



# Méthodes de classification supervisées

Les méthodes de segmentation

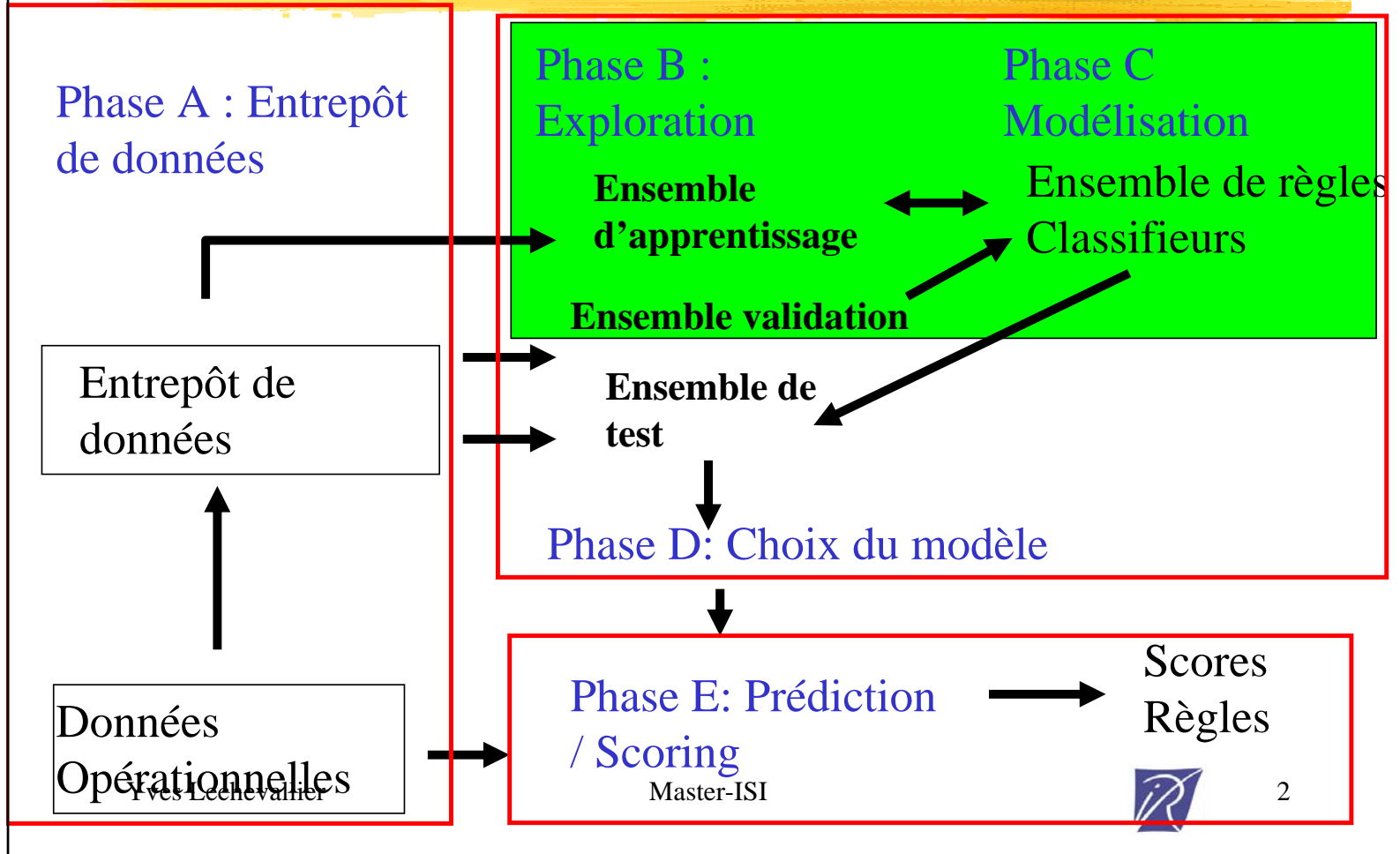
ou

les arbres de décision

Yves Lechevallier  
INRIA-Rocquencourt  
78153 Le Chesnay Cedex  
E\_mail : Yves.Lechevallier@inria.fr



# Processus Data Mining



# Méthodes de classement

## Discrimination

- Les **méthodes de classement** ont pour objet d'identifier la **classe d'appartenance** d'objets définis par leur description
- Un **objet à classer** est une entité appartenant à une population théorique  $\Pi$  constituant l'ensemble des objets susceptibles d'avoir à être classés. Cette population est supposée connue de façon exhaustive.



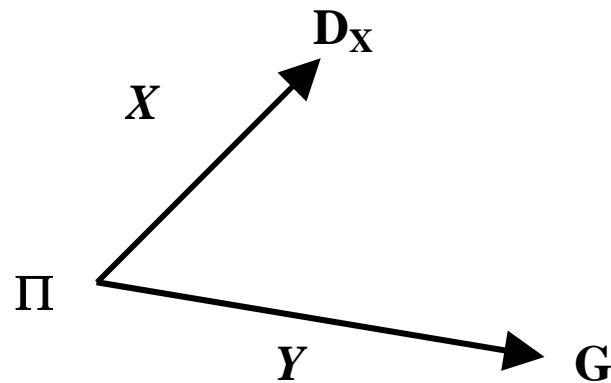
# Notations

$\Pi$  est muni d'une **partition**  $(\Pi_1, \dots, \Pi_K)$ .

- $G = \{1, \dots, K\}$
- $Y$  la **fonction de classement**
- $D_X$  **espace de description** (souvent  $R^p$ )
- Un couple  $(\mathbf{x}, y)$  où  $\mathbf{x}$  représente sa **description** et  $y$  l'indice de **sa classe d'appartenance**.



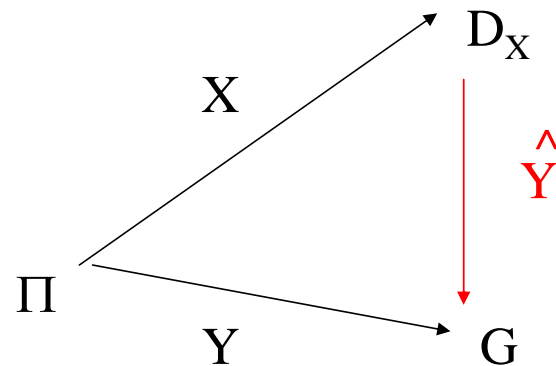
## couple «description, classe»



- Un couple  $(\mathbf{x}, y)$  où  $\mathbf{x}$  représente sa description et  $y$  l'indice de sa classe d'appartenance.

