Classification tree for grouped variables

Audrey Poterie[1]

Joint work with **Jean-François Dupuy**[1], **Valérie Monbet**[2] and **Laurent Rouvière**[3]

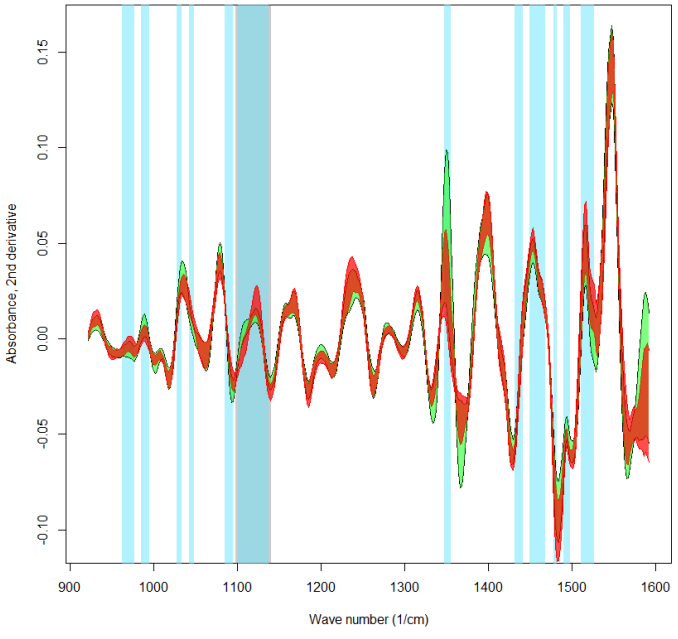
Septième rencontre des jeunes statisticiens

April 3-7 2017

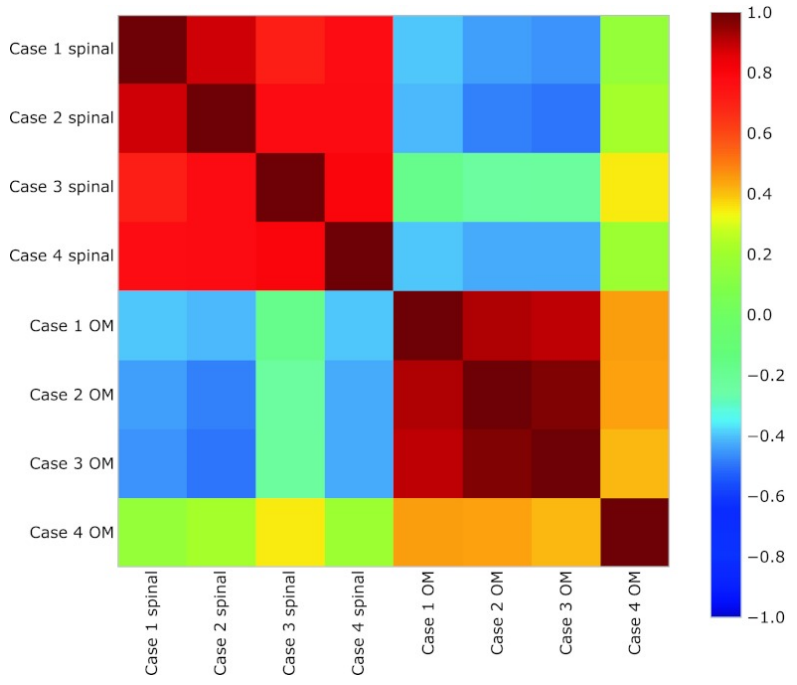
[1] INSA-IRMAR, Rennes
[2] Université de Rennes 1,IRMAR, Rennes
[3] Université de Rennes 2,IRMAR, Rennes

Introduction

Spectrometry data



Gene expression data



Elaboration of classification rule based on the groups of inputs

$\Rightarrow$ Logistic Group-LASSO (Meier, 2008).

$\Rightarrow$ Introduction of the Tree Linear-Discriminant-Analysis algorithm (TLDA)

Introduction

- Let $(\mathbf{X}, Y)$ be a random vector taking values in $\mathbb{R}^p \times \{0, 1\}$
- $\mathbf{X}$ structured into J known groups: let $\mathbf{X}^j = (X_{G_j(1)}, \ldots, X_{G_j(p_j)})^\top$ be the group j
  - p_j: the cardinality of $\mathbf{X}^j$
  - $G_j = \{G_j(1), \ldots, G_j(p_j)\}$: the set of indices of the components of $\mathbf{X}$ belonging to $\mathbf{X}^j$
- Let $(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)$ and $(\mathbf{X}_{n+1}, Y_{n+1}), \ldots, (\mathbf{X}_{n+m}, Y_{n+m})$ be training and validation samples respectively.

Objective

Elaboration of a discrimination rule based on the groups of inputs:

$$\begin{array}{rccc} g: & \mathbb{R}^p & \to & \{0, 1\} \\ & (\mathbf{x}^1, \ldots, \mathbf{x}^J) & \mapsto & y \end{array}$$

Reminder of the Classification Tree algorithms

- Non parametric classification algorithms.
- No constraint on inputs (nature, number).
- Existence of lots of algorithms (CART, CHAID, CRUISE, HHCART...)

Objective

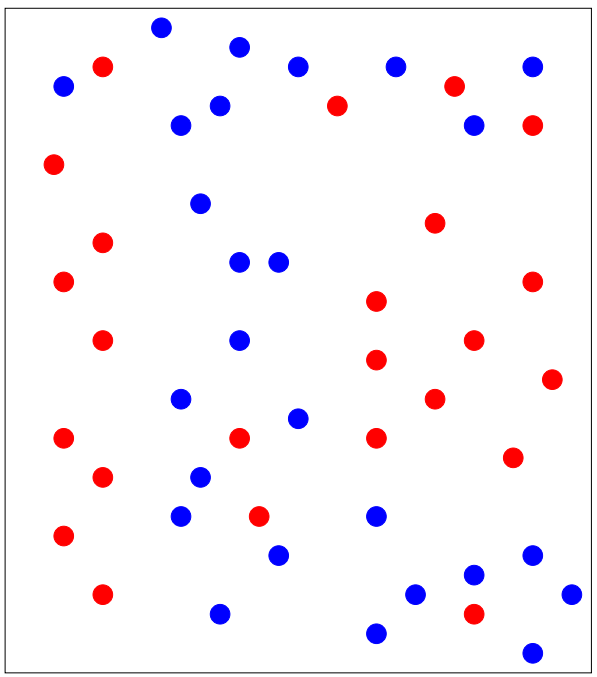
Elaboration of a discrimination rule by means of recursive splits[1] of the data space $\mathbb{R}^p$.

