Apprentissage supervisé
Arbres de classification.

Marie Chavent

Université de Bordeaux

Introduction

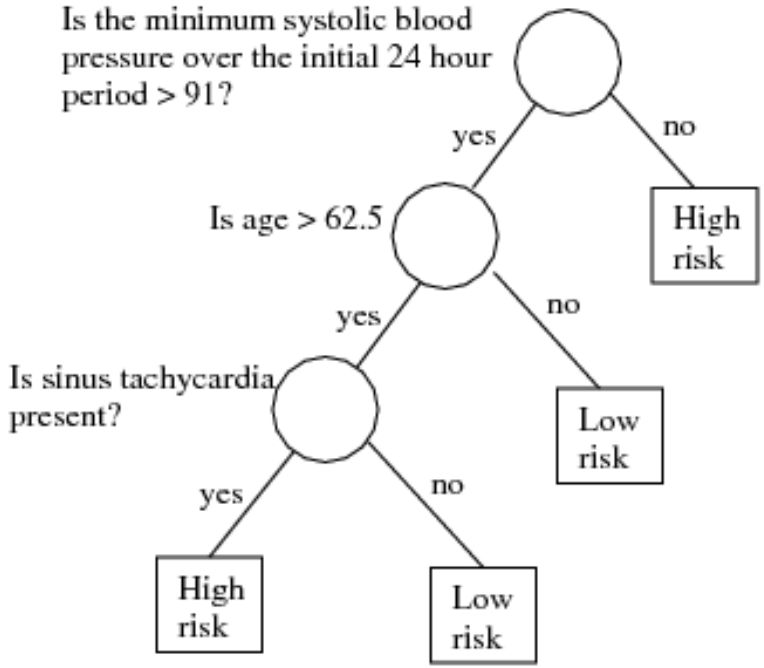
Deux approches possibles pour constuire une règle de classification g.

- Approche basée sur un modèle.
  - Apprentissage de la $Loi(Y|X)$ puis déduction de g
  - Exemples : analyse discriminante linéaire, bayésien naïf, régression logistique, etc.
- Approche de type prototype.
  - Apprentissage direct de la règle classification g
  - Exemples : k-plus proches voisins, arbres de classification, forêts aléatoires, etc.
- Dans ce chapitre : arbres de classification
  - CART : Classification and regression trees. L. Breiman, J. H. Friedman, R.A. Olshen, and C. J. Stone, Chapman & Hall, 1984.
  - On plus généralement d'arbres de décision (classification et régression).

La méthode CART en classification supervisée

- Variables d'entrées quantitatives ou qualitatives $X = (X^1, \ldots, Xp) \in \mathcal{X}$.
- Variable de sortie Y qualitative à K modalités définissant les K classes à prédire.
- La règle de classification $g : \mathcal{X} \to \{1, \ldots, K\}$ est un un arbre de classification constuit à partir des données d'apprentissage (X_i, Y_i), $i = 1, \ldots, n$.

Exemple :



Plan

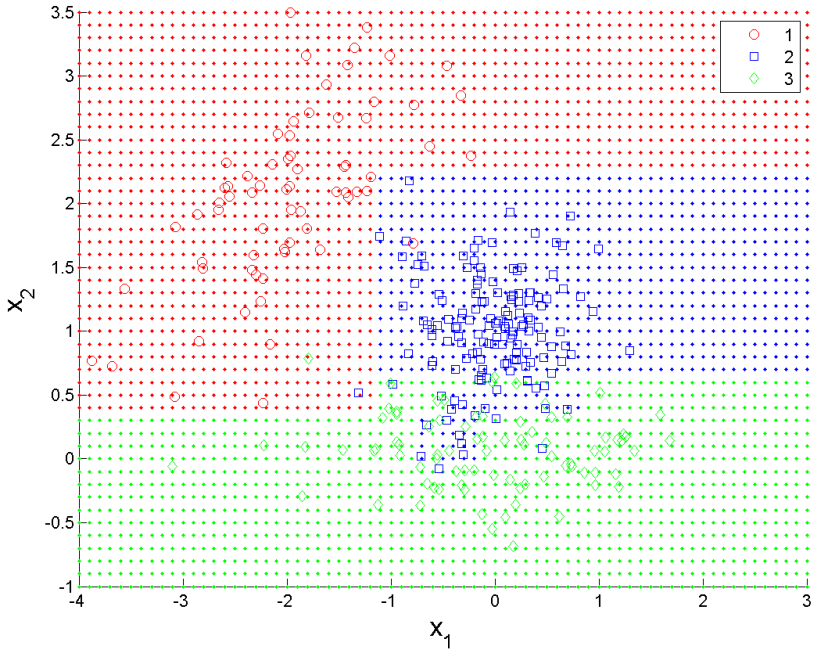
1. Croissance de l'arbre.
2. Evaluation de sa performance.
3. Elaguage.

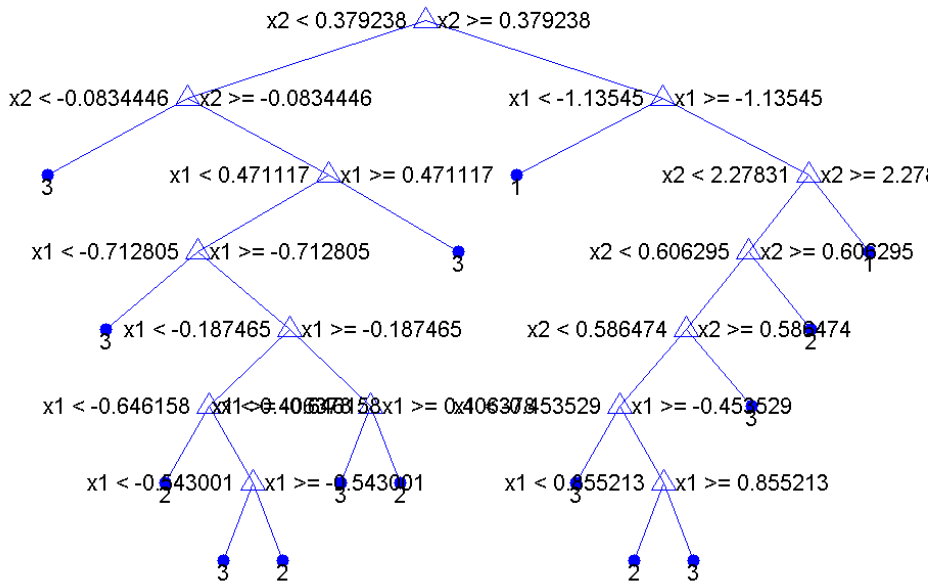
Construction de l'arbre

La méthode CART construit un arbre binaire dont les noeuds sont des sous-échantillons des données d'apprentissage.

