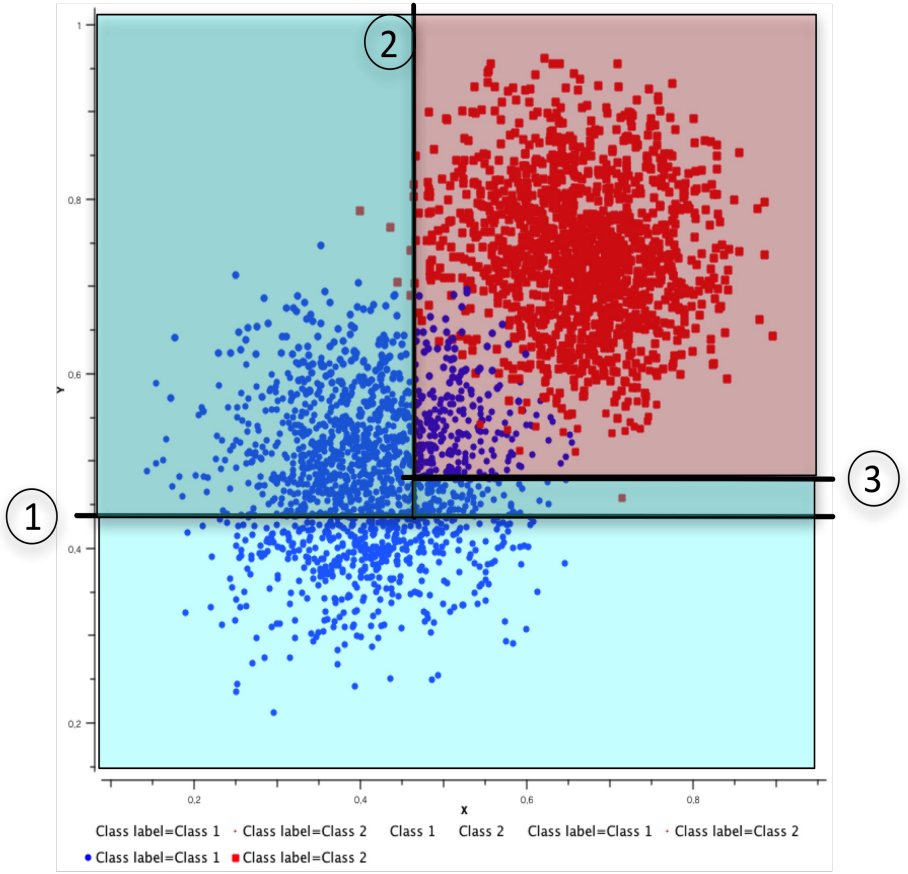




...

Oblique trees





Overfitting in decision trees

Pre and post pruning

Bilan sur les arbres de décision

1. Avantages
   - Interprétables
     - Sélection automatique de variables « pertinentes »
     - Les branches des arbres peuvent se lire comme des règles
   - Non paramétrique
     - Traitement indifférencié des différents types de variables prédictives
     - Robuste face aux données aberrantes
     - Solutions pour traiter les données manquantes
   - Complexité calculatoire faible

2. Inconvénients
   - Problèmes de stabilité sur les petites bases de données (feuilles à très petits effectifs)
   - Méthode gloutonne et myope (pb pour identifier des interactions entre variables (e.g. le XOR)

Arbres de **régression**

Limites des méthodes classiques de régression

- *Y* comme fonction linéaire d'une variable à valeur réelle

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

- Régression multiple : Y fonction linéaire d'un ensemble de variables indépendantes $\mathbf{Y} = \beta_0 + \beta_1 \mathbf{X} + \varepsilon$

- Régression non linéaire $\mathbf{Y} = \beta_0 + \beta_1 \mathbf{X} + \beta_2 \mathbf{X}\mathbf{X}^\top + \varepsilon$

Immensité de l'espace de recherche si on cherche à prendre en compte toutes les combinaisons des attributs

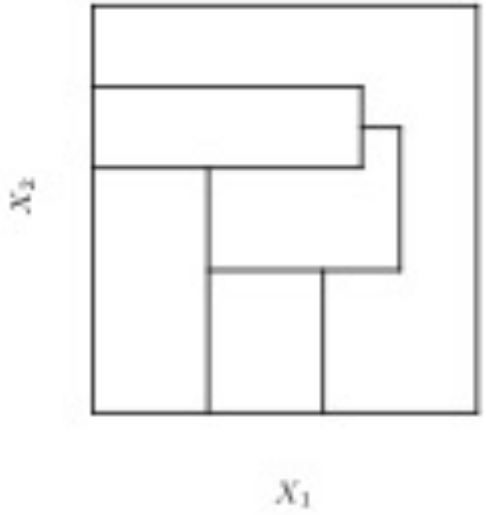
Régression linéaire vs. arbres de régression