1. we restrict our attention to binary splits

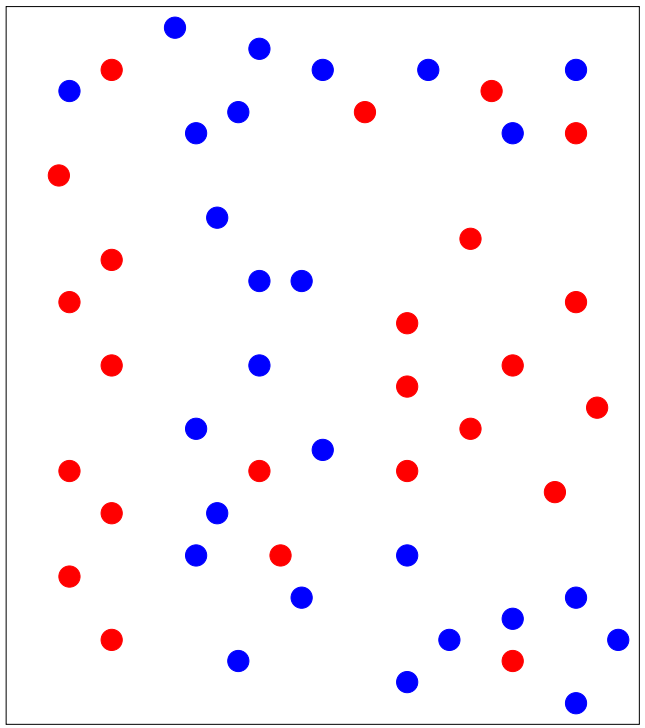
Reminder of the Classification Tree algorithms

Objective

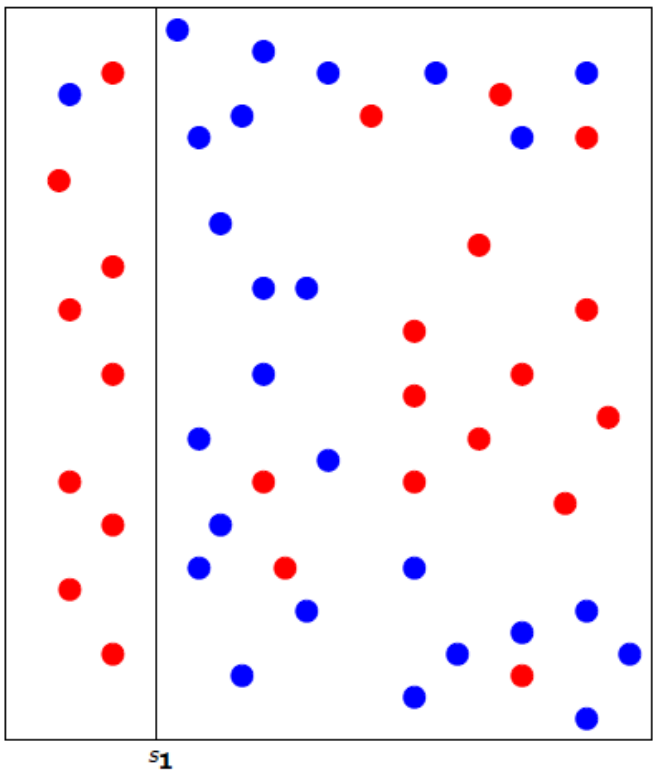
Find a partition of $\mathbb{R}^p$ which best separates the two sets $\{i \mid y_i = 0\}$ and $\{i \mid y_i = 1\}$.



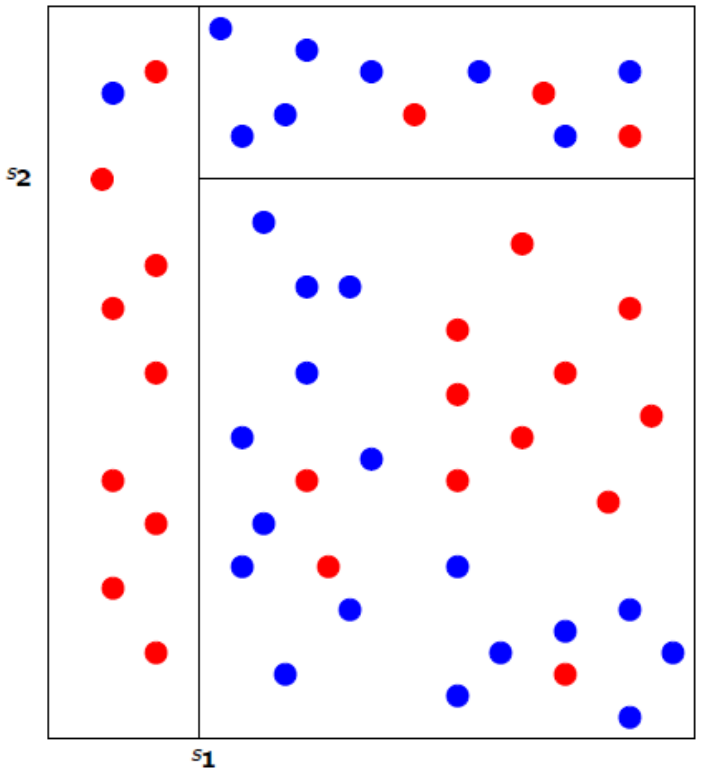
- At each step, definition of a new split $\Leftrightarrow$ simultaneous selection of a splitting variable l and a splitting point s.



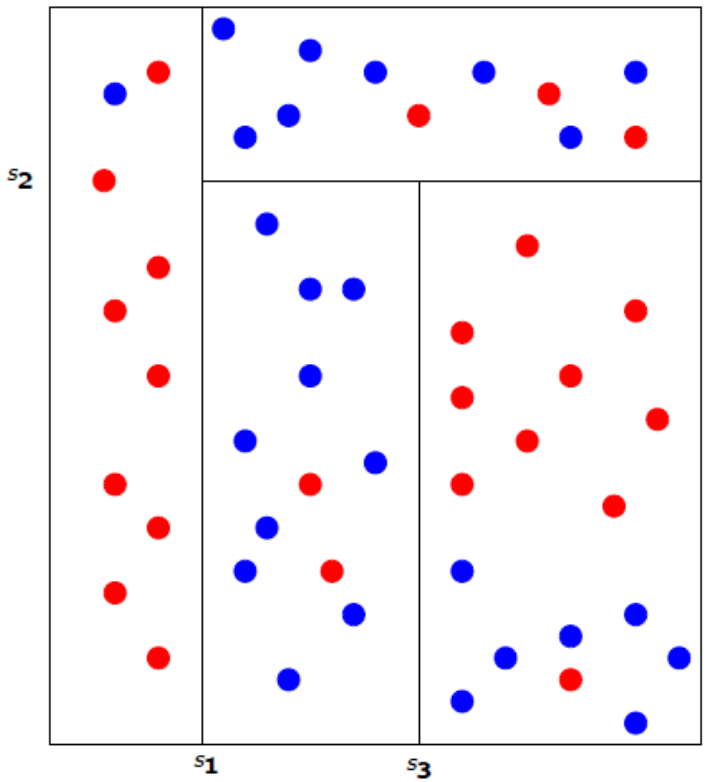
- At each step, definition of a new split ⇔ simultaneous selection of a splitting variable l and a splitting point s.