- Le noeud racine contient toutes les données d'apprentissage.
- A chaque étape noeud est divisé pour construire deux nouveaux noeuds les plus homogènes possible vis à vis de la variable à expliquer.
- L'arbre maximal est obtenu lorsqu'aucun noeud ne peux plus être divisé. Un noeud terminal (qui ne peut plus être divisé) est appellée une feuille.
- Chaque feuille est alors associée à l'une des classes de la variable à expliquer aussi appellée étiquette.

Exemple : données synthétiques

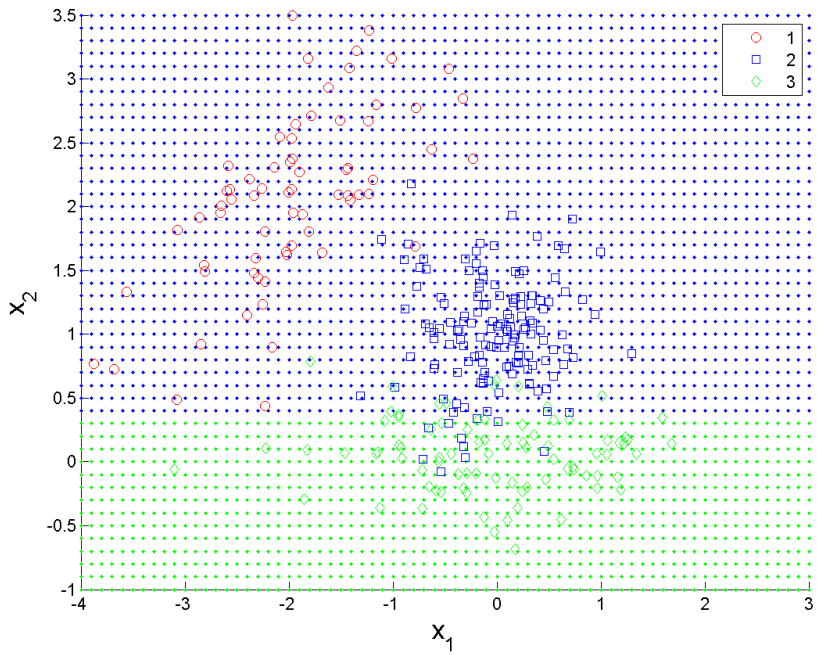


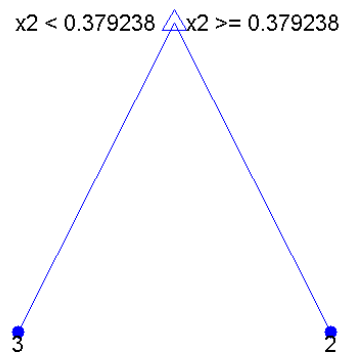


Comment mesurer la qualité d'une division ?

On veut diviser un noeud t en deux sous-noeuds t_L (noeud fils gauche) et t_R (noeud fils droit) qui soient le plus homogènes possible ou encore le moins hétérogène possible vis à vis de la variable à expliquer Y.

Exemple : première division





L'hétérogénéité d'un noeud se mesure à partir d'une fonction d'impureté ϕ définie sur l'ensemble des K-uplets $(p_1, \ldots, p_K)$ satisfaisants $p_k \geq 0$ pour $k = 1 \ldots, K$ et $\sum_{k=1}^{K} p_K = 1$ avec :

- ϕ admet un unique maximum en $(\frac{1}{K}, \ldots, \frac{1}{K})$
- ϕ est minimum aux points $(1, 0, \ldots, 0)$, $(0, 1, \ldots, 0)$...
- ϕ est une fonction symétrique de $p_1, \ldots, p_K$ c'est à dire que ϕ est constante pour toute permutation de p_k.

On définit alors l'impureté $i(t)$ d'un noeud t par :

$$i(t) = \phi(p(1|t), \ldots, p(K|t))$$

où $p(k|t)$ est la probabilité d'avoir l'étiquette k sachant qu'on est dans la cellule correspondant au noeud t.

On estimera ces probabilités sur les données par la proportion de la classe k dans le noeud t :

$$\hat{p}(k|t) = \frac{n_{t,k}}{n_t}$$

L'impureté d'un noeud t est :

- toujours positive ou nulle.
- nulle si toutes les observations du noeud appartiennent à la même classe de Y. On dira que le noeud est pur.
- maximale lorsque les classes de Y sont équiprobables dans le noeud. On dira que le noeud est impur.

Les deux mesures d'impureté standards sont :

- l'indice de Gini :

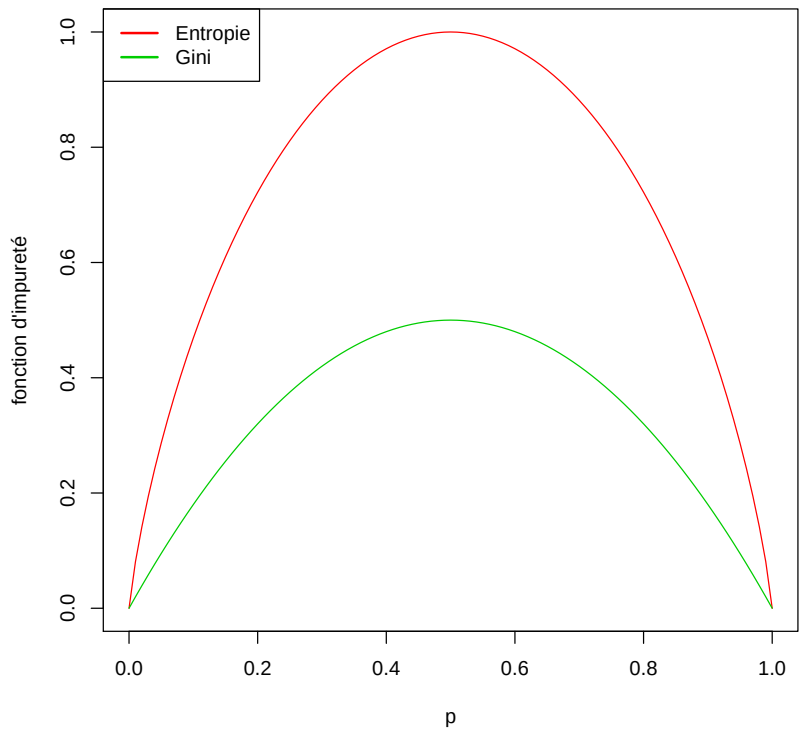
$$i(t) = \sum_{k=1}^{K} p(k|t)(1 - p(k|t)) = 1 - \sum_{k=1}^{K} p(k|t)^2$$

