Forêts aléatoires pour variables structurées

Audrey Poterie - Data Science pour les risques côtiers

13-15 novembre 2023 - Station Biologique de Roscoff (France)

Contexte

Classical statistical framework

Supervised learning: we are given a dataset $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$ where the pairs $(\mathbf{X}_i, Y_i)$ are *i.i.d* distributed as $(\mathbf{X}, Y)$ and such that

- $\mathbf{X} = (X_1, \ldots, X_d) \in \mathbb{R}^d$ (*inputs*)
- $Y \in \mathcal{Y}$ (*response variable*) with
  - $\mathcal{Y} = \mathbb{R}$ in regression,
  - $\mathcal{Y} = \{1, \ldots, K\}$ in classification ($k \in \mathbb{N}$).

Classical statistical framework

Supervised learning: we are given a dataset $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$ where the pairs $(\mathbf{X}_i, Y_i)$ are *i.i.d* distributed as $(\mathbf{X}, Y)$ and such that

- ▷ $\mathbf{X} = (X_1, \ldots, X_d) \in \mathbb{R}^d$ (*inputs*)
- ▷ $Y \in \mathcal{Y}$ (*response variable*) with
  - $\mathcal{Y} = \mathbb{R}$ in regression,
  - $\mathcal{Y} = \{1, \ldots, K\}$ in classification ($k \in \mathbb{N}$).

➜ **Question:** for each $\mathbf{x} \in \mathbb{R}^d$, predict the response $\hat{y} \in \mathcal{Y}$ *i.e.* find a predictor

$$\hat{h} : \mathbb{R}^d \to \mathcal{Y},$$

such that $\hat{h}(\mathbf{x}_i) \approx y_i$, for $i = 1, \ldots, n$.

Classical statistical framework

Supervised learning: we are given a dataset $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$ where the pairs $(\mathbf{X}_i, Y_i)$ are *i.i.d* distributed as $(\mathbf{X}, Y)$ and such that

- ▷ $\mathbf{X} = (X_1, \ldots, X_d) \in \mathbb{R}^d$ (*inputs*)
- ▷ $Y \in \mathcal{Y}$ (*response variable*) with
  - $\mathcal{Y} = \mathbb{R}$ in regression,
  - $\mathcal{Y} = \{1, \ldots, K\}$ in classification ($k \in \mathbb{N}$).

➜ **Question:** for each $\mathbf{x} \in \mathbb{R}^d$, predict the response $\hat{y} \in \mathcal{Y}$ *i.e.* find a predictor

$$\hat{h} : \mathbb{R}^d \to \mathcal{Y},$$

such that $\hat{h}(\mathbf{x}_i) \approx y_i$, for $i = 1, \ldots, n$.

➜ **Multiple approaches:** SVM, random forests, boosting, CNN, etc.

Our statistical framework

Further assumption: we assume that

$$\mathbf{X} = (\mathbf{X}^1, \ldots, \mathbf{X}^J)$$

is **structured into J known groups** where the jth group of size d_j is denoted $\mathbf{X}^j = (X_{j_1}, X_{j_2}, \ldots, X_{j_{d_j}})$.

Our statistical framework

Further assumption: we assume that

$$\mathbf{X} = (\mathbf{X}^1, \ldots, \mathbf{X}^J)$$

is **structured into J known groups** where the jth group of size d_j is denoted $\mathbf{X}^j = (X_{j_1}, X_{j_2}, \ldots, X_{j_{d_j}})$.

➜ **Reference approaches:** group lasso methods (group lasso, sparse group lasso, weighted group lasso, etc.)

Our statistical framework

Further assumption: we assume that

$$\mathbf{X} = (\mathbf{X}^1, \ldots, \mathbf{X}^J)$$

is **structured into J known groups** where the jth group of size d_j is denoted $\mathbf{X}^j = (X_{j_1}, X_{j_2}, \ldots, X_{j_{d_j}})$.