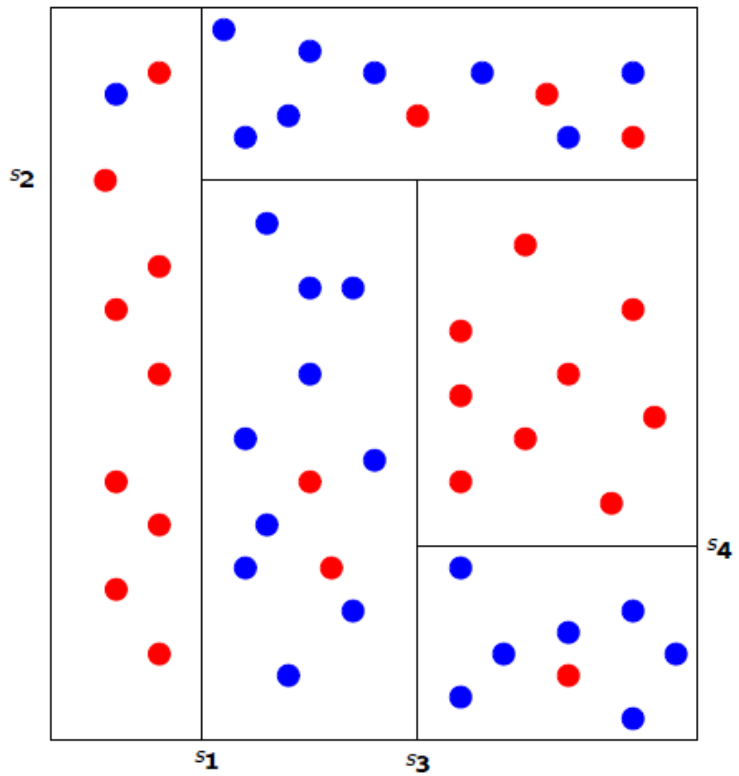
- At each step, definition of a new split ⇔ simultaneous selection of a splitting variable l and a splitting point s.

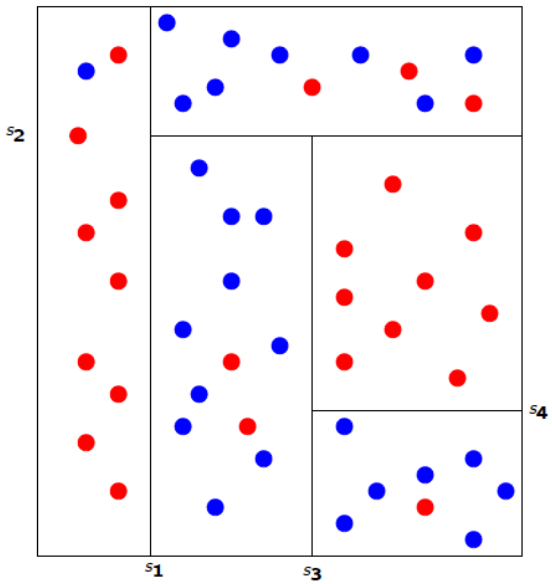


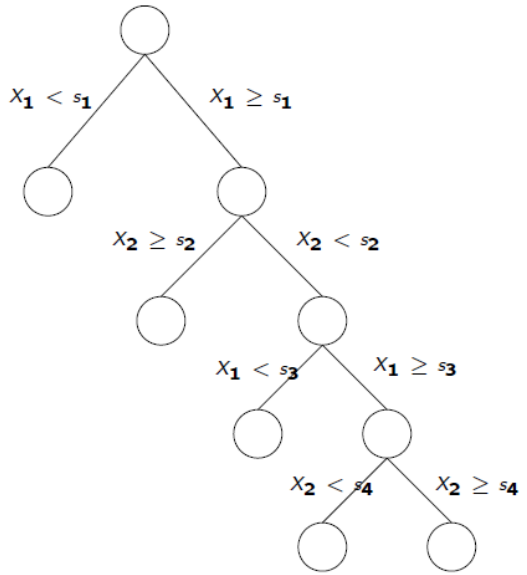
- At each step, definition of a new split ⇔ simultaneous selection of a splitting variable l and a splitting point s.

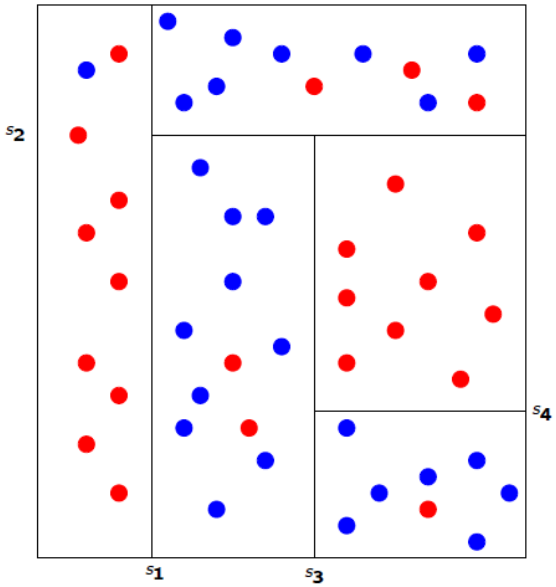


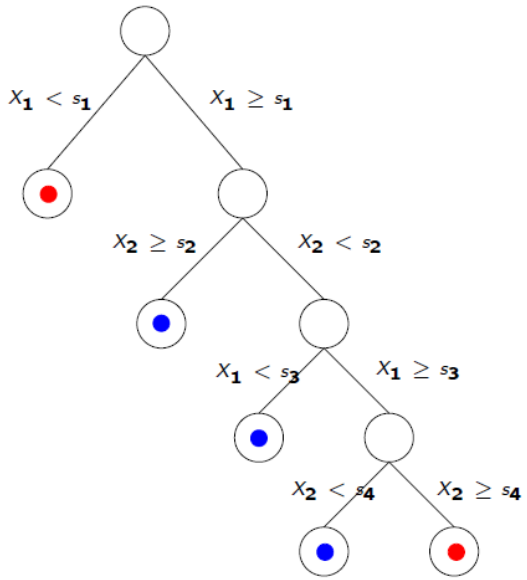
- At each step, definition of a new split ⇔ simultaneous selection of a splitting variable l and a splitting point s.




s_2
s_1
s_3
s_4


$X_1 < s_1$
$X_1 \geq s_1$
$X_2 \geq s_2$
$X_2 < s_2$
$X_1 < s_3$
$X_1 \geq s_3$
$X_2 < s_4$
$X_2 \geq s_4$





Definition of a discrimination rule

The partition defines the discrimination rule g

$$g(\mathbf{x}) = \begin{cases} 1 & \text{if } \#(\{i : Y_i = 1 \text{ and } \mathbf{X}_i \in \mathrm{N}(\mathbf{x})\}) \geq \#(\{i : Y_i = 0 \text{ and } \mathbf{X}_i \in \mathrm{N}(\mathbf{x})\}) \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathrm{N}(\mathbf{x})$ stands for the terminal node which contains $\mathbf{x}$.

The splitting process

- The split of a node N defined according to a splitting variable l and a splitting point s:

$$N_1(l, s) = \{\mathbf{X} \in N \mid X_l \leq s\} \quad \textit{and} \quad N_2(l, s) = \{\mathbf{X} \in N \mid X_l > s\}.$$

- Based on the minimisation of an impurity criterion (Information criterion, Gini index).

The splitting process

- The split of a node N defined according to a splitting variable l and a splitting point s:

$$N_1(l, s) = \{\mathbf{X} \in N \mid X_l \leq s\} \quad \textit{and} \quad N_2(l, s) = \{\mathbf{X} \in N \mid X_l > s\}.$$

- Based on the minimisation of an impurity criterion (Information criterion, Gini index).

How do we deal with grouped variables ?