- l'entropie (avec la convention $0 \log(0) = 0$) :

$$i(t) = -\sum_{k=1}^{K} p(k|t) \log_2(p(k|t))$$

Par exemple, si la variable Y est binaire, en notant $p = p(1|t)$ on a :

- Gini : $i(t) = 2p(1-p)$
- Entropie : $i(t) = -p\log_2(p) - (1-p)\log_2(1-p)$



La qualité de la division (t_L, t_R) du noeud t est la réduction de l'impureté induite par cette division :

$$\Delta(t_L, t_R) = i(t) - p_L i(t_L) - p_R i(t_R)$$

où p_L (resp. p_R) est la probabilité qu'une donnée appartienne à la cellule t_L (resp. t_R) sachant qu'elle se trouvait dans la cellule t.

On estimera ces probabilités sur les données par :

$$\hat{p}_L = \frac{n_L}{n}, \quad \hat{p}_R = \frac{n_R}{n}$$

où n est le nombre de données dans le noeud t, n_L dans le noeud t_L et n_R dans le noeud t_R.

Une bonne division occasionera une forte diminution de l'impureté.

Comment diviser un noeud ?

L'algorithme consiste à choisir parmi toutes les divisions (t_L, t_R) possibles, celle qui maximise $\Delta(t_L, t_R)$ c'est à dire qui maximise la diminution de l'impureté.

Ici les divisions (t_L, t_R) d'un noeud t sont induites par des questions binaires.
Une question binaire est définie à partir d'une variable explicative X^j de la manière suivante :

- Si $X^j \in \mathbb{R}$ est quantitative, la question binaire sera du type

$$X^j \leq c \, ?$$

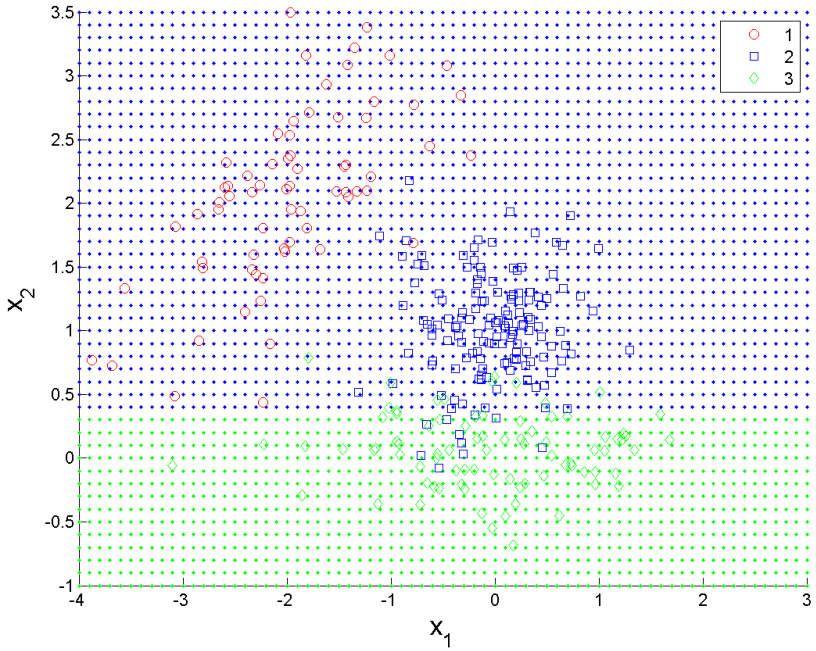
  Il existe une infinité de valeurs de coupures c possibles mais elles induisent au maximum $n_t - 1$ divisions différentes.

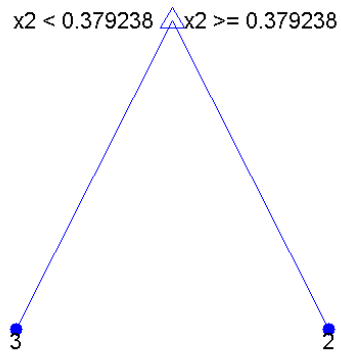
- Si $X^j \in \{1, \ldots, M\}$ est qualitative, la question binaire sera du type

$$X^j \subset A \, ?$$

  où $A \subset \{1, \ldots, M\}$. Il existe $2^{M-1} - 1$ questions binaires et donc au maximum $2^{M-1} - 1$ divisions différentes.

Exemple : première division





- Combien de questions binaires ont été évaluées ici si $n = 100$ données dans le noeud racine ?
- Quelle est la meilleure question binaire finalement retenue ?

Comment associer une étiquette (une classe) à un noeud ?

On notera $\tau(t)$ la classe associée au noeud terminal t.

- $\tau(t)$ est la classe la plus probable à posteriori pour une fonction de coût 0-1 :

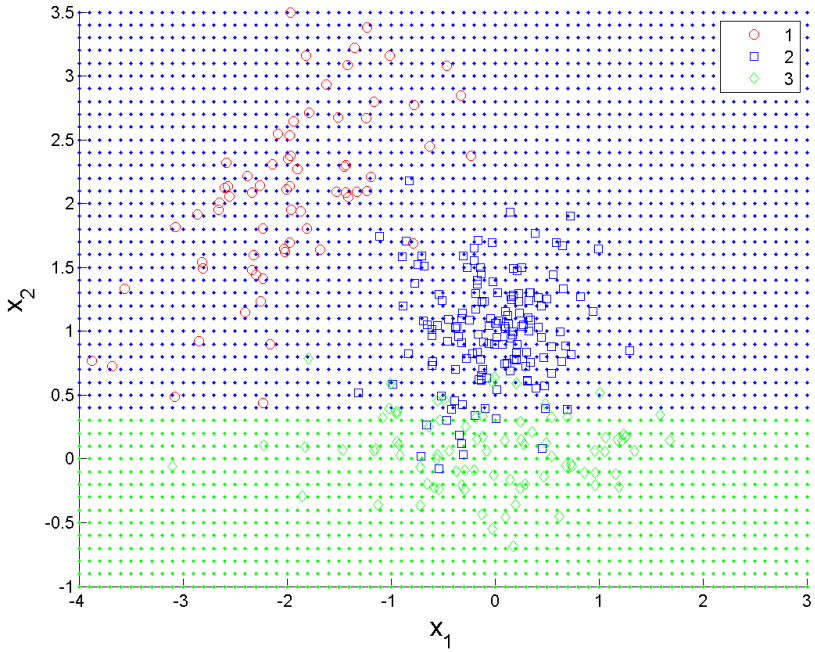
$$\tau(t) = \underset{\ell \in \{1,\dots,K\}}{\arg\max}\ p(k|t)$$

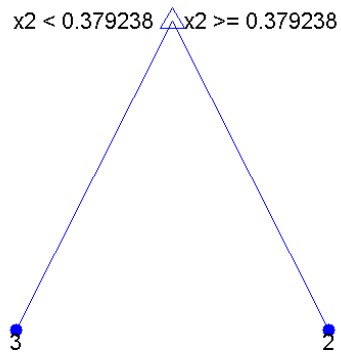
  $\tau(t)$ est alors simplement classe majoritaire.