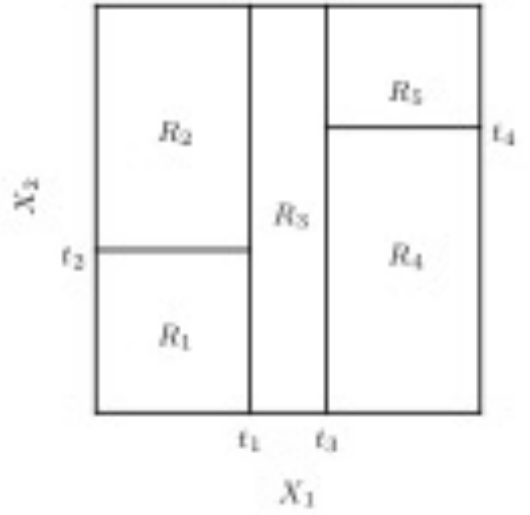
- **Modèle global** défini sur l'ensemble de l'espace de description

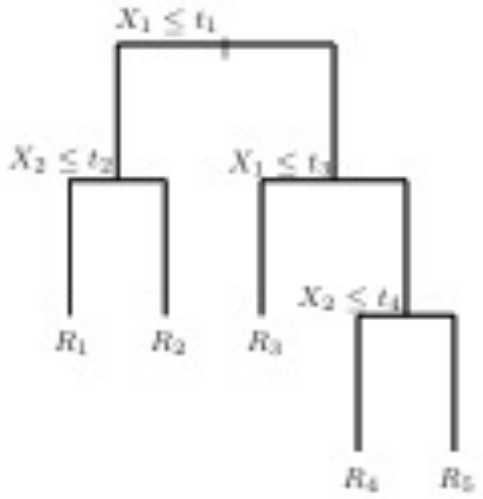
- Partition de l'espace avec des **modèles locaux**

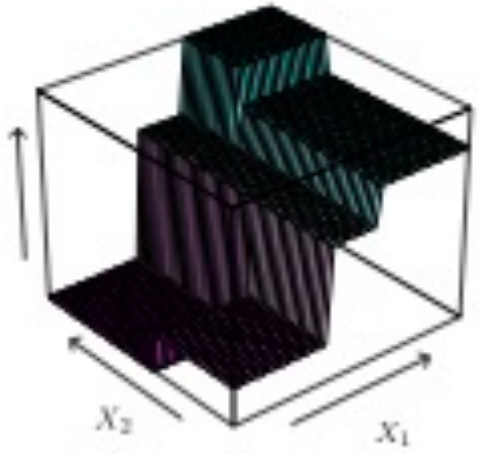
Arbres de décision : quels concepts ?

Domaine continu

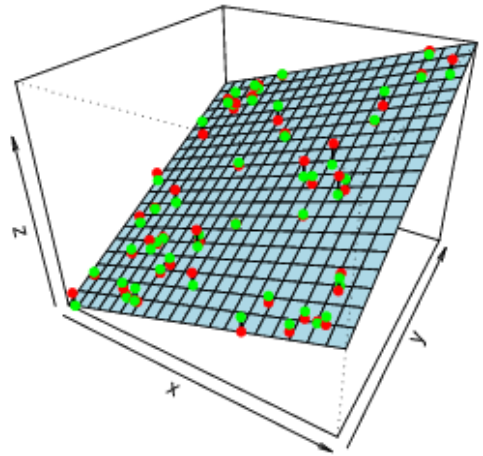


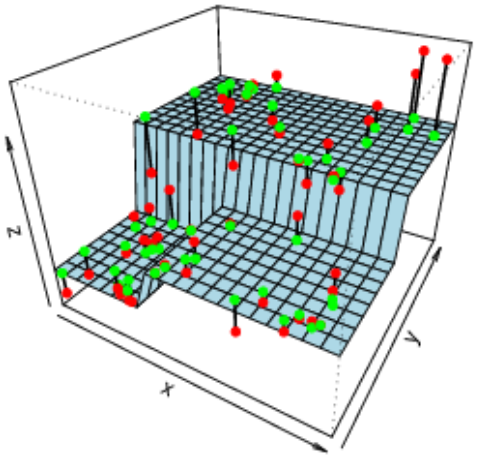


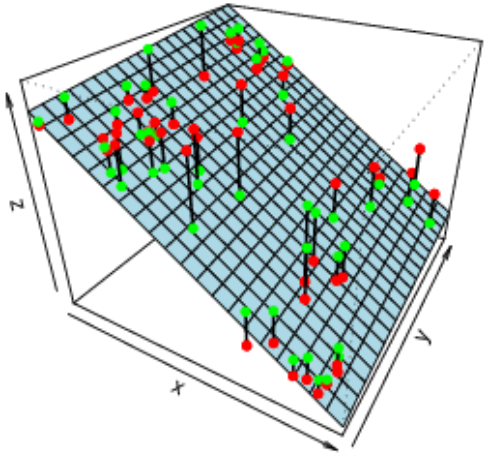


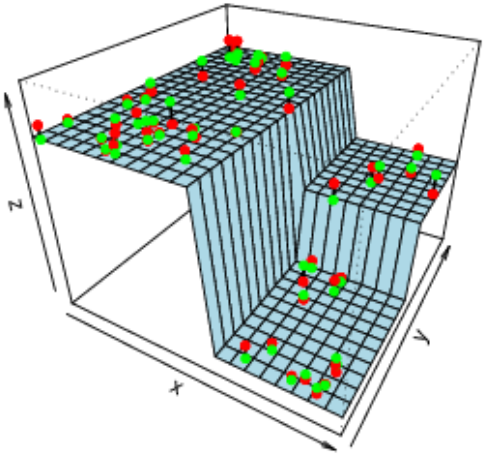


Arbre de régression vs. régression linéaire









Régression linéaire

Arbre de régression

Particularités des **arbres de régression**

- Les attributs et la classe sont à **valeur continue**
- On associe à chaque région R_i de X une valeur constante c_i.
- On cherche en général à minimiser l'*erreur quadratique* :