# Objectif des méthodes de classement

Trouver une procédure de classement  $\hat{Y}$ , dite **fonction de décision**, qui à toute description de  $D_X$  fournit l'indice d'une classe de  $\Pi$ .



Cette procédure devra être aussi **bonne** que possible et fournir le classement des objets de  $\Pi$  à partir de leur description.

# Fonction de décision

Toute **fonction de décision** induit sur une partition en classes  $(R_1, \dots, R_k, \dots, R_K)$  appelées **région d'affectation** de  $\hat{Y}$

$$R_k = \hat{Y}^{-1}(k) = \{x \in D_X / \hat{Y}(x) = k\}$$

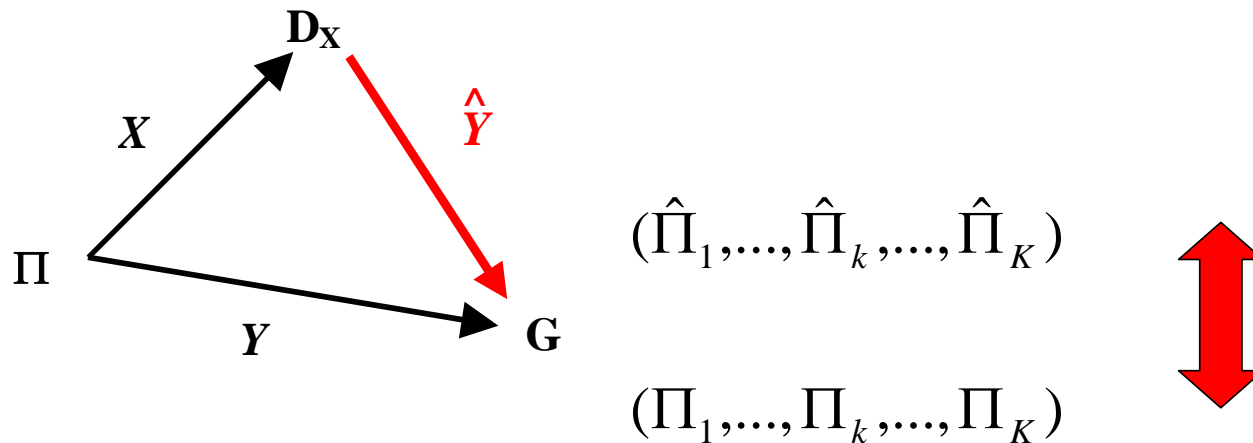
Pour un **descripteur**  $X$  et une **fonction de décision**  $\hat{Y}$  on peut définir sur  $\Pi$  une partition en  $K$  classes d'affectation.



# Fonction de décision $\hat{Y}$

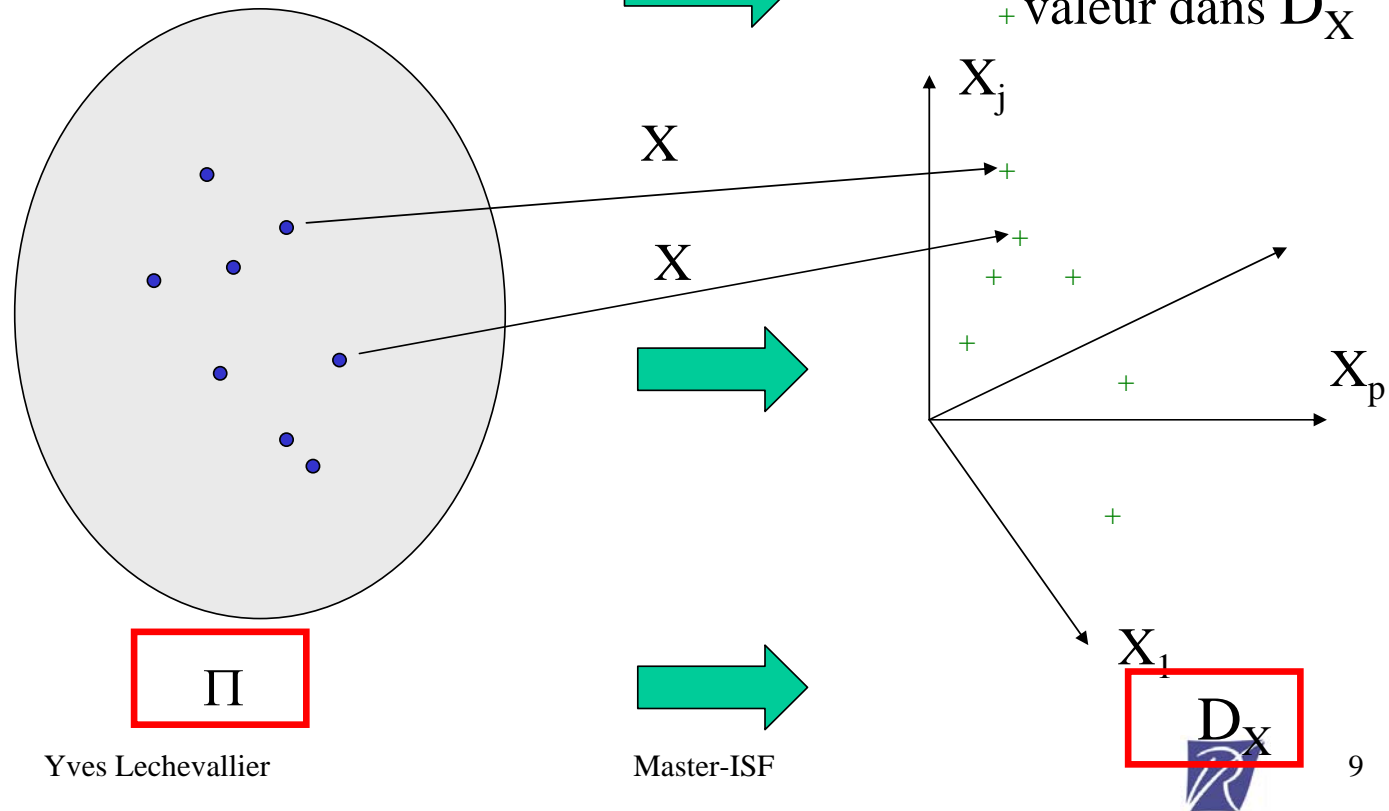
Tous les objets appartenant à une même classe d'affectation sont attribués de la même façon par  $\hat{Y}$