- $\tau(t)$ est la classe la moins risquée à posteriori pour une fonction de coût quelconque :

$$\tau(t) = \underset{\ell \in \{1,\dots,K\}}{\arg\min} \sum_{k=1}^{K} C_{k\ell}\ p(k|t)$$

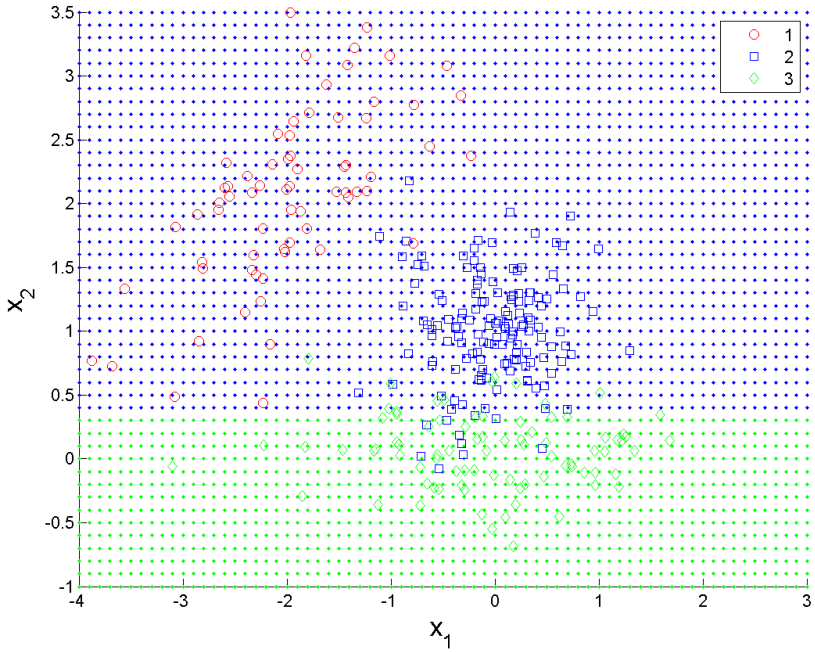
Exemple : première division

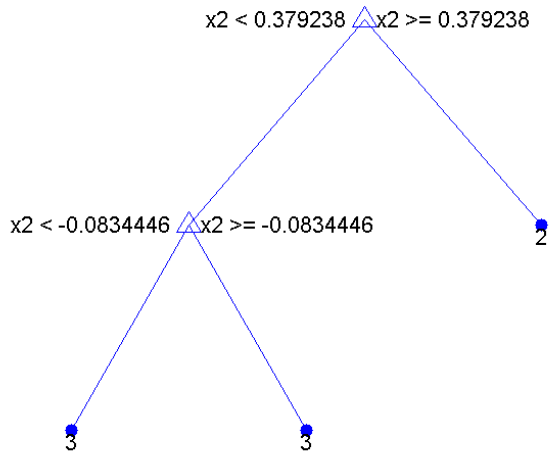




- Quelles sont les étiquettes des noeuds fils gauche et droite ?
- Pourquoi ?

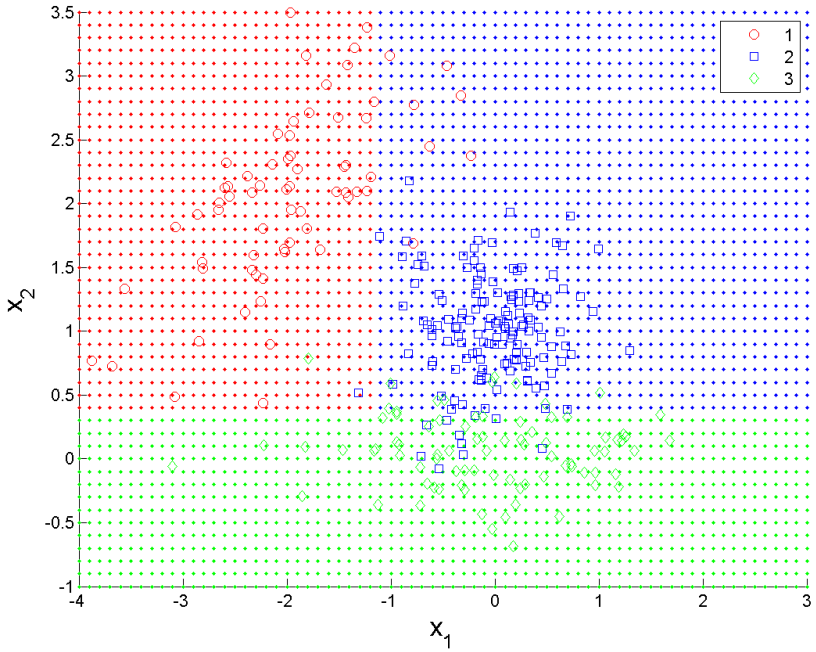
Exemple : deuxième division

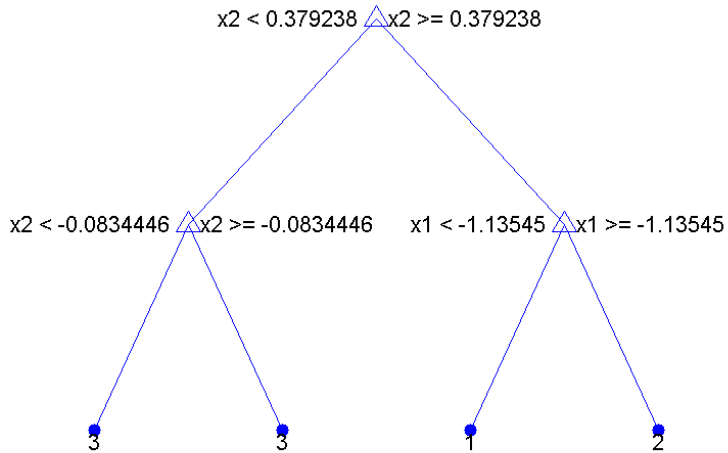




- Tracer un droite verticale ou horizontale sur le graphique de gauche pour visualiser cette seconde division.
- Quelle est le noeud pur à l'issue de cette division ?
- Ce noeud peux-il être redivisé ?