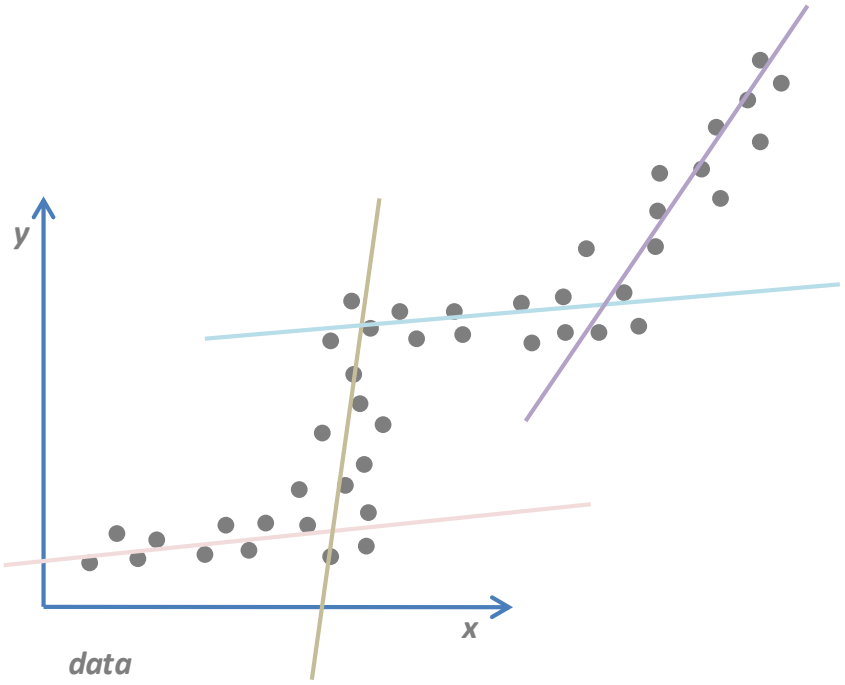
$$MSE = \sum_{i=1}^{m} \sum_{k=1}^{K} \mathcal{I}(R_k)(y_i - c_k)^2$$

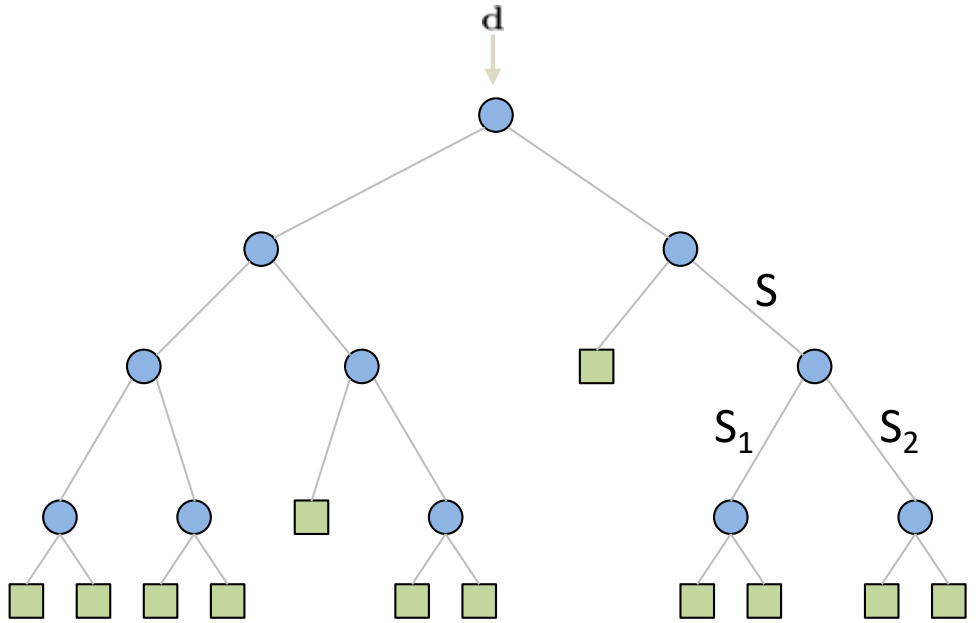
Induction des arbres de régression

- **Choix de l'attribut** et du **point de division** minimisant la somme des écarts quadratique à la moyenne dans chacune des régions de l'espace créées

- **Arrêt** lorsque
  - Plus assez de points par région
  - Différence des moyennes entre régions sous un seuil fixé

Regression trees (model trees)





regression tree

« Introduction to Decision Trees » (A. Cornuéjols)

- Real-valued output *y*
- Object function: maximize $Err(S) - \sum_{i=1}^{2} \frac{|S_i|}{|S|} Err(S_i)$