$$\hat{\Pi}_k = X^{-1} \circ \hat{Y}^{-1}(k) = X^{-1}(R_k)$$



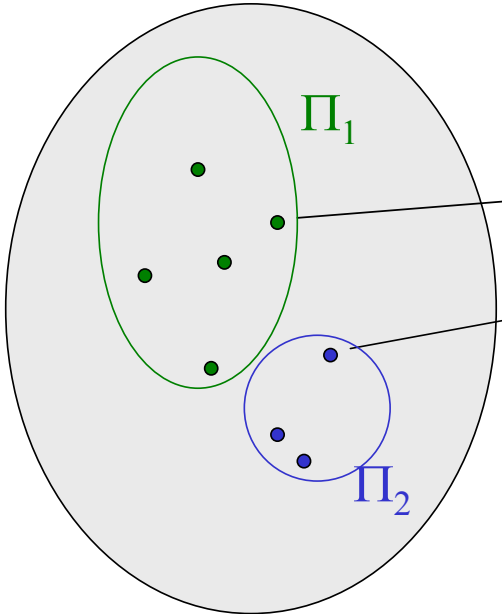
# Espace de description $D_X$

- élément de E



# Classes a priori

- élément de E



$\Pi$

Yves Lechevallier



$X,$

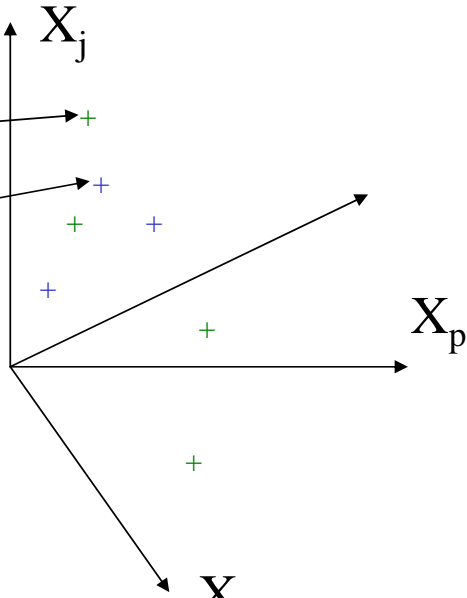
$Y$

$X,$

$Y$

Master-ISF

+ valeur dans  $D_X$

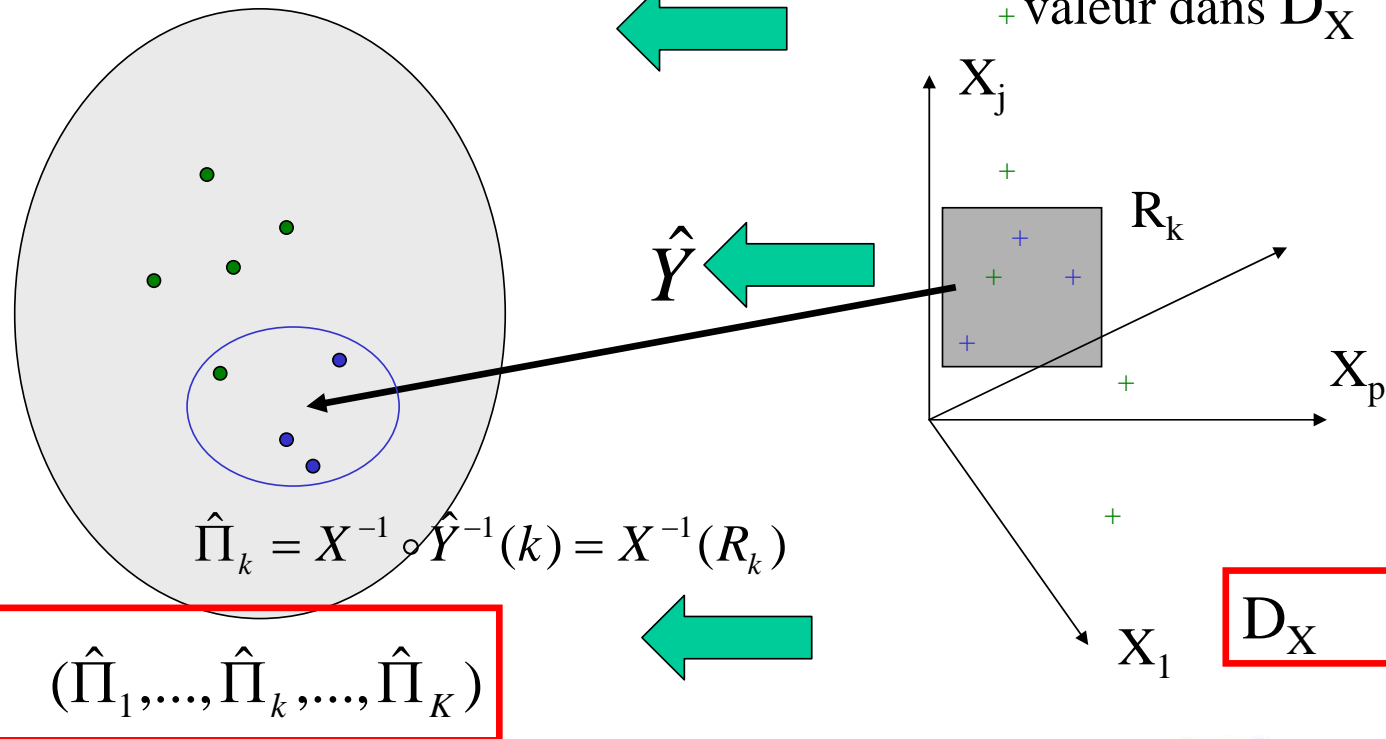


$D_X$



# Fonction de décision $\hat{Y}$

- élément de E



# Tableau de données

Tableau de données (modèle « vectoriel »)

Iris d'Anderson/Fisher

#	Sépale		Pétale		Espèce
	long.	larg.	long.	larg.	
1	5.1	3.5	1.4	0.2	setosa
51	7.0	3.2	4.7	1.4	versicolor
114	5.7	2.5	5.0	2.0	virginica



Représentation dans  $R^p$  de trois Iris.