The splitting process

- The split of a node N defined according to a splitting variable l and a splitting point s:

$$N_1(l, s) = \{\mathbf{X} \in N \mid X_l \leq s\} \quad \textit{and} \quad N_2(l, s) = \{\mathbf{X} \in N \mid X_l > s\}.$$

- Based on the minimisation of an impurity criterion (Information criterion, Gini index).

$\Rightarrow$ Introduction of TLDA : a two-stages splitting process to deal with groups

Step 1: Choice of a split for each group

- Perform a linear discriminant analysis (LDA) on each group $\mathbf{X}^j$, (for $j = 1, \ldots, J$).

  $\Rightarrow$ Determination of scores $\hat{\delta}^j_{0,\mathrm{N}}$ and $\hat{\delta}^j_{1,\mathrm{N}}$.

  $\Rightarrow$ A new observation $\mathbf{x} \in \mathbb{R}^p$ is affected to group 1 if $\hat{\delta}^j_{1,\mathrm{N}}(\mathbf{x}) \geq \hat{\delta}^j_{0,\mathrm{N}}(\mathbf{x})$

- Split of N into two child nodes

$$\mathrm{N}_0(j) = \{\mathbf{X} \in \mathrm{N} \mid \hat{\delta}^j_{0,\mathrm{N}}(\mathbf{X}) \geq \hat{\delta}^j_{1,\mathrm{N}}(\mathbf{X})\}$$

and

$$\mathrm{N}_1(j) = \{\mathbf{X} \in \mathrm{N} \mid \hat{\delta}^j_{0,\mathrm{N}}(\mathbf{X}) < \hat{\delta}^j_{1,\mathrm{N}}(\mathbf{X})\}.$$

Remark: To be less sensitive to the group size, use of regularized LDA (Witten, 2011).

$\Rightarrow$ The Tree Penalized Linear Discriminant Analysis algorithm (TPLDA)

Step 2: Choice of the splitting group

- Use of the Gini impurity function defined by

$$\mathcal{I}(\mathrm{N}) = \pi_{1,\mathrm{N}}(1 - \pi_{1,\mathrm{N}})$$

where $\pi_{1,\mathrm{N}} = \mathbf{P}(Y = 1 | \mathbf{X} \in \mathrm{N})$ is estimated on the training sample by

$$\widehat{\mathcal{I}}(\mathrm{N}) = \frac{n_{1,\mathrm{N}}}{n_{\mathrm{N}}}(1 - \frac{n_{1,\mathrm{N}}}{n_{\mathrm{N}}})$$

with $n_{1,\mathrm{N}} = \#(\{i : Y_i = 1 \text{ and } \mathbf{X}_i \in \mathrm{N}\})$.

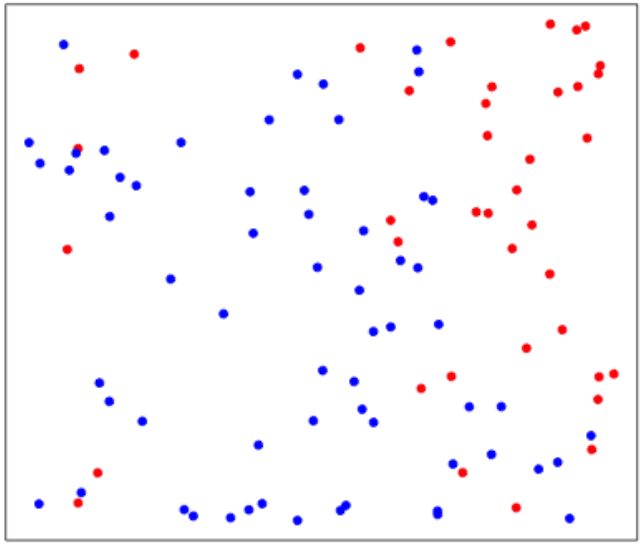
$\Rightarrow$ Selection of the group which maximizes the impurity decrease

$$\Delta_j(\mathrm{N}) = \widehat{\mathcal{I}}(\mathrm{N}) - \left[\frac{n_{\mathrm{N}_0(j)}}{n_{\mathrm{N}}} \hat{\mathcal{I}}(\mathrm{N}_0(j)) + \frac{n_{\mathrm{N}_1(j)}}{n_{\mathrm{N}}} \hat{\mathcal{I}}(\mathrm{N}_1(j)) \right].$$

Example with a unique group

- $Y \in \{0, 1\}$, $\mathbf{X} = (X_1, X_2) \in [0; 1]^2$ structured as a unique group $\mathbf{X}^1$
- Training sample: $D_n = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n))$

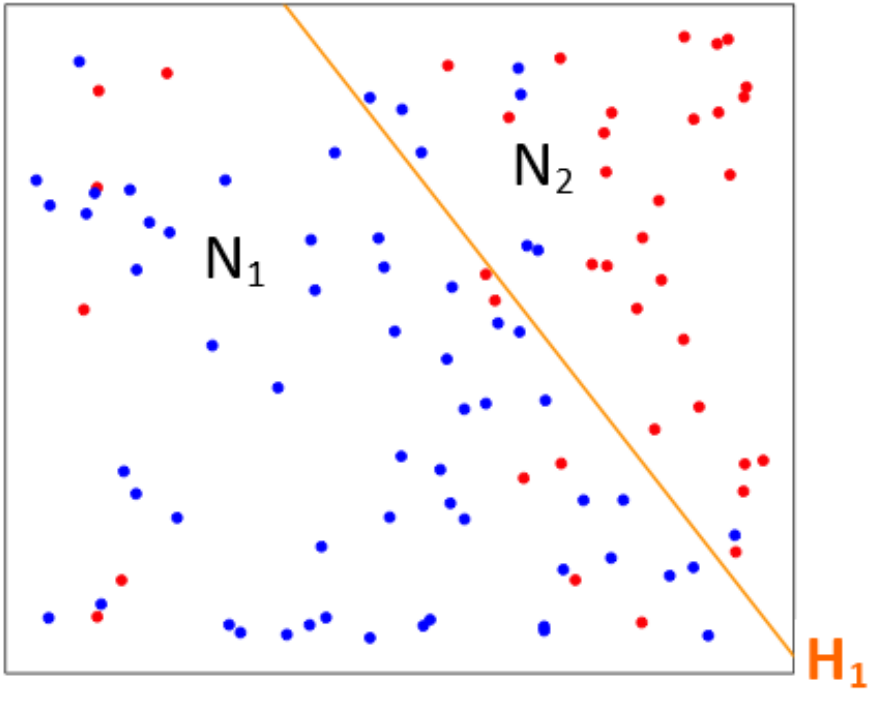
$\Rightarrow$ To predict the class Y based on $\mathbf{X}^1$ and D_n



Example with a unique group

- Definition of a first split by performing a LDA on $\mathbf{X} = (X_1, X_2)$ and D_n:

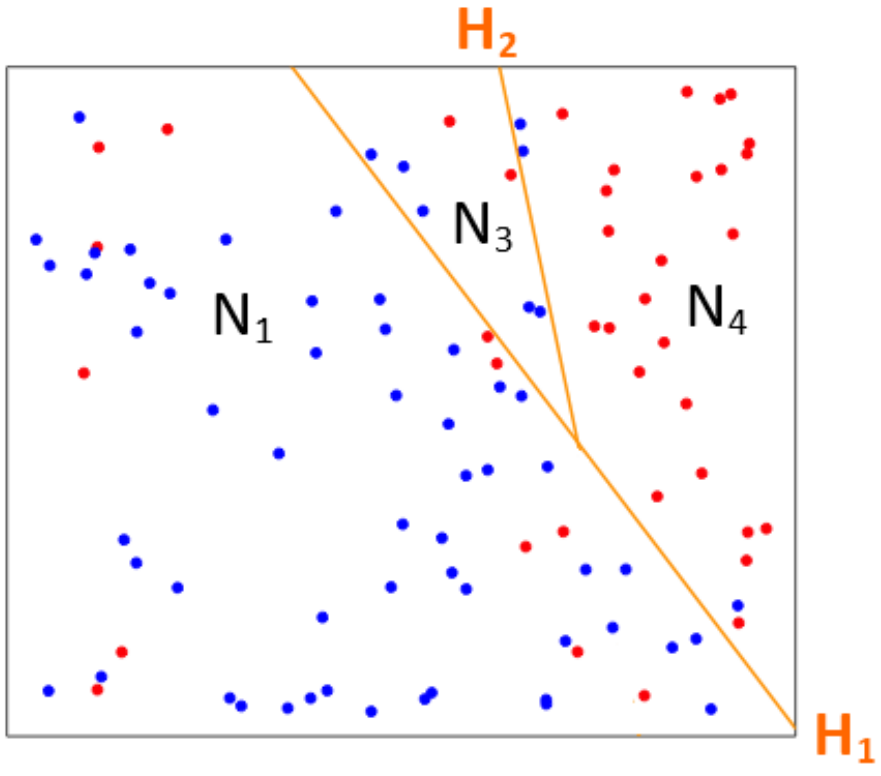
$$\Rightarrow H_1 = \{(x_1, x_2) \in [0;1]^2 | \ \hat{\delta}^1_{0,D_n}(\mathbf{x}) = \hat{\delta}^1_{1,D_n}(\mathbf{x})\}$$



Example with a unique group

- Repetition of the splitting process

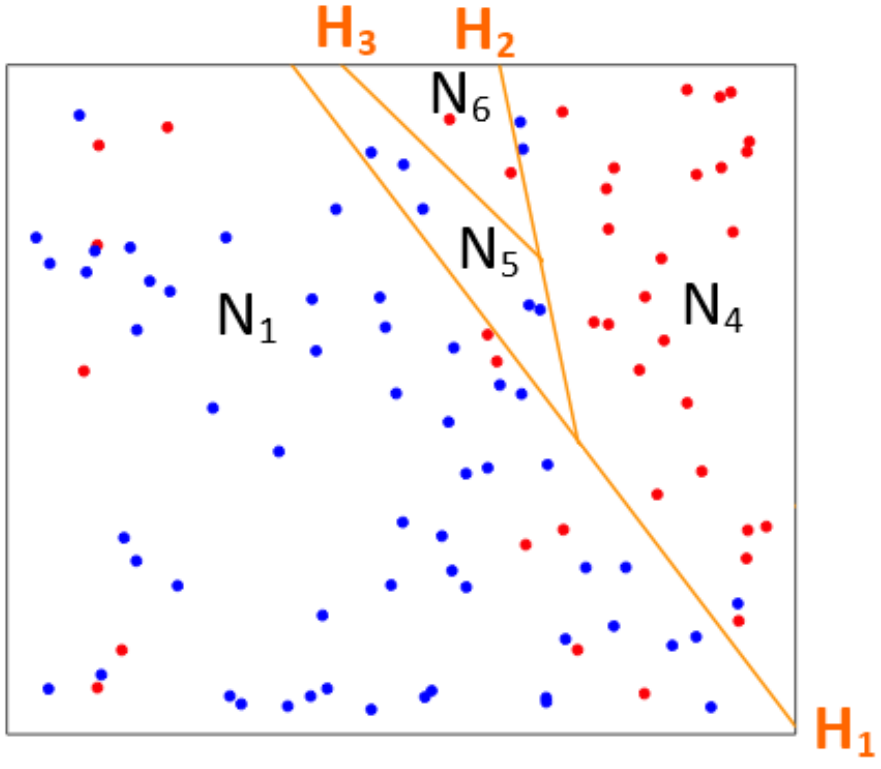
$\Rightarrow$ a second split: $H_2 = \{(x_1, x_2) \in N_2 |\ \hat{\delta}^1_{0,N_2}(\mathbf{x}) = \hat{\delta}^1_{1,N_2}(\mathbf{x})\}$



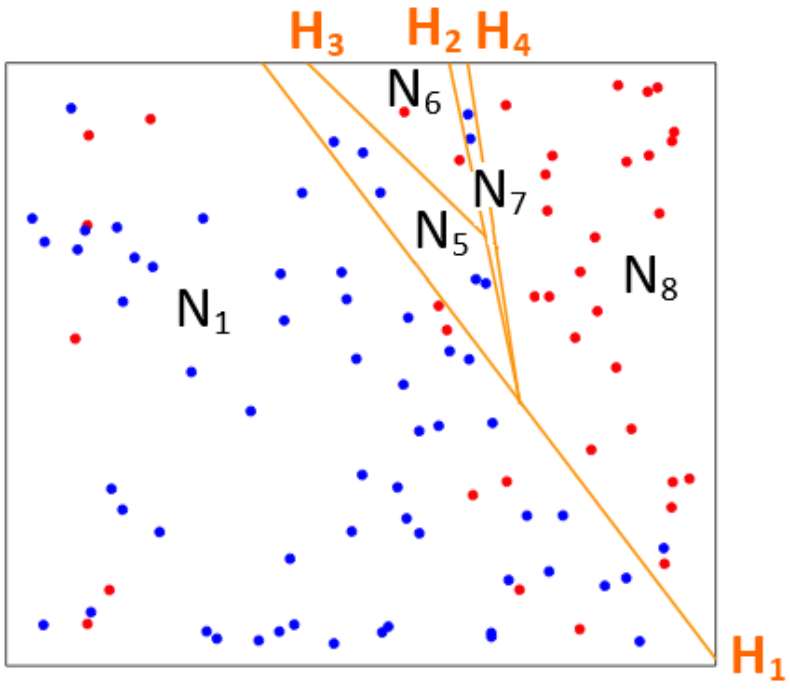
Example with a unique group

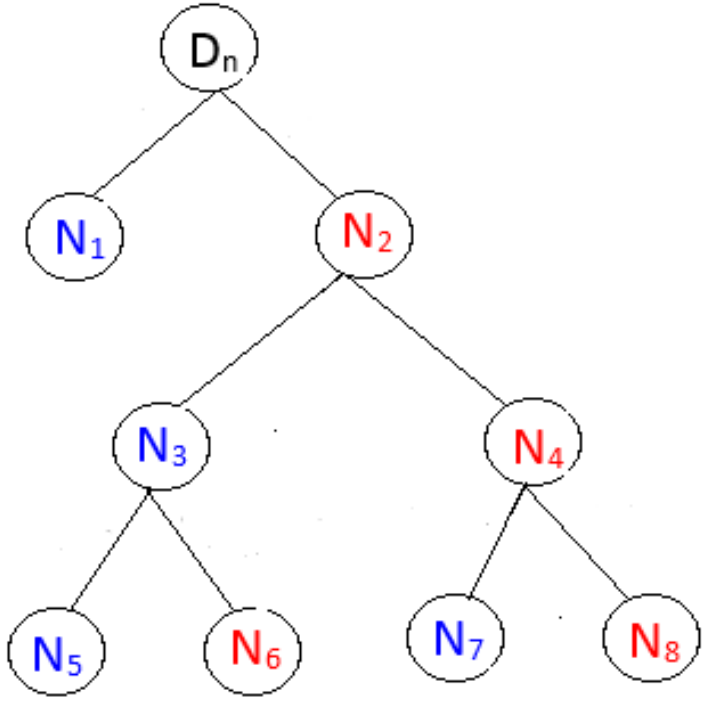
- Repetition of the splitting process

$\Rightarrow$ a third split: $H_3 = \{(x_1, x_2) \in N_5 | \ \hat{\delta}^1_{0,N_5}(\mathbf{x}) = \hat{\delta}^1_{1,N_5}(\mathbf{x})\}$



Example with a unique group





$\Rightarrow$ Elaboration of a tree T of depth $D(T) = 3$.