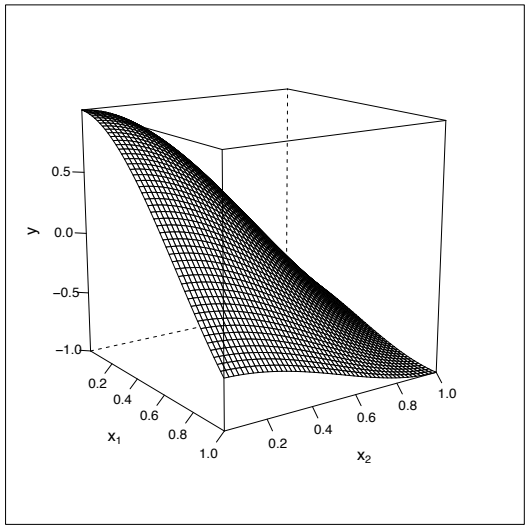
measure of fit of model

$$Err(S) = \sum_{j \in S} \left(y_j - y(x_j)\right)^2$$

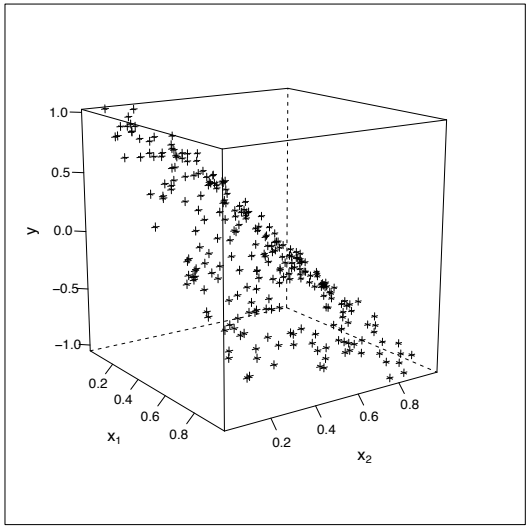
e.g. linear model y = ax+b,
Or just constant model