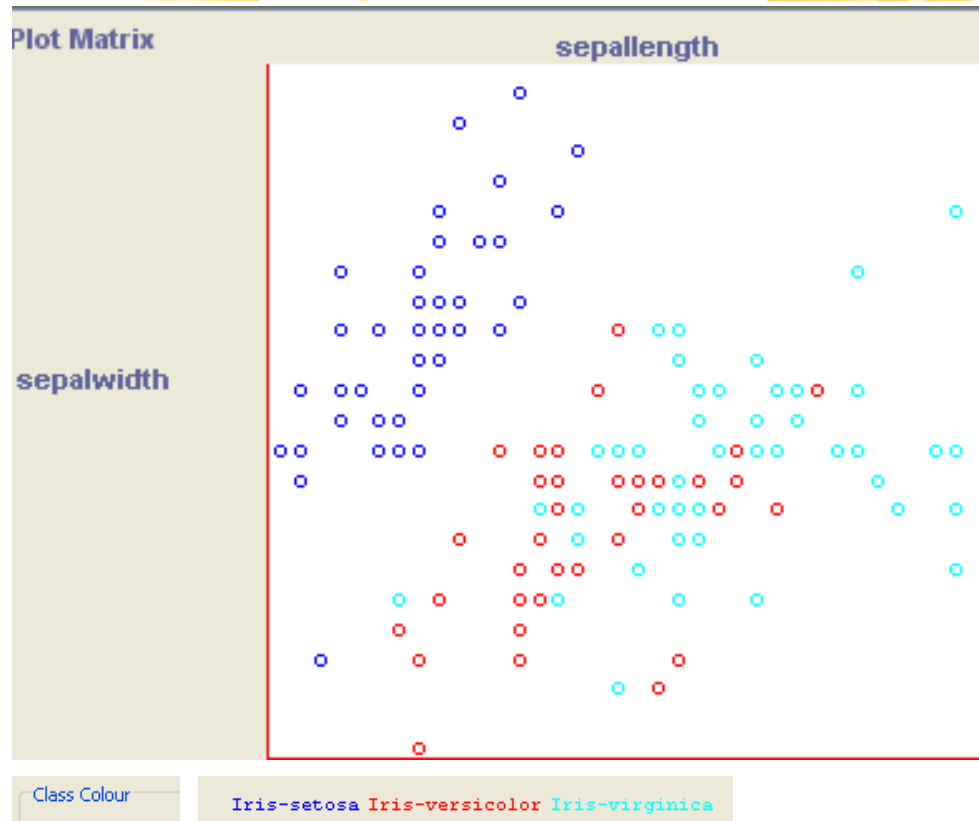


Web Site : <http://www.ics.uci.edu/~mlearn/MLSummary.html>

# Visualisation des iris

On sélectionne  
deux variables:  
Sepalwidth et  
Sepallength

Les trois classes  
sont représentées  
par 3 couleurs  
différentes



# Erreur de classement

A chaque **fonction de décision** on a une **règle de décision**

*Si  $\hat{Y}(x) = k$  alors  $x \in \hat{\Pi}_k$*

La performance globale  $R(\hat{Y})$  de la fonction de décision  $\hat{Y}$  est la moyenne des probabilités d'erreur de cette fonction de décision sur l'espace de description.

$$R(\hat{Y}) = \Pr[\hat{Y} \neq Y] = \sum_k \sum_{h \neq k} \Pr(\Pi_k \cap \hat{\Pi}_h) = 1 - \sum_k \Pr(\Pi_k \cap \hat{\Pi}_k)$$

La règle d'affectation  $\hat{Y}^*$  est la **règle de bayes d'erreur minimale** si elle est vérifiée :

$$\forall \hat{Y} \quad R(\hat{Y}) \geq R(\hat{Y}^*)$$



# Approche Bayésienne

- Probabilités a priori des classes  $\pi_k$
- Les lois de probabilité  $L_k(x)$  du vecteur  $x$  dans chaque classe a priori.
- Une fonction  $C$  de coût du classement d'un objet de la classe a priori  $P_k$  dans la classe d'affectation  $P_h$  coût  $C(h/k)$
- Une fonction de décision  $Y^*$ .



# Règle de Bayes d'erreur minimale

$$\forall x \quad Y^*(x) = k \text{ où } k \text{ est tel que } \Pr(k / x) = \max \Pr(h / x)$$

Cette définition est peu opérationnelle, en effet, on connaît rarement la probabilité d'un classement sachant une description.

**Théorème de Bayes**  $\Pr(k / x) = \frac{\pi_k L_k(x)}{L(x)}$

$$\pi_k = \Pr[Y = k]$$

$L_k(x) = \Pr[X = x / Y = k]$  est la densité de la classe  $k$

$$\forall x \quad Y^*(x) = k \text{ où } k \text{ est tel que } \Pr(k / x) = \max \pi_k L_k(x)$$



# Les descriptions suivent une loi normale

Le descripteur  $X$  des exemples est constitué de  $p$  descripteurs numériques et que sa distribution, conditionnellement aux classes, suit une **loi normale multidimensionnelle centrée** sur le vecteur  $\mu_k$  et de **matrice de variance-covariance**  $\Sigma_k$ .

**La vraisemblance conditionnelle de**  $X$  pour la classe  $k$  s'écrit alors

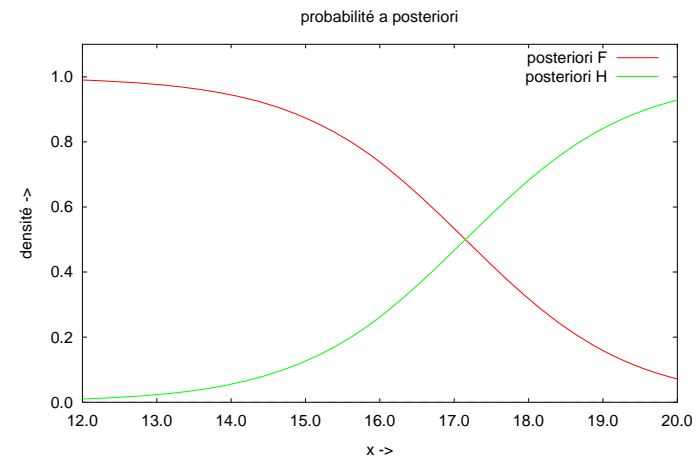
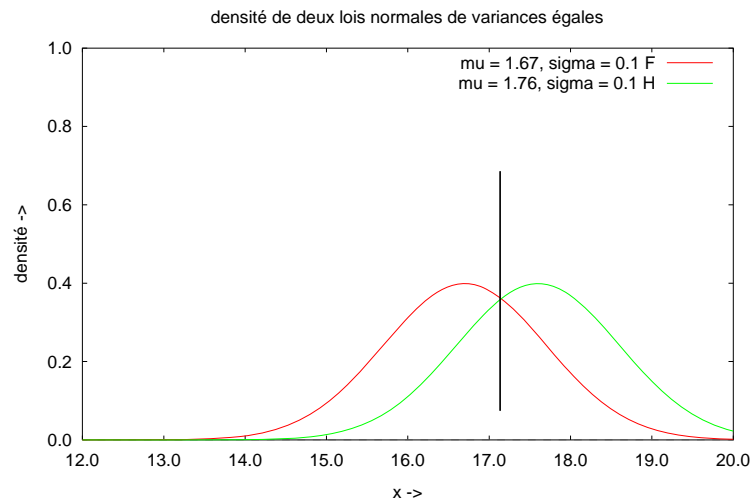
$$L_k(x) = \left( (2\pi)^p \det \Sigma_k \right)^{-\frac{1}{2}} \exp\left( -\frac{1}{2} (x - \mu_k)^t \Sigma_k^{-1} (x - \mu_k) \right)$$