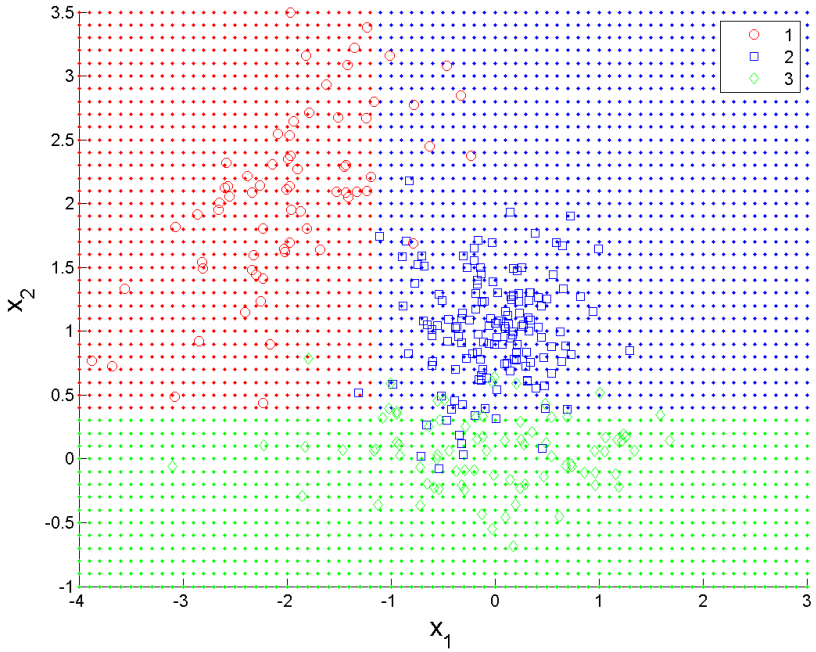
Exemple : troisième division

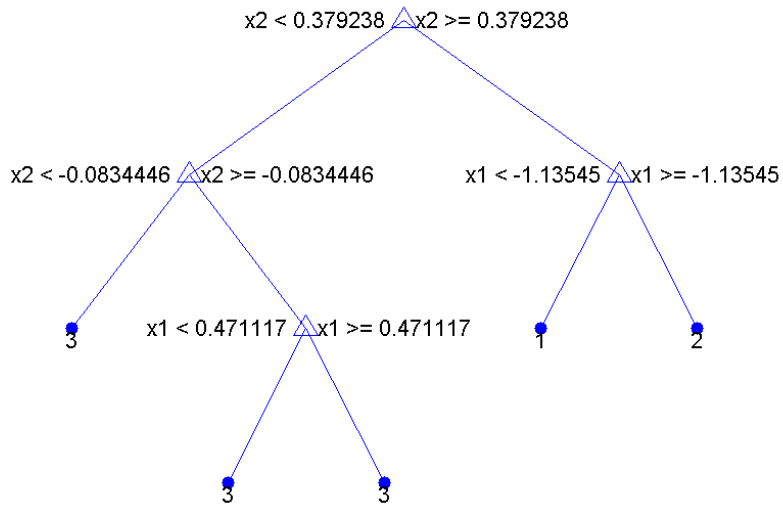




- Tracer sur le graphique de gauche les cellules (zones de $\mathbb{R}^2$ ici) associées aux 4 noeuds terminaux.
- Quelle noeuds peuvent être redivisés ?

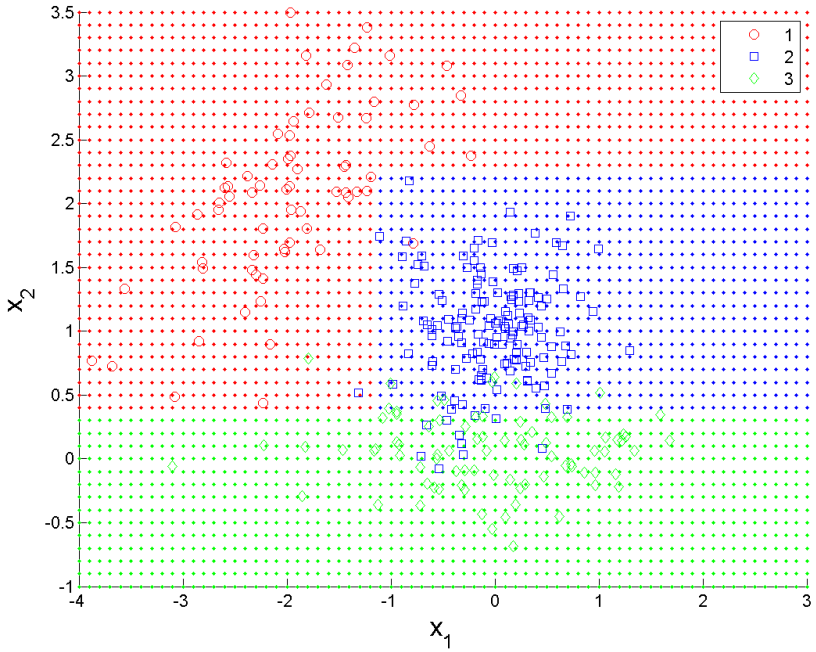
Exemple : quatrième division

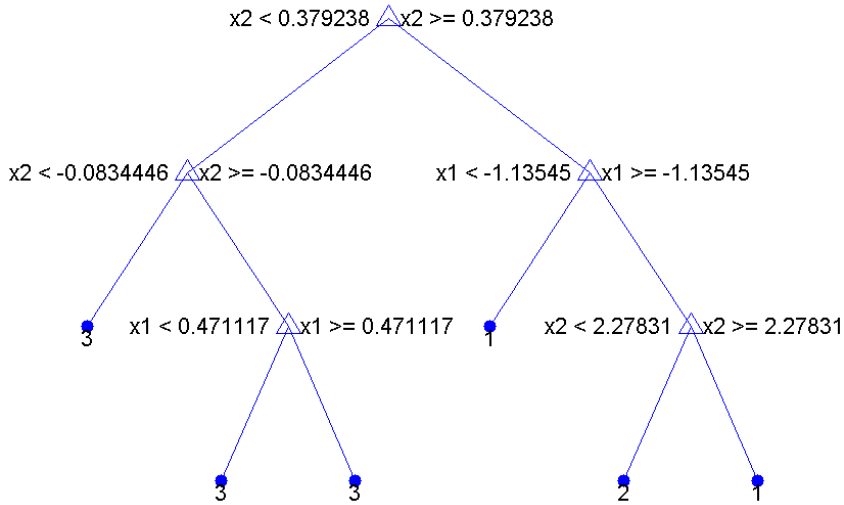




- Tracer sur le graphique de gauche les cellules (zones de $\mathbb{R}^2$ ici) associées aux 5 noeuds terminaux.
- Quelle classe est prédite par cet arbre pour une nouvelle données $\mathbf{x} = (0, 3)$?
- Quelle noeuds peuvent être redivisés ?