Arbres de régression : exemple

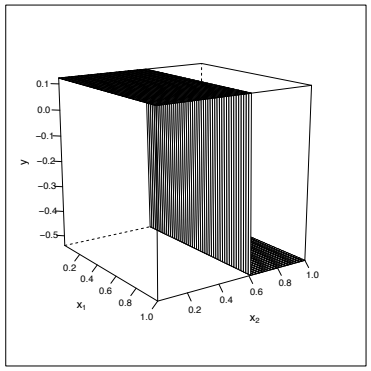
Function surface



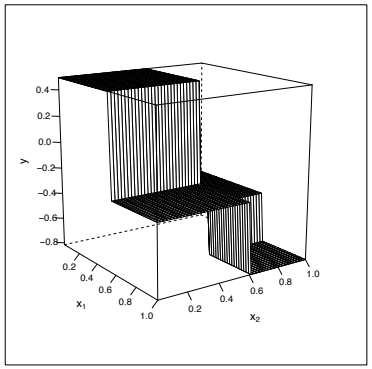
Sample points



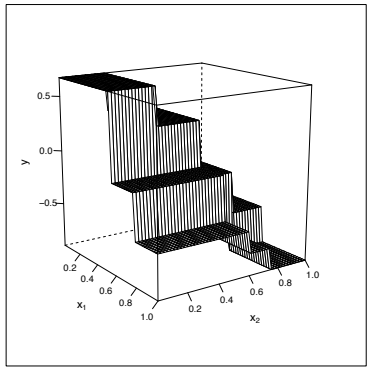
Regression tree surface, maxdepth = 1



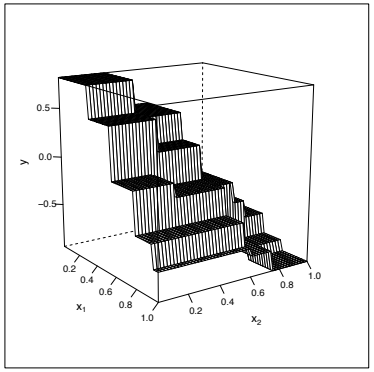
Regression tree surface, maxdepth = 2



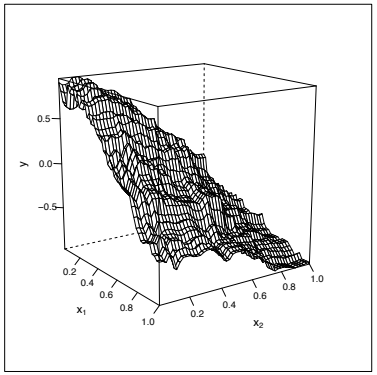
Regression tree surface, maxdepth = 3



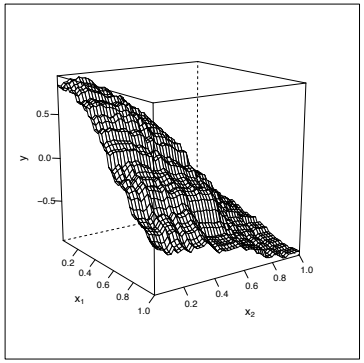
Regression tree surface, maxdepth = 4



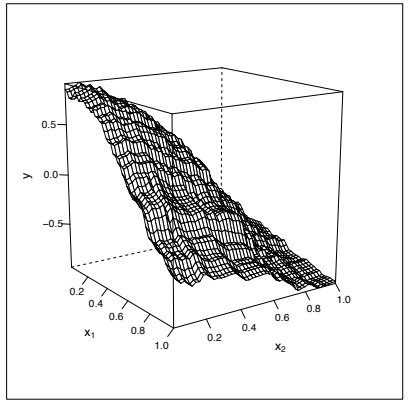
Random forest surface, B = 10



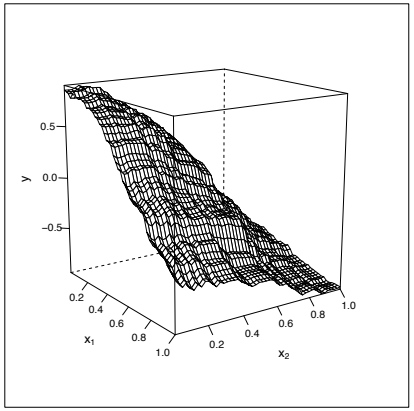
Random forest surface, B = 50



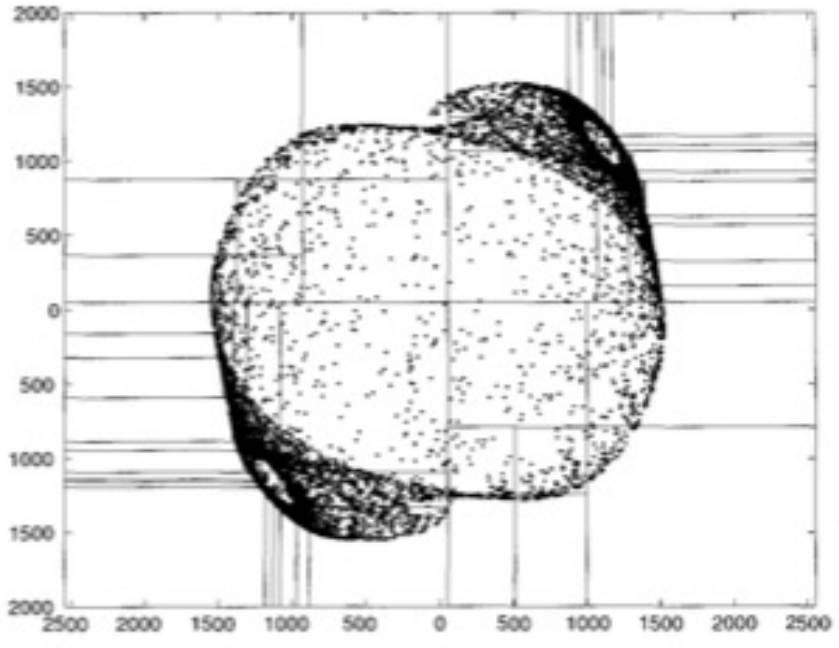
Random forest surface, B = 100

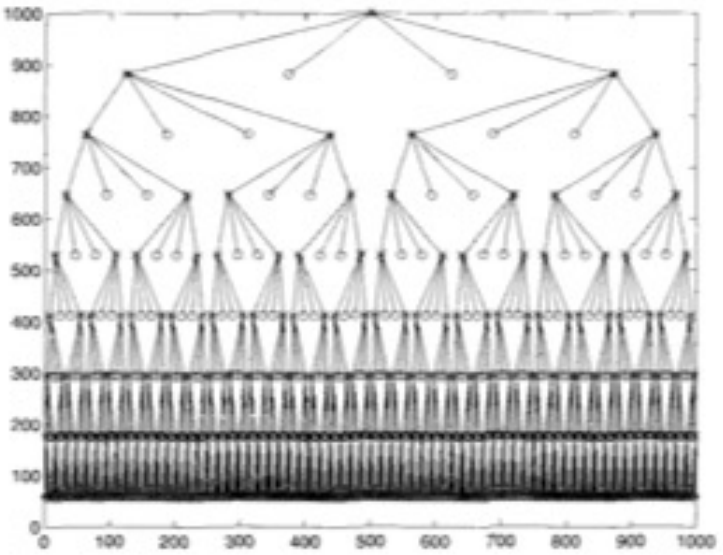


Random forest surface, B = 500



Arbres de régression : exemple





Tiré de [Anne-Emmanuelle BADEL*, Olivier MICHEL* et Alfred HERO, « Arbres de régression modélisation non paramétrique et analyse des séries temporelles », 1996]

Quand utiliser des arbres de régression

- La régression classique ne marche pas
  - Dimension de l'espace d'entrée élevée
- L'interprétabilité du modèle est importante
- Le problème se prête bien à une **division selon les axes**