# Exemple 1

$L_k(x)$

$$\Pr(k / x) = \frac{\pi_k L_k(x)}{L(x)}$$

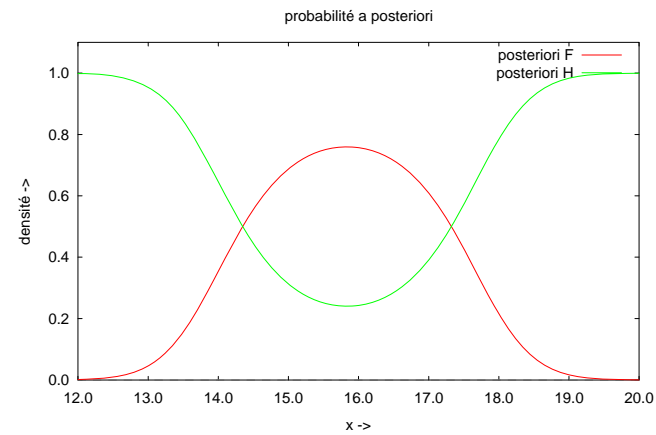
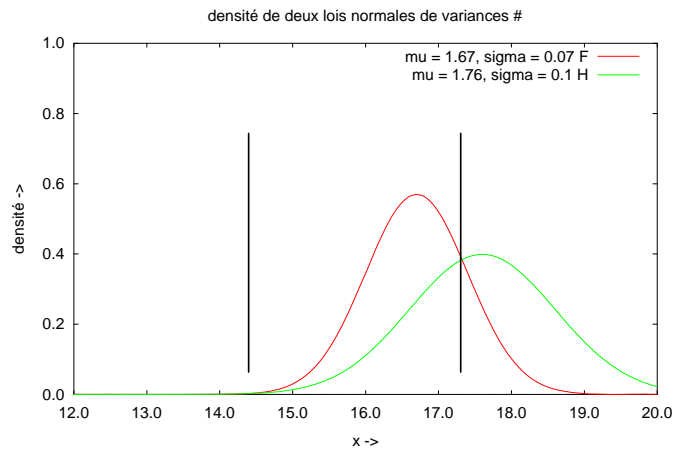


**Les variances et les probabilités a priori sont égales**

# Exemple 2

$$L_k(x)$$

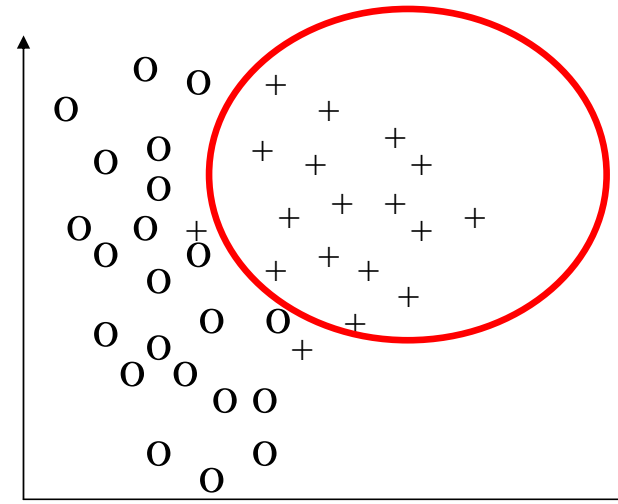
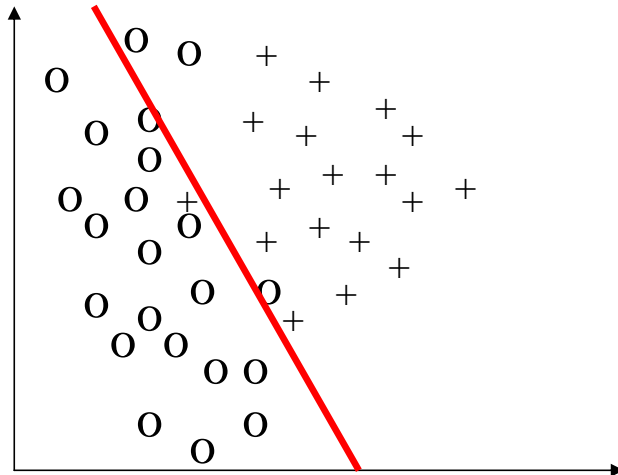
$$\Pr(k / x) = \frac{\pi_k L_k(x)}{L(x)}$$