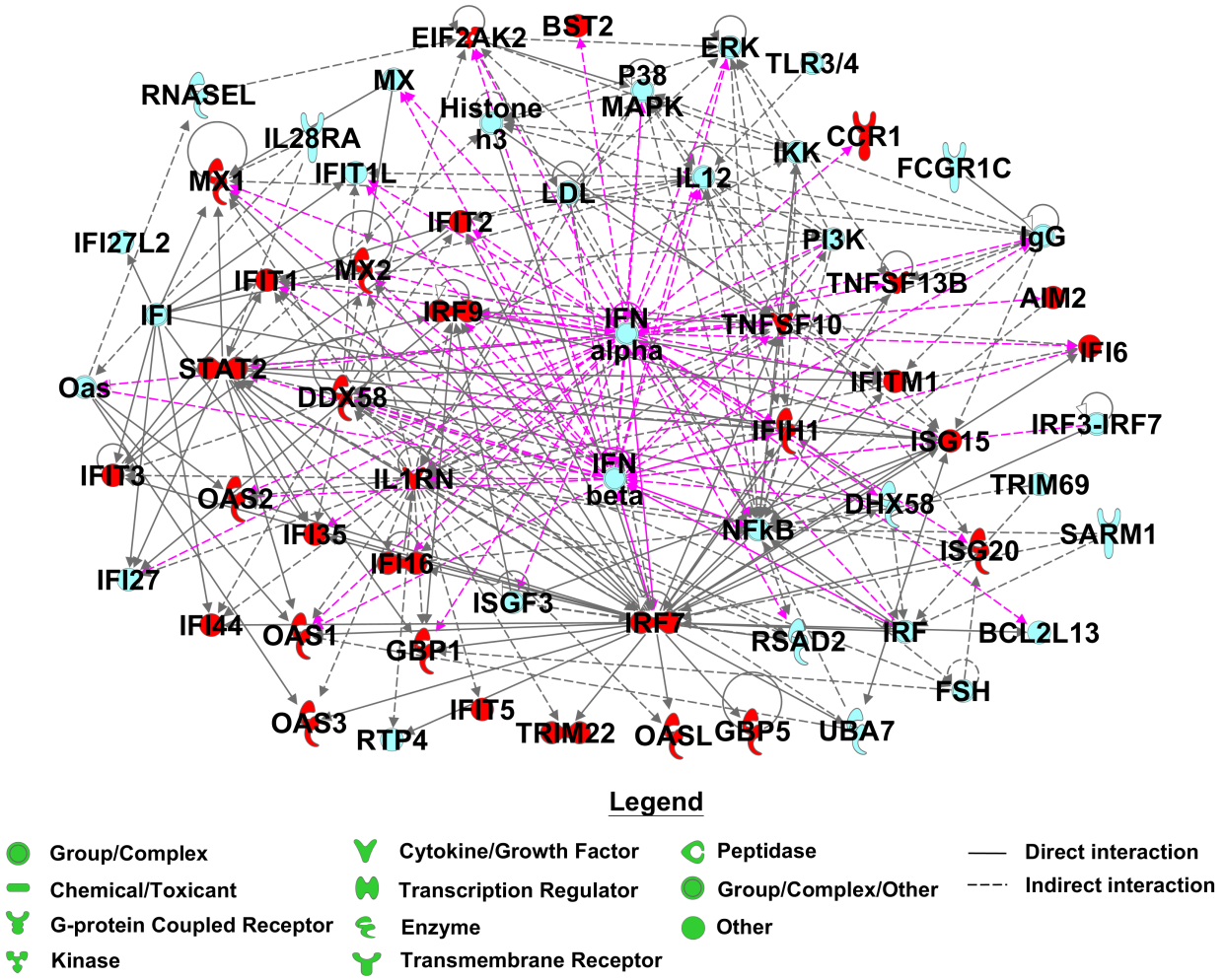
➜ **Reference approaches:** group lasso methods (group lasso, sparse group lasso, weighted group lasso, etc.)

Objective

Develop a random forest method to supervised problems in which $\mathbf{X}$ has a known group structure.

Examples of inputs with a known group structure

Gene expression data:



Reynier *et al.*, (2011). *Importance of correlation between gene expression levels: application to the type I interferon signature in rheumatoid arthritis.* PloS one.

Examples of inputs with a group structure

Spectrometry data:

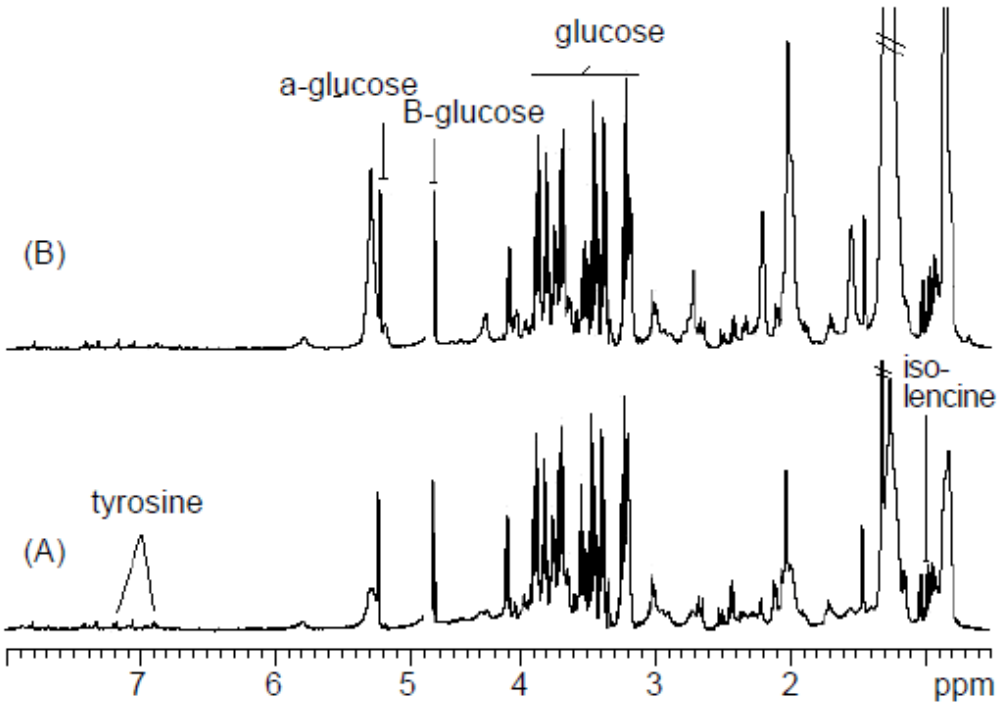


Figure 4: Normal blood glucose group (A) and high blood glucose group (B) typical serum ^{1}H NMR spectrum.

Luabi *et al.*, (2015). *Non invasive blood glucose level measurement using nuclear magnetic resonance.* Proceedings of the 8th IEEE GCC Conference and Exhibition.

Random forests

Random forests

Phil Cutler

Random forests [Breiman, 2001] are a class of algorithms used to solve regression and classification problems

- ▷ Large applicability: used in many applied fields since they handle high-dimensional settings.
- ▷ Successful methods: good predictive power, can outperform state-of-the-art methods.

Random forests

Phil Cutler

Random forests [Breiman, 2001] are a class of algorithms used to solve regression and classification problems

- ▷ Large applicability: used in many applied fields since they handle high-dimensional settings.
- ▷ Successful methods: good predictive power, can outperform state-of-the-art methods.

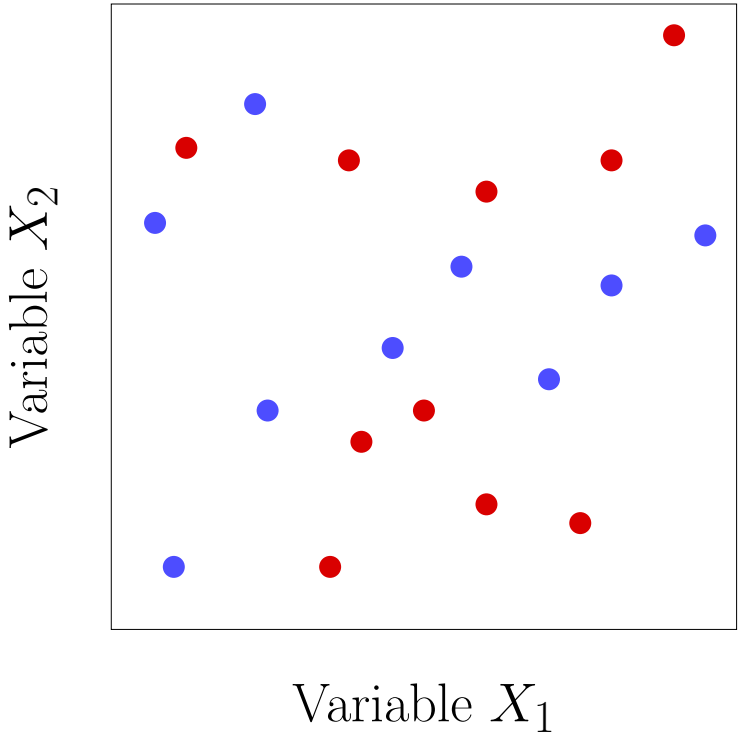
➜ **A random forest = aggregation of many *random* decision trees.**

How to build a CART tree ?