Bagging = Bootstrapping aggregation

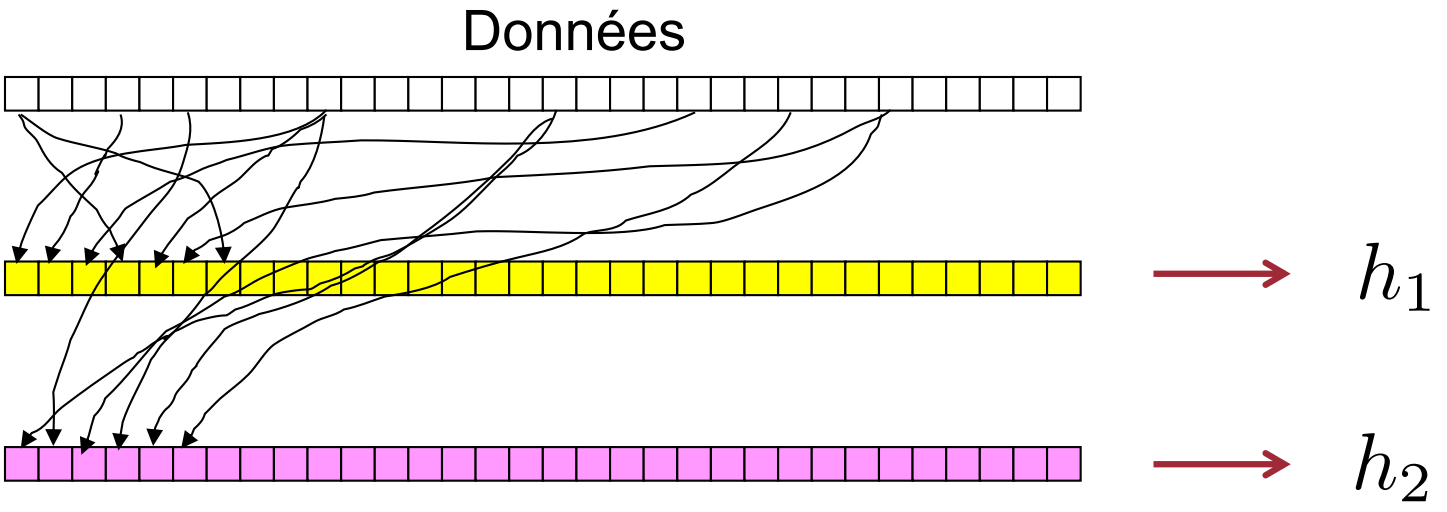
Bagging [Breiman,96]

- **Génération de *k* échantillons « indépendants »** par tirage avec remise dans l'échantillon S_m

- **Pour chaque échantillon**, apprentissage d'un classifieur en utilisant le même algorithme d'apprentissage

- La **prédiction finale** pour un nouvel exemple est obtenue par vote (simple) des classifieurs

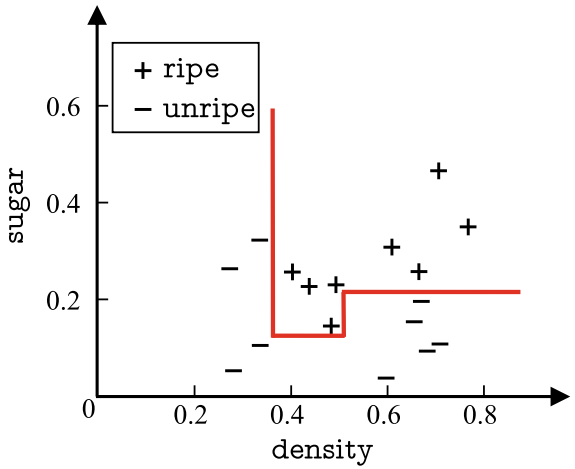
Le Bagging



$$H(\mathbf{x}) = \text{sign}\left[\sum_{t=1}^{T} h_t(\mathbf{x})\right]$$

Autre illustration du bagging sur jeu de données

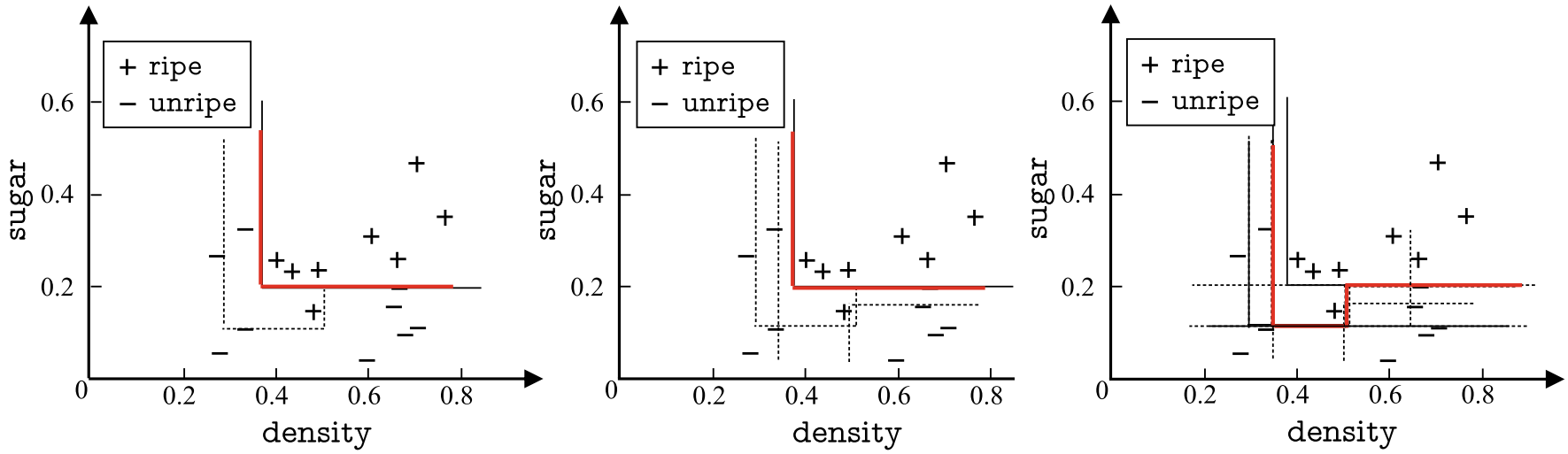
ID	density	sugar	ripe
1	0.697	0.460	true
2	0.774	0.376	true
3	0.634	0.264	true
4	0.608	0.318	true
5	0.556	0.215	true
6	0.403	0.237	true
7	0.481	0.149	true
8	0.437	0.211	true
9	0.666	0.091	false
10	0.243	0.267	false
11	0.245	0.057	false
12	0.343	0.099	false
13	0.639	0.161	false
14	0.657	0.198	false
15	0.360	0.370	false
16	0.593	0.042	false
17	0.719	0.103	false