**Les variances sont inégales égales**  
**Les probabilités a priori sont égales**

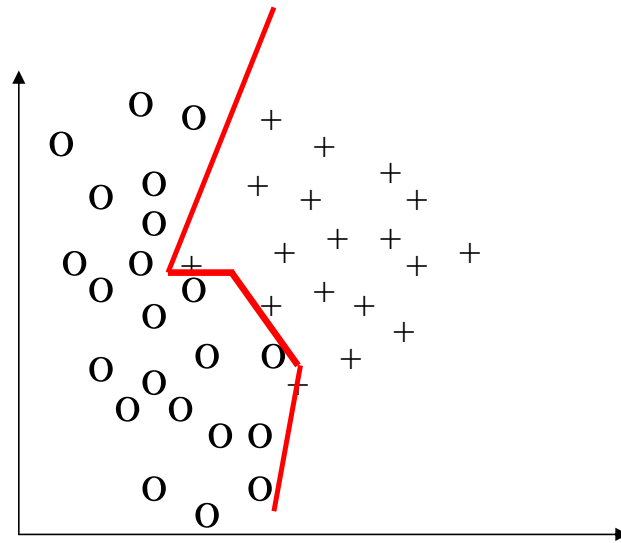
# Généralisation

Capacité de bien affecter de nouvelles données



**Modèle simple**

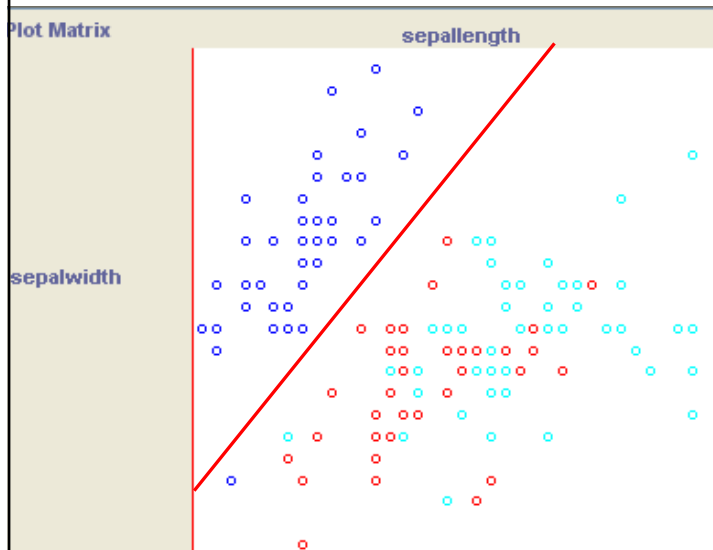
# Généralisation



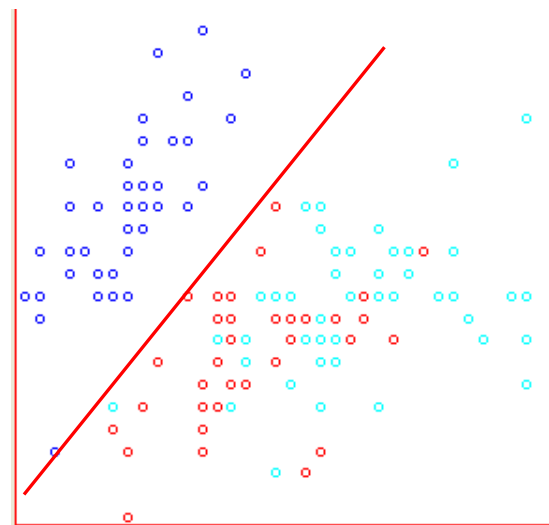
Modèle un peu trop flexible

**Complexité du modèle :** Comment adapter au mieux le modèle aux données sachant que l'on ne possède qu'un échantillon ?

# Complexité du modèle

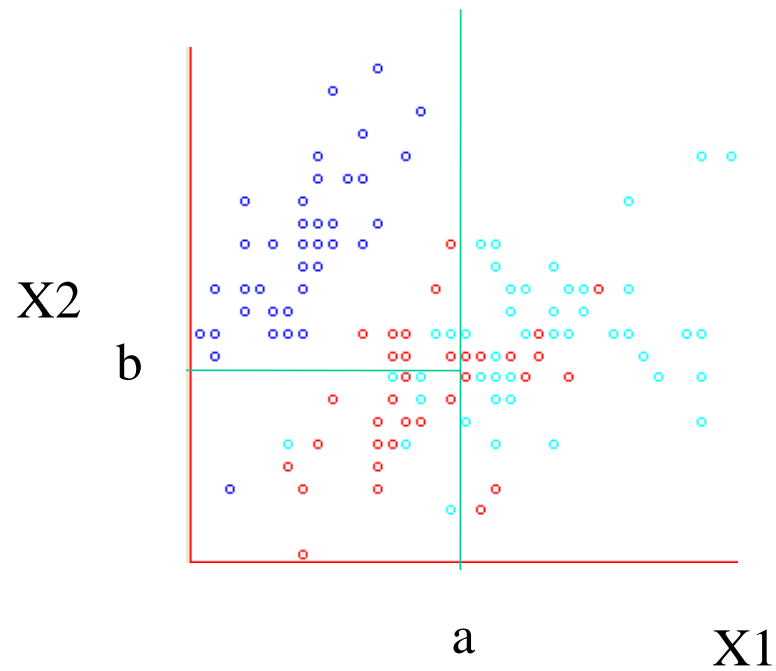


Analyse discriminante  
linéaire



Perceptron

# Comment améliorer cette solution ?

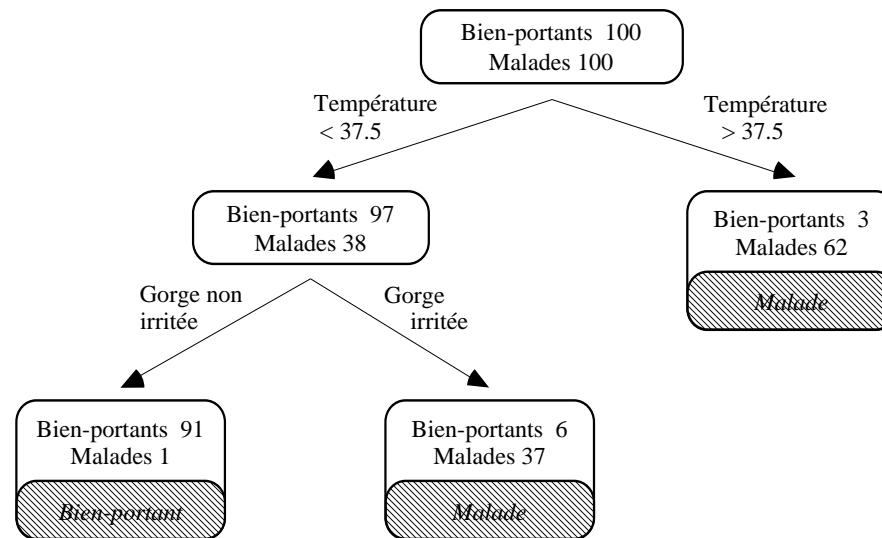


# Les arbres de décision

- Un **arbre de décision** est un enchaînement hiérarchique de **règles logiques ou de production** construites de manière automatique à partir de  $E$ .
- La construction de l'arbre de décision consiste à utiliser les descripteurs, pour les **subdiviser progressivement** l'ensemble  $E$  en sous-ensembles de plus en plus fins.



# Exemple d'arbre binaire



"Malade"

Règle 1

Règle 2

"Bien-portant"

Règle 3

Règle 1 : [température > 37.5]

Règle 2 : [température < 37.5] ET [gorge irritée]:

Règle 3 : température < 37.5] ET NON[gorge irritée]

# Segmentation/arbres de décision

- *Découpage successif* de  $E$  à l'aide d'une séquence de **règles de production**.
- Dans chaque sous-ensemble, une nouvelle *évaluation* est faite, celle-ci va permettre un nouveau découpage.
- Les ensembles terminaux sont appelés *feuilles* et les ensembles intermédiaires sont appelés *nœuds*.



# Construction d'un arbre de décision

- un mode d'écriture des **questions binaires**,
- une **règle d'étiquetage** de chacun des **segments terminaux**,
- un **critère d'évaluation** de la qualité d'une subdivision pour déterminer la meilleure subdivision d'un nœud intermédiaire,
- un **critère d'arrêt** permettant d'arrêter la construction de l'arbre et décider si un nœud est une **feuille**.