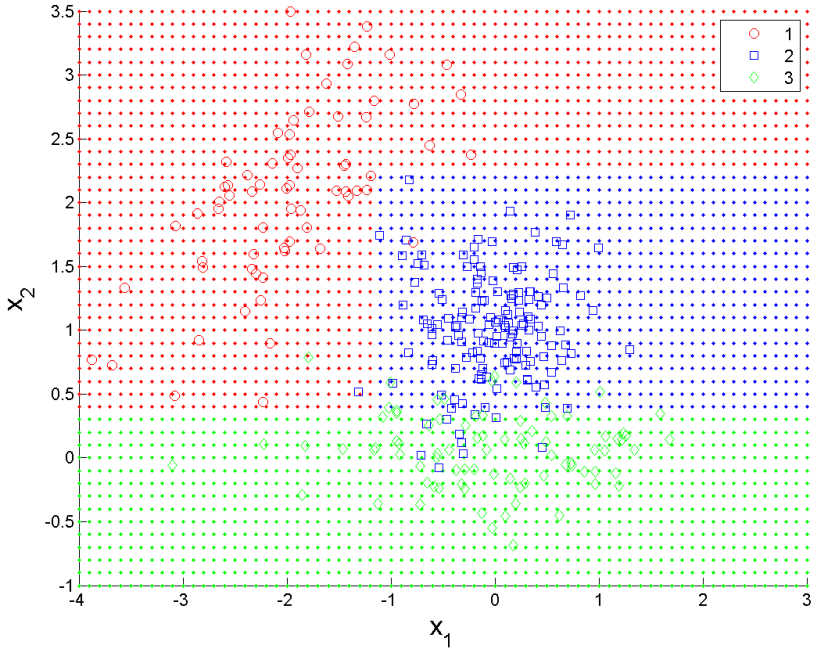
Exemple : cinquième division

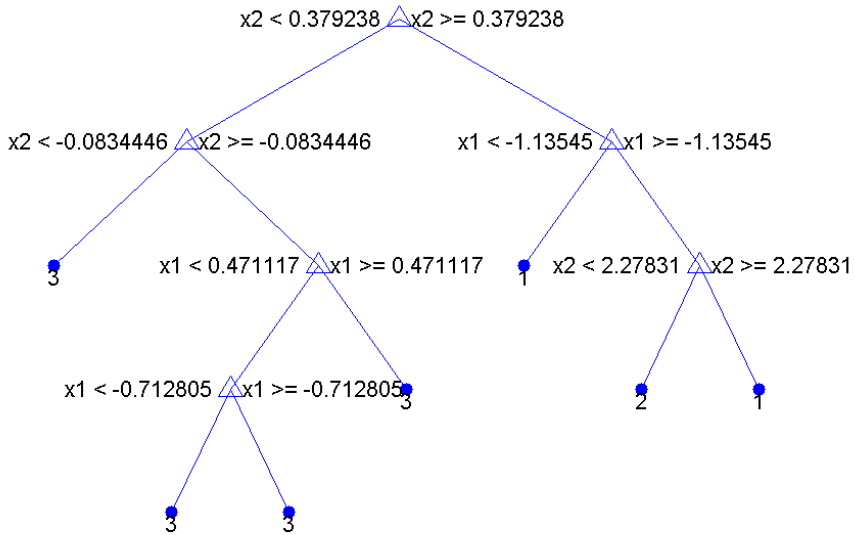




- Tracer sur le graphique de gauche les cellules (zones de $\mathbb{R}^2$ ici) associées aux 6 noeuds terminaux.
- Quelle classe est maintenant prédite par cet arbre pour la nouvelle données $\mathbf{x} = (0, 3)$?
- Quelle noeuds peuvent être redivisés ?

Exemple : sixième division





- Tracer sur le graphique de gauche les cellules (zones de $\mathbb{R}^2$ ici) associées aux 7 noeuds terminaux.
- Quelle noeuds peuvent être redivisés ?
- Quand arrêter les divisions ?

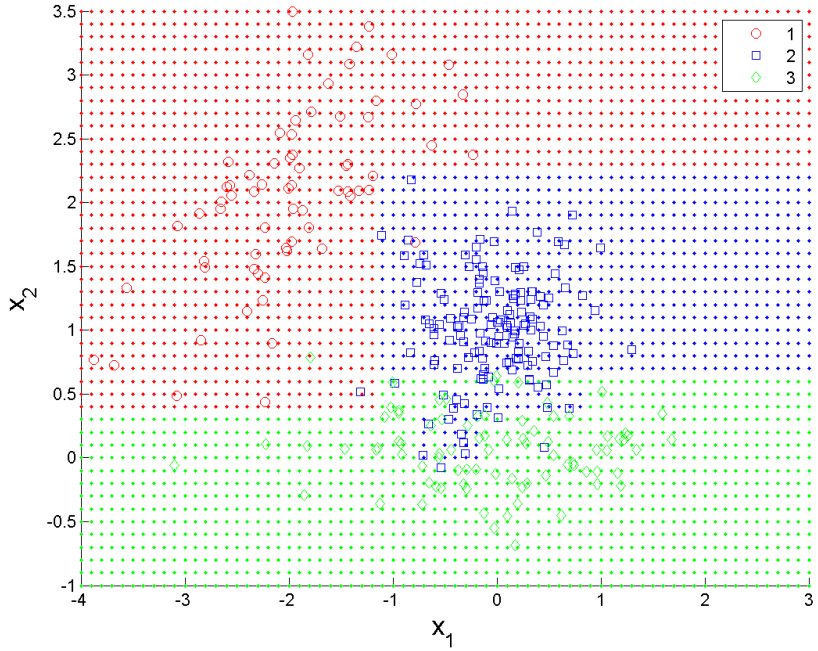
Arrêt des divisions.