When does the splitting process stop ?

- A too small tree suffers from bias
- A too large tree tends to overfit the data

The proposed strategy

1. To built a fully grown tree T_{max}
2. To prune T_{max}

$\Rightarrow$ Introduction of a new pruning approach

1. Elaboration of a maximal tree T_{max}

- Every sub-region N is homogeneous (or near so) with respect to a particular class i.e.

$$\frac{n_{1,\mathrm{N}}}{n_{\mathrm{N}}} < \epsilon \quad \text{or} \quad \frac{n_{1,\mathrm{N}}}{n_{\mathrm{N}}} > 1 - \epsilon$$

  for a chosen ϵ value;

- For every sub-region N, no further partition can reduce the impurity, which is equivalent to

$$\Delta_j(\mathrm{N}) \leq 0, \ \forall\, j = 1, \ldots, J.$$

Pruning of T_{max}: choice of the maximal tree

Notations:

- $|T|$ = the number of nodes of a tree T
- $\{\mathrm{N}_1(T), \ldots, \mathrm{N}_{|T|}(T)\}$ = the nodes of T
- $d(\mathrm{N}_\ell(T))$ = the depth of the node $(\mathrm{N}_\ell(T))$
- $D(T_{max})$ = the depth of T_{max}

$\Rightarrow$ Definition of the nested sequence

$$(T_k)_k = \{T_0 \subset \ldots \subset T_{D(T_{max})} = T_{max}\}$$

where T_k maximizes over all trees $T \subset T_{max}$ the quantity

$$\sum_{\ell=1,\ldots,|T|} d(\mathrm{N}_\ell(T)) \quad \text{subject to} \quad d(\mathrm{N}_\ell(T)) \leq k.$$

Pruning of T_{max}: choice of the maximal tree

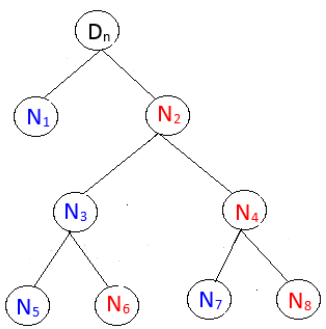
$\Rightarrow$ Definition of the nested sequence

$$(T_k)_k = \{T_0 \subset \ldots \subset T_{D(T_{max})} = T_{max}\}$$

where T_k maximizes over all trees $T \subset T_{max}$ the quantity

$$\sum_{\ell=1,\ldots,|T|} d(\mathrm{N}_\ell(T)) \quad \text{subject to} \quad d(\mathrm{N}_\ell(T)) \leq k.$$

Example



Tree	Terminal Nodes
T_0	D_n
T_1	$\mathrm{N}_1, \mathrm{N}_2$
T_2	$\mathrm{N}_1, \mathrm{N}_3, \mathrm{N}_4$
T_3	$\mathrm{N}_1, \mathrm{N}_5, \mathrm{N}_6, \mathrm{N}_7, \mathrm{N}_8$

Pruning of T_{max}: choice of the maximal tree

- Definition of the associated sequence $(g_k)_k = \{g_0, g_1, \ldots, g_{D(T_{max})}\}$.

Choice of the final tree $T_{\widehat{K}}$

To select the classification rule which minimizes $\mathbf{P}(g_k(\mathbf{X}) \neq Y)$

$\Rightarrow$ $\mathbf{P}(g_k(\mathbf{X}) \neq Y)$ estimated on the validation set:

$$\widehat{\mathbf{P}}(g_k(\mathbf{X}) \neq Y) = \frac{1}{m} \sum_{i=n+1}^{n+m} 1_{g_k(\mathbf{X}_i) \neq Y_i}.$$

$\Rightarrow$ The final tree is $T_{\widehat{K}}$ such that

$$\widehat{K} \in \underset{k=1,\ldots,D_{max}}{\operatorname{argmin}} \widehat{\mathbf{P}}(g_k(\mathbf{X}) \neq Y).$$

Simulation design (Gregorutti, 2013)

- Let $Y \sim \mathcal{B}(0.5)$ and $\mathbf{X} = (\mathbf{X}^1, \ldots, \mathbf{X}^{12})$.
- Let $\mathbf{X}^j = (X_{G_j(1)}, \ldots, X_{G_j(p_j)})^\top$ be the group j such that

$$X_{G_j(k)} \sim \mathcal{N}(Y\mu_j, 1), \ k = 1, \ldots, p_j$$

  with μ_j: the discriminative power of every input in group j

- Within-group correlation: $\mathrm{Cov}\left(X_{G_j(k)}, X_{G_j(l)}\right) = c_w^{|l-k|}$
- Between-group correlation: $\mathrm{Cov}\left(X_{G_j(k)}, X_{G_{j'}(l')}\right) = c_b^{|G_j(k) - G_{j'}(l')|}$

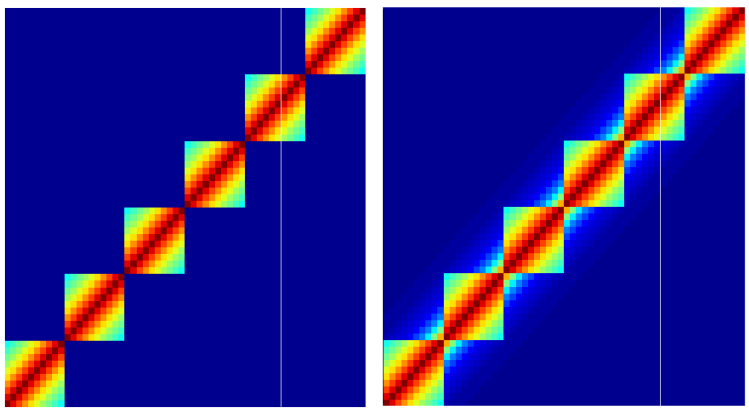


Figure: Correlation structure for 6 groups.(Left panel: $c_w = 0.9$,$c_b = 0$. Right panel: $c_w = 0.9$,$c_b = 0.7$.)

- N generated observations divided into: a training sample, a validation sample and a test sample.
- Fixed parameters: $\mu = (0.5, 0, 0.45, 0, 0.4, 0, 0.35, 0, 0.3, 0, 0.25, 0)$ and $N_{test} = 1000$.
- 500 runs for each scenario.