# Définition d'un arbre binaire

Un *arbre binaire* est défini par un triplet  $(T, g, d)$  constitué d'un ensemble  $T$  non vide d'entiers positifs et de deux fonctions  $g$  et  $d$  définies sur  $T$ . Les fonctions  $g$  et  $d$  respectent les deux propriétés caractéristiques :

$$(g(t) = 0 \wedge d(t) = 0) \vee (g(t) > 0 \wedge d(t) > 0)$$

Autre que le plus petit entier de  $T$ , il existe pour chaque  $t$  un élément unique  $s$  de  $T$  tel que

$$(t = g(s)) \vee (t = d(s))$$

Les éléments de  $T$  sont les *nœuds* de l'arbre et le plus petit élément de  $T$  la *racine* de l'arbre.



# Définitions

si  $(s = g(t)) \vee (s = d(t))$  alors le nœud  $t$  est appelé *père* de  $s$ .

si  $s = g(t)$  alors  $s$  est appelé *fils gauche* du nœud  $t$

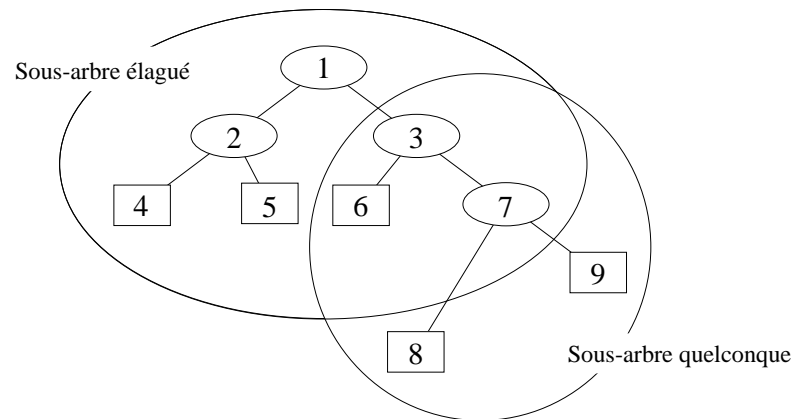
si  $s = d(t)$  alors  $s$  est appelé *fils droit* de  $t$

Si  $g(t) = 0 \wedge d(t) = 0$  le nœud  $t$  n'a pas de fils alors il est appelé *nœud terminal*.

Dans le cas contraire  $t$  est appelé *nœud non terminal* de  $T$ .

On note  $\tilde{T}$  l'ensemble des *segments terminaux* de  $T$ . A chaque segment terminal on peut associer une région de  $D_X$

# Sous-arbre



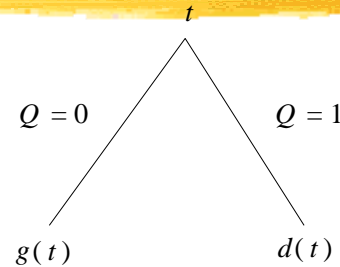
$T'$  est appelé *sous-arbre* de  $T$  si les trois éléments  $T'$ ,  $g'$  et  $d'$  définissent un arbre

Un sous-arbre est dit « élagué » s'il possède la racine et dans ce cas l'ensemble des segments terminaux forme une partition de  $D_X$

# Question binaire

variable continue

$[X > 3.5] ?$



Variable qualitative

$[X \in \{m_1, \dots, m_h\}]?$

- Dans le cas d'une **variable continue** on évalue toutes coupures possibles c'est-à-dire au maximum  $n-1$
- Pour une **variable qualitative ordonnée**  $Y$ , on évalue ainsi au maximum  $m-1$  bipartitions
- Dans le cas d'une **variable qualitative non ordonnée**, on se heurte vite à un problème de complexité, le nombre de dichotomies du domaine d'observation étant alors égal à  $2^m-1$ .

# Règle optimale d'étiquetage d'un segment terminal

Une règle d'étiquetage d'un arbre  $T$  est une application définie sur l'ensemble  $T'$  des **segments terminaux** de l'arbre  $T$  dans  $G$

$$\bar{C}(k/t) = \sum_{h \in G} C(k/h) \Pr(h/t)$$

La performance globale est mesurée par le risque associé à son utilisation

$$C_{\hat{Y}^*} = \sum_{t \in \tilde{T}} \bar{C}(\hat{Y}^*/t) \Pr(t)$$

À partir des fréquences empiriques

$$\hat{C}(k/t) = \sum_{h \in G} C(k/h) \frac{\nu_k(t)}{\nu(t)}$$

$$\hat{C}_{\hat{Y}} = \sum_{t \in \tilde{T}} \hat{C}(\hat{Y}/t) \frac{\nu(t)}{n}$$

# Choix d'un critère d'évaluation

Il y a  $K$  étiquetages possibles pour  $t$ . La moyenne pondérée des risques associés à ces différentes étiquettes s'écrit sous la forme.

$$\begin{aligned}\bar{C}(t) &= \sum_{k \in G} \bar{C}(k/t) \Pr(k/t) \\ &= \sum_{k \in G} \sum_{h \in G} C(k/h) \Pr(h/t) \Pr(k/t)\end{aligned}$$

Cette quantité représente également l'espérance mathématique du risque encouru à affecter aléatoirement les descriptions de  $t$  suivant la loi  $\Pr(\hat{Y}(x) = k / x \in t) = \Pr(k/t)$

Le gradient  $\Delta$  du risque, induit par une question  $Q$  au nœud  $t$

$$\Delta_{\bar{C}}(Q, t) = \bar{C}(t) - [\bar{C}(g(t)) \Pr(g(t)/t) + \bar{C}(d(t)) \Pr(d(t)/t)]$$

**Rechercher**  $\Delta_{\bar{C}}(Q^*, t) = \max_Q \Delta_{\bar{C}}(Q, t)$

## *Cas où les coûts d'un mauvais classement sont identiques*

Si les **coûts d'un mauvais classement** sont tous **identiques** alors le risque associé au segment  $t$  prend la forme de **l'indice d'impureté** de Gini utilisé dans CART

$$i(t) = \sum_{\substack{k \in G \\ h \in G \\ h \neq k}} \Pr(k / t) \Pr(h / t)$$

La notion d'**impureté** a été introduite par Breiman et *al.* [BRE84] et elle caractérise un concept très utile dans les méthodes de segmentation.