- **CART** [Breiman et al., 1984]: non parametric learning algorithm.
- Idea: a tree is built recursively by recursively splitting the inoput space $\mathbb{R}^d$. into two disjoint regions until some stopping criterion is satisfied.

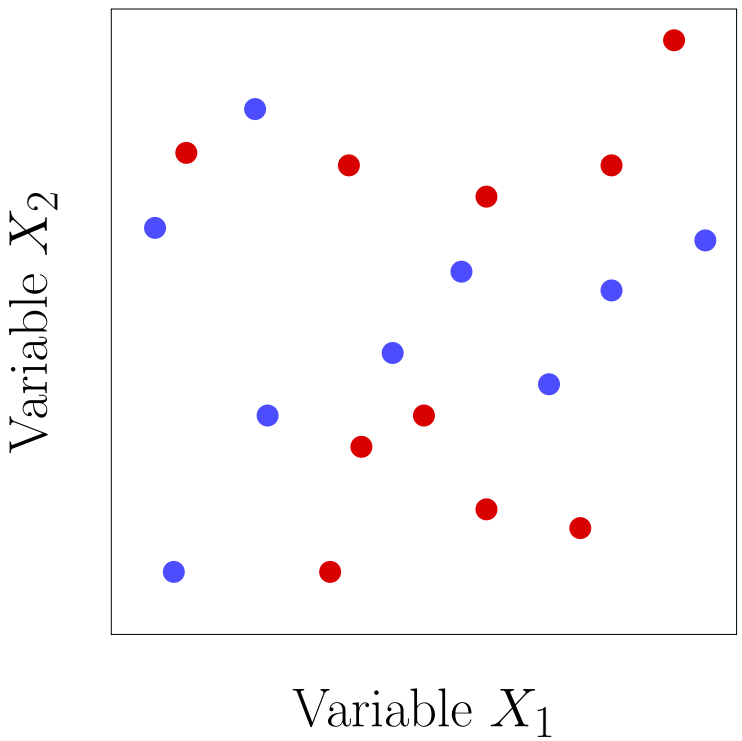
How to build a CART tree ?

- **CART** [Breiman et al., 1984]: non parametric learning algorithm.
- Idea: a tree is built recursively by recursively splitting the inoput space $\mathbb{R}^d$. into two disjoint regions until some stopping criterion is satisfied.



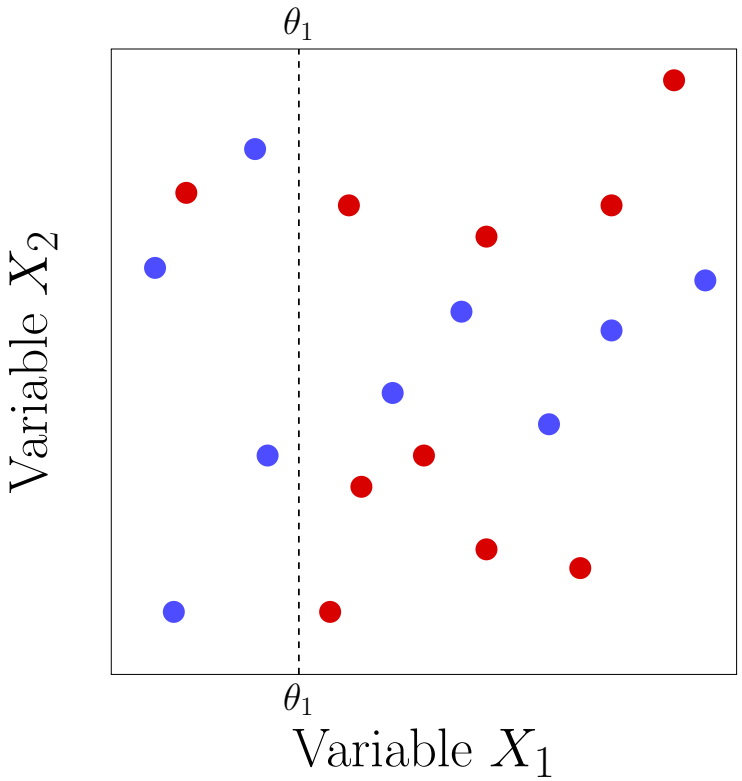
How to build a CART tree ?