Apprentissage par

arbre de décisions

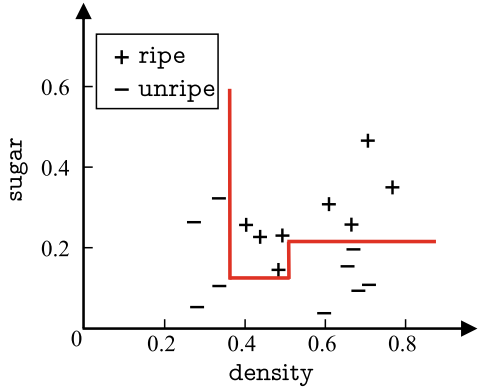
Autre illustration du bagging sur jeu de données



(a) 3 base learners. (b) 5 base learners. (c) 11 base learners.

Here "**base learner**" = decision stump

Arbre de décisions →



From [Zhi-Hua ZHOU « Machine Learning ». Springer, 2021]

Bagging (suite)

- Il est souvent dit que :
  - Le bagging fonctionne en réduisant la variance
    en laissant le biais inchangé

- Mais, encore incomplètement compris
  - Voir [Yves Grandvalet : « Bagging equalizes influence »,
    Machine Learning , 55(3), pages 251-270, 2004.]

Les forêts aléatoires

(*Random forests*)

Du bagging aux forêts aléatoires

Il conserve le **biais** en diminuant la **variance**

- Le **bagging** est favorisé par :
  - Des hypothèses faibles **différentes**
  - Des hypothèses faibles de **faible biais** (e.g. arbres de décision profonds)
  - On peut avoir un **nombre d'itérations élevé** sans sur-apprentissage (par augmentation de la marge)

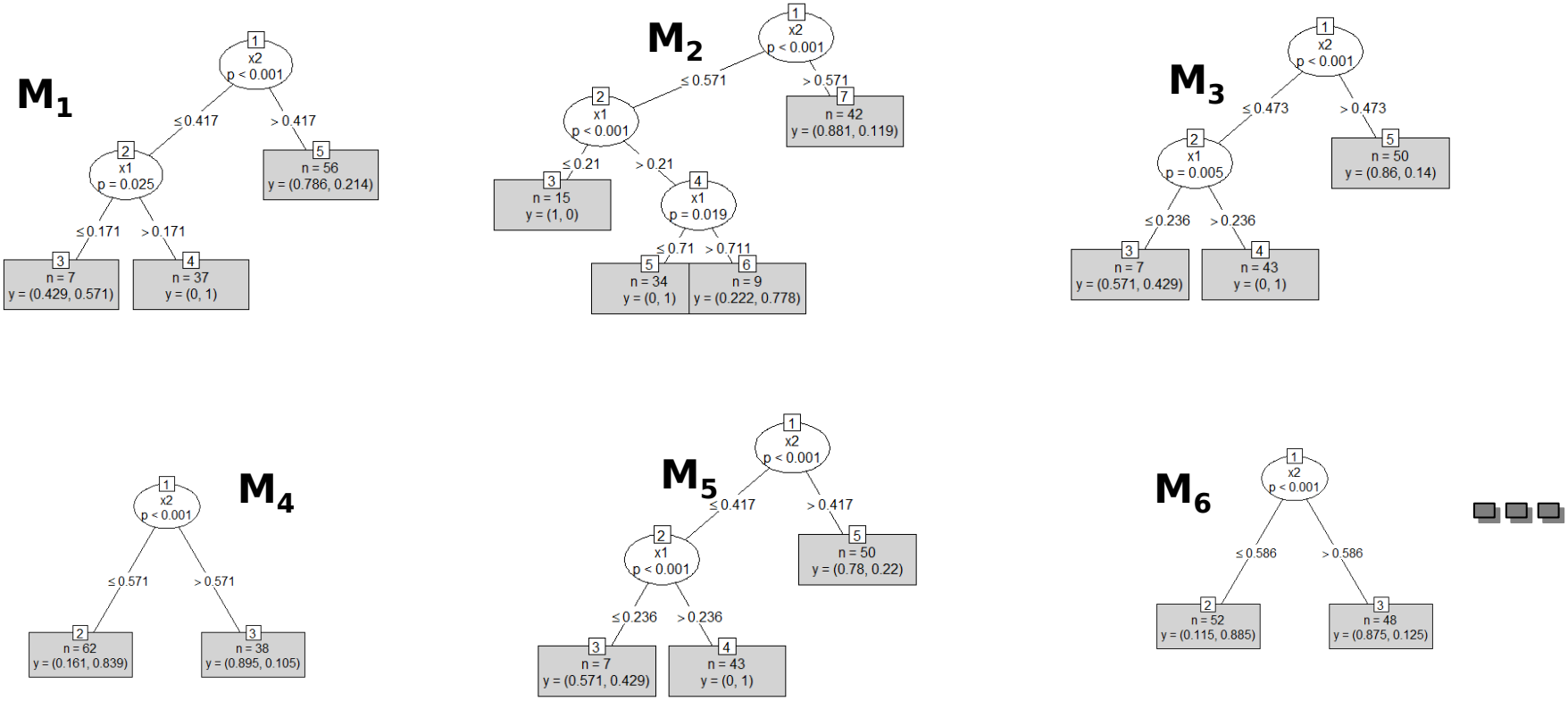
- D'où les **forêts aléatoires**
  - On va s'arranger pour avoir des arbres de décisions
    - Profonds
    - Différents
    - Nombreux