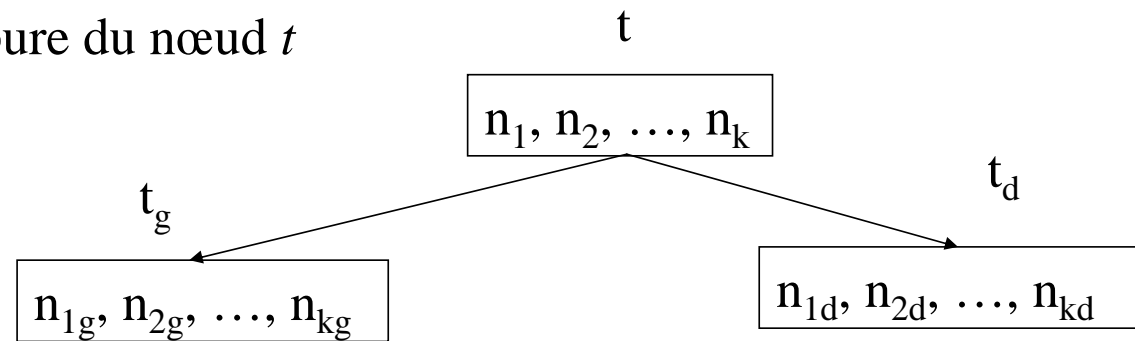
# Impureté

Pour mesurer la *qualité* d'une coupure au nœud  $t$  et le *pouvoir discriminant* de l'arbre on va utiliser la notion d'**impureté** qui caractérisera le degré de mélange du nœud.

Un nœud est dit **pur** si le segment qui lui est associé ne contient que des descriptions d'éléments d'une même classe. Inversement un segment est **d'impureté maximum** quand les  $K$  classes sont équiprobables dans ce segment.

# $i(t)$ : impureté d'un nœud $t$

Coupure du nœud  $t$



L'algorithme consiste à maximiser de diminution de l'impureté

$$\Delta i = i(t) - \left[ p(t_g)i(t_g) + p(t_d)i(t_d) \right]$$

avec  $p(t) = n(t) / N = \sum_{k=1}^K p(k, t) = p(t) \sum_{k=1}^K p(k / t)$  et  $p(t_g) = n(t_g) / n(t)$

# Propriétés de l'impureté

$$i(t) = \Phi(p(1/t), p(2/t), \dots, p(K/t))$$

- être une fonction **symétrique** des  $p(k/t)$
- être minimum si le nœud est **pur**
  - $(p(1/t), \dots, p(K/t)) = (1, 0, \dots, 0)$  ou  $(0, 1, \dots, 0)$  ou  $(0, \dots, 1)$
- Être maximum si le mélange est identique à la distribution de départ (**parfait**)
  - $(p(1/t), \dots, p(K/t)) = (n_1/n, n_2/n, \dots, n_K/n)$
- Être une fonction **concave** afin que la diminution d'impureté soit toujours positive ou nulle
  - La diminution est nulle si quel que soit  $k$  on a :  $p(k/t) = p(k/t_g)$

# Quelques définitions de l'impureté

- Indice de diversité de Gini (CART)

$$i(t) = \sum_{r=1}^K \sum_{s=1, s \neq r}^K p(r/t)p(s/t) = \left[ \sum_{r=1}^K p(r/t) \right]^2 - \sum_{r=1}^K p^2(r/t)$$

- L'entropie de Shannon (ID3)

$$i(i) = - \sum_{r=1}^K p(r/t) \log[p(r/t)]$$

# Impureté de l'arbre

$\tilde{T}$  est l'ensemble des nœuds terminaux, l'impureté de l'arbre  $T$  est:

$$I(T) = \sum_{t \in \tilde{T}} I(t) = \sum_{t \in \tilde{T}} i(t)p(t)$$

On a :  $\Delta I(T) = I(T) - I(T - t + t_g + t_d) = p(t)\Delta i(t)$

**Minimiser l'impureté à chaque coupure revient à minimiser l'impureté totale de l'arbre**



# Règle de décision

## Règle d'affectation d'un nœud

Le nœud  $t$  est affecté à la classe  $j$  si  $p(j/t)$  est supérieur à tous les  $p(k/t)$

$$r(t) = 1 - \max_k p(k/t) = 1 - p(j/t) = \sum_{r=1; r \neq j}^K p(r/t)$$

$r(t)$  est le taux apparent de mauvais classement du nœud  $t$

## Taux apparent de mauvais classement de l'arbre

$$R(T) = \sum_{t \in \tilde{T}} r(t)p(t)$$



# Critères d'arrêt de l'arbre

On arrête le découpage du nœud  $t$  si:

- $t$  est pur
- l'impureté est au dessous d'un seuil  $s$
- variation de l'impureté trop faible
- nombre d'individus dans  $t$  est trop faible
- $t$  est presque pur

On obtient ainsi l'**arbre maximal**  $T_{max}$



# Validation de l'arbre

La croissance de l'arbre permet de faire converger l'estimateur de  $\sum_t p(t)p(k/t)$  vers  $p(k/x)$

Au nœud  $t$  quand le nombre de nœuds croît

il y a une réduction du biais  $E\left(\frac{n_k(t)}{n(t)}\right) = p(k/t)$

une augmentation de la variance  $Var\left(\frac{n_k(t)}{n(t)}\right) = \frac{1}{n(t)} p(k/t)(1 - p(k/t))$

Compromis biais/variance



# Le compromis biais/variance

- la complexité du modèle est elle suffisante pour réaliser une approximation correcte de la fonction de décision  $Y^*$ ?
- L'erreur d'estimation réalisée sur l'échantillon est un bon indicateur de la performance du modèle sur les données futures?
- L'estimation de  $Y^*$  est elle très dépendante de l'échantillon?



# Recherche de l'arbre optimal

## Élagage de l'arbre:

L'arbre maximal  $T_{\max}$  est construit en minimisant l'impureté. L'arbre est trop développé pour être robuste

En élaguant progressivement l'arbre maximal, on construit une suite de sous-arbres qui sont tous **emboîtés** avec l'arbre maximal



# Critère de réduction de la complexité

$$R_\alpha(T) = R(T) + \alpha |\tilde{T}|$$

$\alpha$  est un coefficient de pénalité

Pour  $\alpha$  donné, on choisit l'arbre élagué  $T_\alpha$  optimal en minimisant, sur l'échantillon test ou par validation croisée, le risque moyen  $R_\alpha(T)$ . Puis, parmi ces arbres, le meilleur est retenu.

- Existe-t-il parmi ces sous-arbres un arbre  $T$  qui minimise ce risque?
- Est-il possible de construire un algorithme d'élagage efficace ?



# Avantages des arbres

- **Avantages**
  - Méthode est non paramétrique et insensible aux valeurs extrêmes
  - Elle permet de traiter de variables de natures différentes
  - Elle comporte une sélection des variables
  - Elle détermine des sous-populations définies par des règles facilement interprétables.
  - On peut isoler certains nœuds et définir des classes de risque



# Inconvénients des arbres

- **Inconvénients**

- La méthode peut être peu robuste car elle sélectionne pas à pas les variables
- Elle est liée à la définition de seuils donc elle est sensible à de légères perturbations sur les données
- La construction est assez délicate en particulier au moment de l'élagage. Il est difficile de sélectionner l'arbre optimal