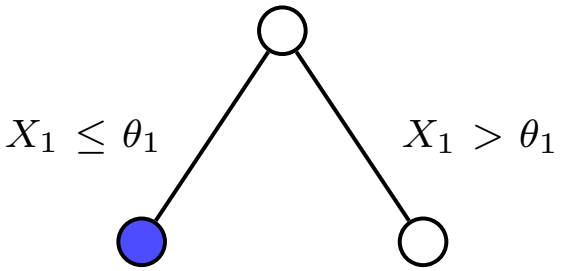
- **CART** [Breiman et al., 1984]: non parametric learning algorithm.
- Idea: a tree is built recursively by recursively splitting the inoput space $\mathbb{R}^d$. into two disjoint regions until some stopping criterion is satisfied.



How to build a CART tree ?

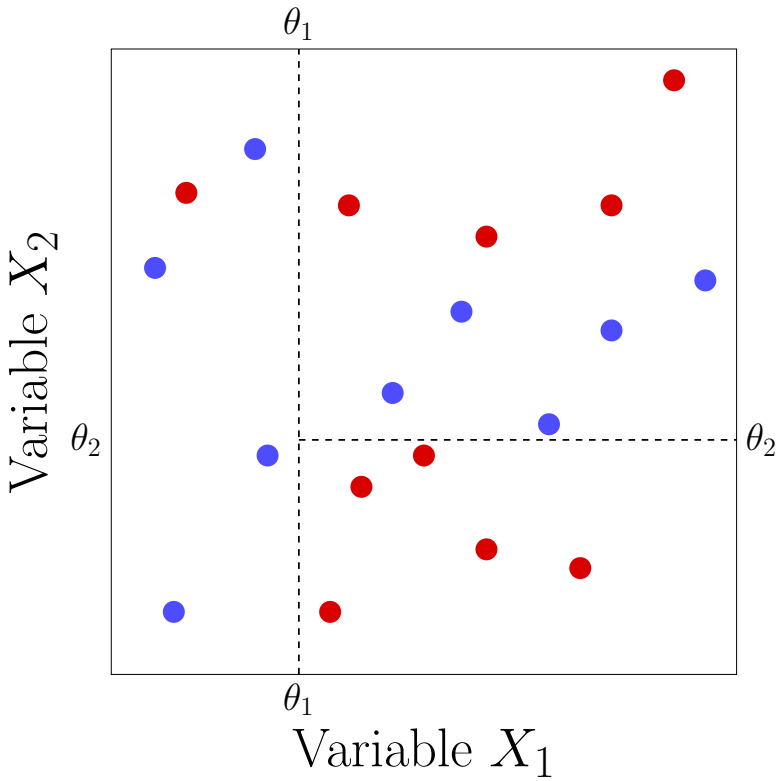
- **CART** [Breiman et al., 1984]: non parametric learning algorithm.
- Idea: a tree is built recursively by recursively splitting the inoput space $\mathbb{R}^d$. into two disjoint regions until some stopping criterion is satisfied.

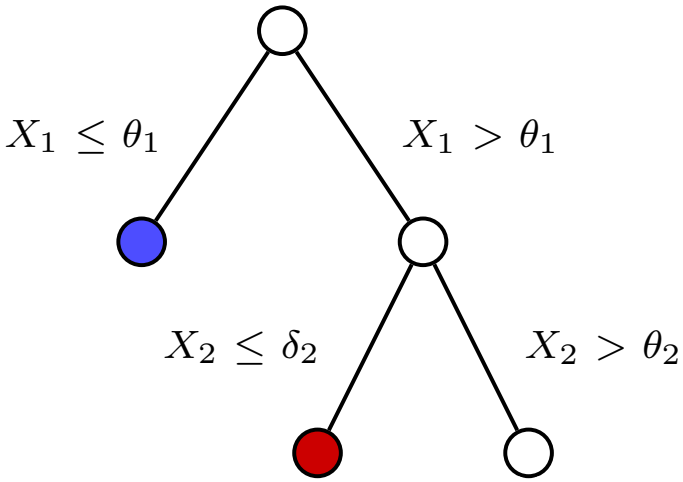




How to build a CART tree ?