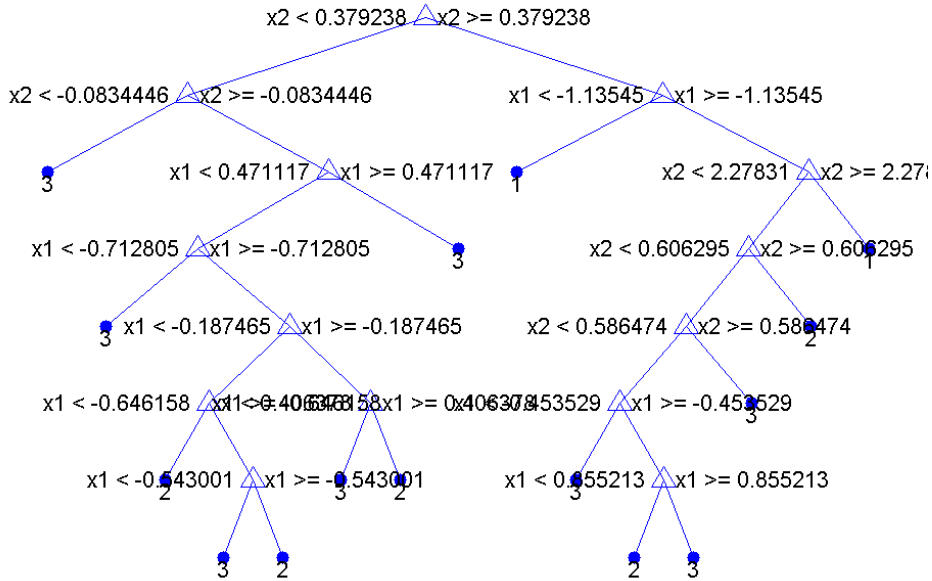
Le critère d'arrêt peut-être :

- ne pas découper un noeud pur,
- ne pas découper un noeud qui contient moins de n_{min} données avec souvent n_{min} compris entre 1 et 5.

L'arbre ainsi obtenu est appellé l'arbre le longueur maximale.

Exemple : quatorzième division





- D'après vous s'agit-il de l'arbre de longeur maximale ?
- L'arbre de longueur maximale sera-il bon pour prédire de nouvelles données ?

Evaluation de la performance d'un arbre

Le risque théorique d'une feuille t d'étiquette $\tau(t)$ est défini par :

$$r(t) = \sum_{k=1}^{K} C_{k\tau(t)}\ p(k|t)$$

où $C_{k\tau(t)}$ est le coût de mauvaise classification d'une donnée de la classe k dans la classe $\tau(t)$ et $p(k|t)$ est la probabilité d'appartenir à la classe k sachant qu'on est dans le noeud t.

Le risque théorique de l'arbre T est alors définit par :

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

où $\tilde{T}$ est l'ensemble des feuilles de l'arbre T et $p(t)$ est la probabilité d'appartenir à la feuille t et $r(t)$ est le risque du noeud t.

On considère maintenant un échantillon d'observations $(X_1, Y_1), \ldots, (X_n, Y_n)$.

Dans le cas d'une matrice de coût quelconque, le risque empirique d'une feuille est le coût moyen de mauvais classement dans la feuille :

$$\begin{aligned}\hat{r}(t) &= \sum_{k=1}^{K} C_{k\tau(t)} \frac{n_{t,k}}{n_t} \\ &= \frac{1}{n_t} \sum_{x \in t} C_{\tau(x)\tau(t)}\end{aligned}$$

où $\tau(x)$ est la vraie classe de l'observation x.

Dans le cas d'une matrice de coût 0-1, le risque empirique d'une feuille est le taux de mauvais classement dans la feuille :

$$\hat{r}(t) = \frac{1}{n_t} \sum_{x \in t} \mathbb{1}_{\tau(t) \neq \tau(x)}$$

Dans le cas d'une matrice de coût quelconque, le risque empirique d'un arbre est le coût moyen de mauvais classement dans l'arbre :

$$\begin{aligned}\hat{R}(T) &= \sum_{t \in \tilde{T}} \frac{n_t}{n} \frac{1}{n_t} \sum_{x \in t} C_{\tau(x)\tau(t)} \\ &= \frac{1}{n} \sum_{t \in \tilde{T}} \sum_{x \in t} C_{\tau(x)\tau(t)}\end{aligned}$$

$n\hat{R}(T)$ est appellé le coût de mauvais classement de T.

Dans le cas d'une matrice de coût 0-1, le risque empirique d'un arbre est le taux de mauvais classement de l'arbre :

$$\hat{R}(T) = \frac{1}{n} \sum_{t \in \tilde{T}} \sum_{x \in t} \mathbb{1}_{\tau(t) \neq \tau(x)}$$

$n\hat{R}(T)$ est alors le nombre de mal classées de T.

Elaguage

La procédure d'élagage (pruning) permet d'éviter le sur-apprentissage tout en permettant d'obtenir un modèle plus parcimonieux.

Cette procédure consiste à :

- constuire une suite de sous-arbres emboîtés à partir de l'arbre maximale construit avec les données d'apprentissage,
- choisir le sous-arbre optimal au sens du critère mesurant un compromis entre la taille de l'arbre et son coût de mauvais classement.

Le critère de coût-complexité $C_\alpha(T)$ est coût de mauvais classement de T pénalisé par la complexité de l'arbre :

$$C_\alpha(T) = n\hat{R}(T) + \alpha|\tilde{T}|,$$

où

- $|\tilde{T}|$ est le nombre de noeuds terminaux de T,
- $n\hat{R}(T)$ est le nombre de mal classés (pour une fonction de coût 0-1) et le coût de mauvais classement (pour une fonction de coût quelconque).

Pour une valeur fixée du paramètre α, on voudra minimiser $C_\alpha(T)$.

Construction de la suite d'arbres emboités.

Pour $\alpha = 0$, $C_\alpha(T) = n\hat{R}(T)$ est minimum pour l'abre maximal. On notera T_L l'arbre maximal à L feuilles et $\alpha_L = 0$.

Pour α qui augmente, l'arbre maximal T_L minimise C_α jusqu'à ce qu'une des divisions de T_L soit superflue et que les deux feuilles de cette division soient regroupées (élaguées) : T_L devient alors T_{L-1}.