Les forêts aléatoires

- Principe :
  - **réduire la corrélation entre apprenants faibles**

- **Apprendre** des arbres
  - Sur des **jeux de données différents** (comme le bagging)
  - Sur des **sous-ensembles d'attributs** tirés aléatoirement

Interprétabilité et sélection d'attributs

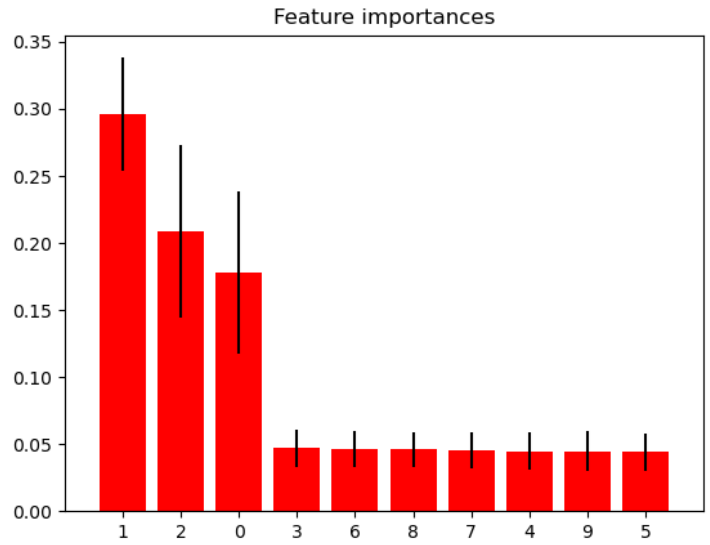


- Multitude d'arbres. Plus de lecture directe possible de l'importance des variables.
- On mesure la fréquence d'apparition

Interprétabilité et sélection d'attributs

- On **additionne** les importances de chaque attribut dans tous les arbres
  - Le **niveau** des attributs dans l'arbre
  - Le **gain** (d'entropie ou de Gini)
- On calcule **la moyenne** (ou la médiane) des importances
- Et on retient les attributs **dépassant cette valeur** seuil
  - Ou les ***N*** premiers attributs

```python
print(__doc__)

import numpy as np
import matplotlib.pyplot as plt

from sklearn.datasets import make_classification
from sklearn.ensemble import ExtraTreesClassifier

# Build a classification task using 3 informative features
X, y = make_classification(n_samples=1000,
                           n_features=10,
                           n_informative=3,
                           n_redundant=0,
                           n_repeated=0,
                           n_classes=2,
                           random_state=0,
                           shuffle=False)

# Build a forest and compute the impurity-based feature importances
forest = ExtraTreesClassifier(n_estimators=250,
                              random_state=0)

forest.fit(X, y)
importances = forest.feature_importances_
std = np.std([tree.feature_importances_ for tree in forest.estimators_],
             axis=0)
indices = np.argsort(importances)[::-1]

# Print the feature ranking
print("Feature ranking:")

for f in range(X.shape[1]):
    print("%d. feature %d (%f)" % (f + 1, indices[f], importances[indices[f]]))

# Plot the impurity-based feature importances of the forest
plt.figure()
plt.title("Feature importances")
plt.bar(range(X.shape[1]), importances[indices],
        color="r", yerr=std[indices], align="center")
plt.xticks(range(X.shape[1]), indices)
plt.xlim([-1, X.shape[1]])
plt.show()
```




```
Feature ranking:
1. feature 1 (0.295902)
2. feature 2 (0.208351)
3. feature 0 (0.177632)
4. feature 3 (0.047121)
5. feature 6 (0.046303)
6. feature 8 (0.046013)
7. feature 7 (0.045575)
8. feature 4 (0.044614)
9. feature 9 (0.044577)
10. feature 5 (0.043912)
```

https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html

Bilan sur les forêts aléatoires

- Souvent **puissant**

- **Mais**
  - Perte d'interprétabilité
  - Et plus coûteux en temps calcul

Conclusions

Conclusions

- Good if
  - Vectorial input space
  - The target concept corresponds to recursive boxes with borders parallel to the axes
- Very simple
- Computational complexity of learning in $O(A^2 \,.\, m)$
  - A attributes
  - m examples