Scenario	$\mathbf{n}$[4]	$\mathbf{p_j},\ \mathbf{j = 1, , 12}$
1	500	$p_j = 1, \forall j$
2	500	$p_j = 5, \forall j$
3	500	$p_j = 10, \forall j$
4	100	$p_j = 50, \forall j$
5	100	$p_j = \begin{cases} 10 & \text{if } j \equiv 1 \pmod 2; \\ 50 & \text{otherwise.} \end{cases}$
6*[5]	500	$p_j = 10, \forall j$
7*[5]	100	$p_j = 50, \forall j$

$\Rightarrow$ Comparison of the algorithms with CART and Group-LASSO.

[4] The size of the training and the validation samples.
[5] Inputs are correlated ($c_w = 0.8$,$c_b = 0.7$).

- N generated observations divided into: a training sample, a validation sample and a test sample.
- Fixed parameters: $\mu = (0.5, 0, 0.45, 0, 0.4, 0, 0.35, 0, 0.3, 0, 0.25, 0)$ and $N_{test} = 1000$.
- 500 runs for each scenario.

Scenario	$\mathbf{n}$[4]	$\mathbf{p_j},\ \mathbf{j = 1, , 12}$
1	500	$p_j = 1, \forall j$
2	500	$p_j = 5, \forall j$
3	500	$p_j = 10, \forall j$
4	100	$p_j = 50, \forall j$
5	100	$p_j = \begin{cases} 10 & \text{if } j \equiv 1 \pmod 2; \\ 50 & \text{otherwise.} \end{cases}$
6*[5]	500	$p_j = 10, \forall j$
7*[5]	100	$p_j = 50, \forall j$

$\Rightarrow$ Comparison of the algorithms with CART and Group-LASSO.

[4] The size of the training and the validation samples.
[5] Inputs are correlated ($c_w = 0.8$,$c_b = 0.7$).

- N generated observations divided into: a training sample, a validation sample and a test sample.
- Fixed parameters: $\mu = (0.5, 0, 0.45, 0, 0.4, 0, 0.35, 0, 0.3, 0, 0.25, 0)$ and $N_{test} = 1000$.
- 500 runs for each scenario.

Scenario	**n**[4]	$\mathbf{p_j},\ \mathbf{j = 1, , 12}$
1	500	$p_j = 1, \forall j$
2	500	$p_j = 5, \forall j$
3	500	$p_j = 10, \forall j$
4	100	$p_j = 50, \forall j$
5	100	$p_j = \begin{cases} 10 & \text{if } j \equiv 1 \pmod 2; \\ 50 & \text{otherwise.} \end{cases}$
6*[5]	500	$p_j = 10, \forall j$
7*[5]	100	$p_j = 50, \forall j$

$\Rightarrow$ Comparison of the algorithms with CART and Group-LASSO.

[4] The size of the training and the validation samples.
[5] Inputs are correlated ($c_w = 0.8$,$c_b = 0.7$).

- N generated observations divided into: a training sample, a validation sample and a test sample.
- Fixed parameters: $\mu = (0.5, 0, 0.45, 0, 0.4, 0, 0.35, 0, 0.3, 0, 0.25, 0)$ and $N_{test} = 1000$.
- 500 runs for each scenario.

Scenario	$\mathbf{n}$[4]	$\mathbf{p_j},\ \mathbf{j = 1, , 12}$
1	500	$p_j = 1, \forall j$
2	500	$p_j = 5, \forall j$
3	500	$p_j = 10, \forall j$
4	100	$p_j = 50, \forall j$
5	100	$p_j = \begin{cases} 10 & \text{if } j \equiv 1 \pmod 2; \\ 50 & \text{otherwise.} \end{cases}$
6*[5]	500	$p_j = 10, \forall j$
7*[5]	100	$p_j = 50, \forall j$

$\Rightarrow$ Comparison of the algorithms with CART and Group-LASSO.

[4] The size of the training and the validation samples.
[5] Inputs are correlated ($c_w = 0.8$,$c_b = 0.7$).

Predictive performances

	TLDA	TPLDA	CART1[6]	CART2[7]	GL
1	0.61 (0.59,0.62)	0.61 (0.59,0.63)	0.62 (0.6,0.63)	0.62 (0.61,0.64)	0.74 (0.72,0.75)
2	0.74 (0.72,0.76)	0.75 (0.73,0.77)	0.64 (0.62,0.66)	0.65 (0.63,0.66)	0.92 (0.91,0.92)
3	0.82 (0.8,0.84)	0.83 (0.81,0.84)	0.64 (0.63,0.66)	0.65 (0.64,0.67)	0.97 (0.97,0.98)
4	0.87 (0.85,0.89)	0.94 (0.92,0.95)	0.61 (0.59,0.63)	0.62 (0.59,0.64)	0.98 (0.98,0.99)
5	0.51 (0.48,0.55)	0.72 (0.59,0.76)	0.59 (0.56,0.62)	0.59 (0.56,0.62)	0.94 (0.93.0.95)
6	0.63 (0.61,0.65)	0.68 (0.66,0.7)	0.63 (0.61,0.64)	0.63 (0.62,0.65)	0.81 (0.79,0.82)
7	0.59 (0.55,0.62)	0.71 (0.68,0.73)	0.6 (0.58,0.62)	0.61 (0.59,0.62)	0.67 (0.62,0.7)

[6] CART + cost-complexity pruning
[7] CART + proposed pruning strategy

Predictive performances

	TLDA	TPLDA	CART1[6]	CART2[7]	GL
1	0.61 (0.59,0.62)	0.61 (0.59,0.63)	0.62 (0.6,0.63)	0.62 (0.61,0.64)	0.74 (0.72,0.75)
2	0.74 (0.72,0.76)	0.75 (0.73,0.77)	0.64 (0.62,0.66)	0.65 (0.63,0.66)	0.92 (0.91,0.92)
3	0.82 (0.8,0.84)	0.83 (0.81,0.84)	0.64 (0.63,0.66)	0.65 (0.64,0.67)	0.97 (0.97,0.98)
4					
5					
6					
7					

[6] CART + cost-complexity pruning
[7] CART + proposed pruning strategy

Predictive performances

	TLDA	TPLDA	CART1[6]	CART2[7]	GL
1					
2					
3					
4	0.87 (0.85,0.89)	0.94 (0.92,0.95)	0.61 (0.59,0.63)	0.62 (0.59,0.64)	0.98 (0.98,0.99)
5	0.51 (0.48,0.55)	0.72 (0.59,0.76)	0.59 (0.56,0.62)	0.59 (0.56,0.62)	0.94 (0.93.0.95)
6					
7					

[6] CART + cost-complexity pruning
[7] CART + proposed pruning strategy

Predictive performances

	TLDA	TPLDA	CART1[6]	CART2[7]	GL
1					
2					
3	0.82 (0.8,0.84)	0.83 (0.81,0.84)	0.64 (0.63,0.66)	0.65 (0.64,0.67)	0.97 (0.97,0.98)
4	0.87 (0.85,0.89)	0.94 (0.92,0.95)	0.61 (0.59,0.63)	0.62 (0.59,0.64)	0.98 (0.98,0.99)
5					
6	0.63 (0.61,0.65)	0.68 (0.66,0.7)	0.63 (0.61,0.64)	0.63 (0.62,0.65)	0.81 (0.79,0.82)
7	0.59 (0.55,0.62)	0.71 (0.68,0.73)	0.6 (0.58,0.62)	0.61 (0.59,0.62)	0.67 (0.62,0.7)