Plus précisément aucune division n'est superflue tant que $C_\alpha(T_L) < C_\alpha(T_{L-1})$ soit :

$$n\hat{R}(T_L) + \alpha L < n\hat{R}(T_{L-1}) + \alpha(L-1),$$

donc tant que

$$\alpha < n\hat{R}(T_{L-1}) - n\hat{R}(T_L),$$

On élague donc la division pour laquelle $nR(T_{L-1}) - nR(T_L)$ est minimum afin d'obtenir l'arbre T_{L-1}.

On pose alors :

$$\alpha_{L-1} = n\hat{R}(T_{L-1}) - n\hat{R}(T_L)$$

Pour toute valeur $\alpha \in [0, \alpha_{L-1}[$ c'est donc T_L qui minimise $C_\alpha(T)$.

Le procédé est itéré pour obtenir la séquence d'arbres emboités suivante :

$$T_{max} = T_L \supset T_{L-1} \supset \ldots \supset T_1$$

où T_1 est l'arbre réduit au noeud racine qui contient toutes les données.

Les arbres de cette séquence minimisent $C_\alpha(T)$ sur les plages de valeurs de α suivantes :

$$\alpha_L = 0 < \alpha_{L-1} < \ldots < \alpha_1.$$

Soit :

$$\begin{aligned}
&[0, \alpha_{L-1}[\rightarrow T_L \\
&[\alpha_{L-1}, \alpha_{L-2}[\rightarrow T_{L-1} \\
&\vdots \\
&[\alpha_2, \alpha_1[\rightarrow T_2 \\
&[\alpha_1, \infty[\rightarrow T_1
\end{aligned}$$

Les valeurs $\alpha_1 \ldots \alpha_j \ldots \alpha_L$ de cette séquence sont appellées les paramètres de complexités et mesurent la diminution du coût de mauvais classement obtenu en élaguant T_{j+1} pour obtenir T_j :

$$\alpha_j = n\hat{R}(T_j) - n\hat{R}(T_{j+1})$$

Paramètre de complexité associé à un noeud t.

Soit t la feuille de T_j qui a été divisé pour obtenir T_{j+1}. En notant (t_L, t_R) la division de t on a :

$$\alpha_j = n\hat{r}(t) - n\hat{r}(t_L) - n\hat{r}(t_R).$$

Le paramètre de compléxité de t est alors :

$$cp(t) = LOSS(t) - LOSS(t_L) - LOSS(t_R)$$

où

$$LOSS(t) = n\hat{r}(t) = \sum_{x \in t} C_{\tau(x)\tau(t)}$$

est simplement le coût de mauvais classement (ou le nombre de mauvais classement pour des coûts 0-1) dans t. Cette fonction $LOSS$ est utilisée dans les sorties de la fonction `rpart` du package R du même nom.

Algorithme de construction.

Pour constuire la séquence de sous-arbres emboités, il suffit de trier par ordre croissant les noeuds de l'arbre maximal T_{max} en fonction de leur paramètre de complexité, puis de supprimer successivement les divisions associées à ces noeuds.

Remarque.

Il existe différentes implémentations du paramètre de complexité. Par exemple dans la fonction `rpart` du logiciel R :

$$cp(t) = \frac{LOSS(t) - LOSS(t_L) - LOSS(t_R)}{LOSS(t_1)}$$

avec t_1 le noeud racine.

Le critère minimisé est alors :

$$C_\alpha(T) = R(T) + \alpha|\tilde{T}|R(T_1).$$

Dans la fonction `rpart` ce paramètre est noté *cp* (complexity parameter).

Choix du sous-arbre optimal

Il s'agit maintenant de choisir un arbre optimal dans la séquence $T_1 \supset \ldots \supset T_L$ des sous arbres constuits avec les donnée d'apprentissage. Pour cela les risques empiriques $\hat{R}(T_j)$ de tous les arbres T_j de cette séquence sont calculés sur les données test. On peut alors :

- représenter la décroissance ou éboulis du risque en fonction du nombre croissant de feuilles dans l'arbre ou, de manière équivalente, en fonction de la valeur décroissante du paramètre de complexité α du sous-arbre.
- choisir le nombre de feuilles du sous-arbre qui minimise $\hat{R}(T)$.

Si on veut effectuer plusieurs découpages apprentissage-test ou encore faire de la validation croisée, les séquences de sous-arbres seront différentes sur les différents échantillons d'apprentissage.

Comment sélectionner un sous arbre en validation croisée ?

Pour choisir le sous-arbre par validation croisée, la fonction `rpart` procède de la manière suivante :

1. A partir des coefficients de complexités $\alpha_1, \ldots, \alpha_{L-1}$ calculer :

$$\begin{aligned}
\beta_L &= 0 \\
\beta_{L-1} &= \sqrt{\alpha_{L-1}\alpha_{L-2}} \\
&\vdots \\
\beta_2 &= \sqrt{\alpha_2\alpha_1} \\
\beta_1 &= \infty
\end{aligned}$$

   Chaque coefficient β_j est "typique" de l'intervalle $[\alpha_j, \alpha_{j-1}[$.

2. Diviser les données en I groupes $G_1, \ldots, G_I$ de même taille et pour chaque groupe G_i :

   - construire l'arbre maximum T_{max}^{-i} à partir des données privées du groupe G_i et déterminer pour $j = 1, \ldots, L$ les sous-arbres T_j^{-i} et les intervalles $[\alpha_j^{-i}, \alpha_{j-1}^{-i}[$ associés. Pour chaque β_j, retenir alors le sous-arbres T_j^{-i} associé à l'intervalle $[\alpha_j^{-i}, \alpha_{j-1}^{-i}[$ qui contient β_j.
   - prédire pour chaque sous-arbre T_j^{-i} associé à une valeur β_j la classe des observation du groupe G_i.
   - calculer le coût de mauvais classement $C_{\tau(t)\tau(x)}$ pour chaque observation de G_i et sommer ces coûts de mauvais classement.

3. Pour chaque β_j calculer la moyenne (appelée `xerror` dans `rpart`) et l'écart-type (appelé `xstd` dans `rpart`) des I coût (ou taux) de mauvais classement obtenus en prédisant chaque groupe G_i.

4. Sélectionner β_j qui donne le coût de validation croisée minimum. Elaguer les branches de T_{max} qui partent d'un noeud ayant un coeffient de complexité inférieur ou égal au coefficient α_j correspondant.

La règle du 1-SE (Standard-Error) consiste à retenir parmi tous les β_j ceux qui ont une erreur de validation croisée à moins d'un écart-type de l'erreur minimum. L'écart-type utilisé est celui associé à cette erreur de validation croisée minimum. Ensuite, on retient parmi tous les arbres associés à ces β_j le plus simple (avec le moins de feuilles).