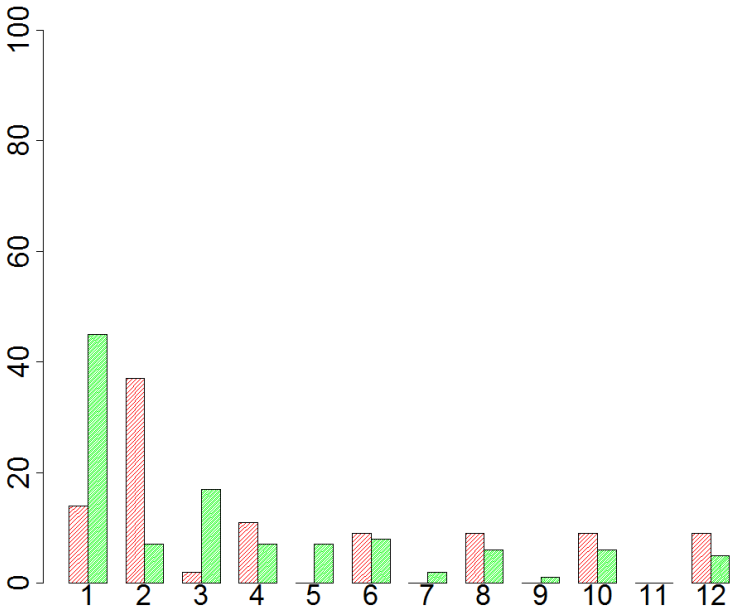
[6] CART + cost-complexity pruning
[7] CART + proposed pruning strategy

Tree complexity

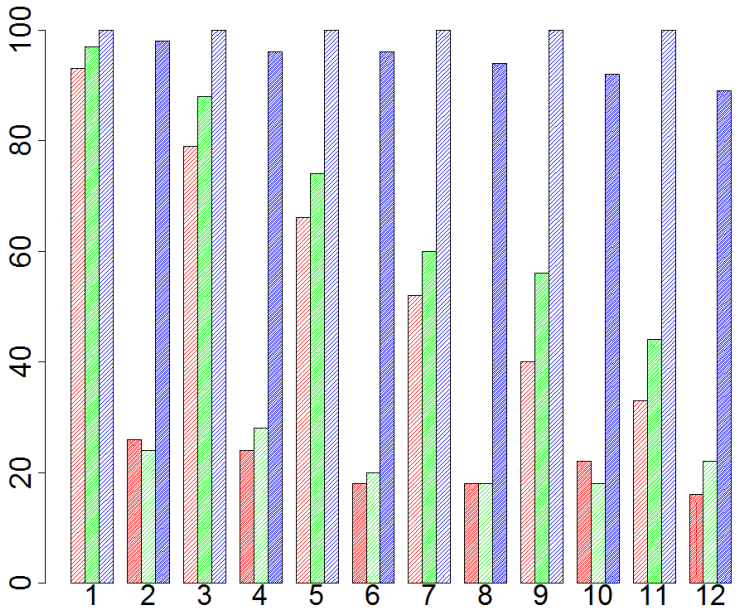
	TLDA	TPLDA	CART1	CART2
1	4 (3,6)	4 (3,6)	6 (4,8)	5 (4,7)
2	3 (3,5)	3 (3,4)	7 (5,8)	6 (5,8)
3	3 (2,4)	3 (3,4)	7 (5,8)	7 (5,8)
4	1 (1,1)	1 (1,1)	4 (3,5)	4 (3,5)
5	1 (0,2)	1 (1,1)	3 (2,5)	3 (2,5)
6	3 (2,4)	3 (3,4)	6 (4,7)	5 (4,7)
7	1 (1,1)	1 (1,3)	4 (3,5)	4 (3,5)

Variable/Group selection selection

Scenario 5
Large noisy group



Scenario 6
Correlated groups of equal size



Algorithms: TLDA, TPLDA, GL

Scenario 5: $n = 100$, $c_w = 0$, $c_b = 0$ *and* $p_j = \begin{cases} 10 & \text{if } j \equiv 1 \ (mod\ 2); \\ 50 & \text{otherwise.} \end{cases}$

Scenario 6: $n = 500$, $c_w = 0.8$, $c_b = 0.7$ *and* $p_j = 10 \ \forall j$

Conclusion:

- Our proposed pruning strategy $\approx$ the cost-complexity pruning
- TPLDA: less sensitive to the group size
- Outperform CART: smaller trees and higher predictive performances
- Best compromise between group selection and prediction

$\Rightarrow$ TLDA and TPDA: two new classification tree algorithms well-adapted to grouped inputs.

Future work:

- Creation of a R package
- Study of other splitting methods
- Extension to random forests

Thank you for your attention

References

Meier, L., Van Der Geer, S., and Buhlman, P. (2008), *The group lasso for logistic regression.* Journal of the Royal Statistical Society: Serie B (Statistical Methodology).

Breiman, L., Friedman, J., Stone, C. J., and Olsen, R. A. (1984). *Classification and regression trees.* CRC press.

Wickramarachi, D., Robertson, B., Reale, M., Price, C., and Brown, J. (2016), *HHCART: An oblique decision tree.* Computational Statistics & Data Analysis.

Bouveyron, C., Girard, S., and Schmid, C. (2007), *High-dimensional discriminant analysis.* Communications in Statistics Theory and Methods.

Witten, D. M., and Tibshirani, R. (2011), *Penalized classification using fisher's linear discriminant.* Journal of the Royal Statistical Society: Serie B (Statistical Methodology).

Gregorutti, B., Michel, B., and Saint-Pierre, P. (2013), *Correlation and variable importance in random forests.* Statistics and Computing.

Step 1: Choice of a split for each group

- Perform a linear discriminant analysis (LDA) on each group $\mathbf{X}^j$,$j = 1, \ldots, J$.

Linear discriminant analysis on $\mathbf{X}^j$

Under the assumption (restricted to the node N) that

$$\{\mathbf{X}^j | Y = k\} \sim \mathcal{N}(\mu^j_{k,\mathrm{N}}, \Sigma^j_{\mathrm{N}}), \quad k = 0, 1$$

the linear discriminant functions $\delta^j_{k,\mathrm{N}}$ are defined by

$$\delta^j_{k,\mathrm{N}}(\mathbf{x}) = \mathbf{x}^{j\top} {\Sigma^j_{\mathrm{N}}}^{-1} \mu^j_{k,\mathrm{N}} - \frac{1}{2} {\mu^j_{k,\mathrm{N}}}^{\top} {\Sigma^j_{\mathrm{N}}}^{-1} \mu^j_{k,\mathrm{N}} + \log \pi_{k,\mathrm{N}} \tag{1}$$

where $\pi_{k,\mathrm{N}} = \mathbf{P}(Y = k | \mathbf{X} \in \mathrm{N})$ and $k = 0, 1$.
A new observation $\mathbf{x} \in \mathbb{R}^p$ is affected to group 1 if $\delta^j_{1,\mathrm{N}}(\mathbf{x}) \geq \delta^j_{0,\mathrm{N}}(\mathbf{x})$.

Parameters are unknown and estimated given the training sequence $(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)$ by their empirical counterpart.