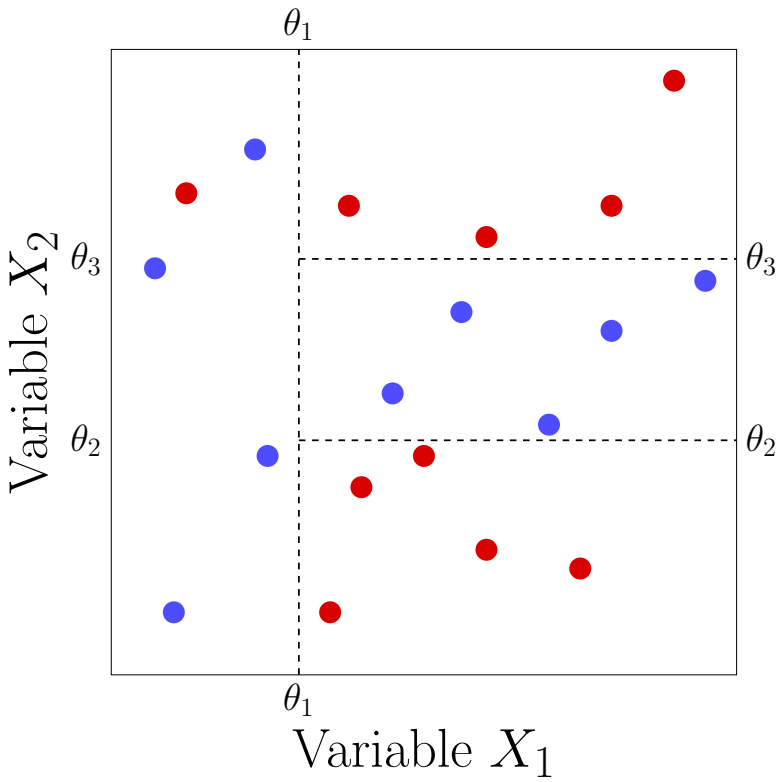
- **CART** [Breiman et al., 1984]: non parametric learning algorithm.
- Idea: a tree is built recursively by recursively splitting the inoput space $\mathbb{R}^d$. into two disjoint regions until some stopping criterion is satisfied.

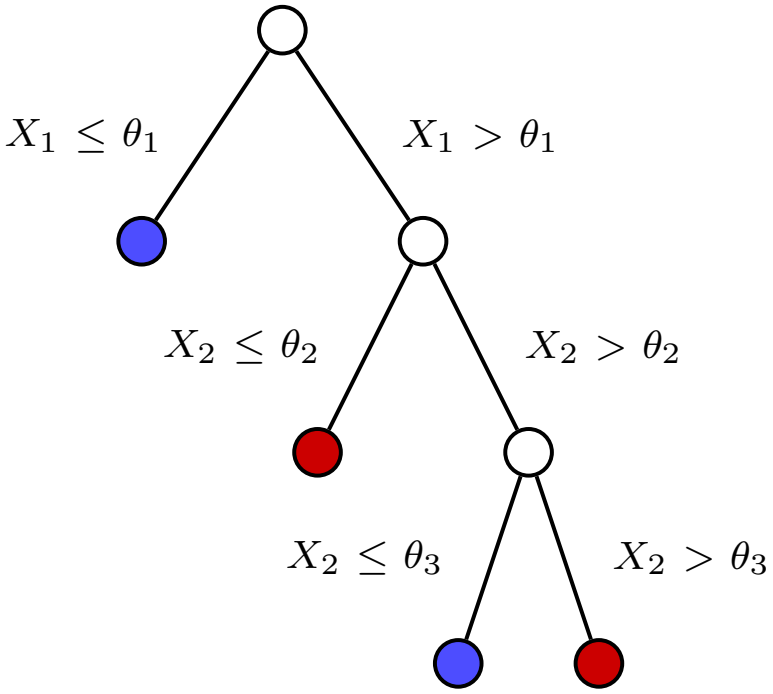




How to build a CART tree ?

- **CART** [Breiman et al., 1984]: non parametric learning algorithm.
- Idea: a tree is built recursively by recursively splitting the inoput space $\mathbb{R}^d$. into two disjoint regions until some stopping criterion is satisfied.





- **Prediction rule in each terminal node:** mean value (regression) vs. the majory vote (classification).

Decision tree are defined by

- **A splitting criterion**: maximize the impurity decrease

$$\Delta I(j,\theta) = \mathrm{I}\left(t\right) - \left[p_{t_L(j,\theta)}\mathrm{I}\left(t_L(j,\theta)\right) + p_{t_R(j,\theta)}\mathrm{I}\left(t_R(j,\theta)\right)\right],$$

  where

  - $t_L(j,\theta) = \{\mathbf{X} \in t | X_j \leq \theta\}$ and $t_R(j,\theta) = \{\mathbf{X} \in t | X_j > \theta\}$,
  - $\mathrm{I}(t)$: impurity in node t,
  - $\mathrm{I}\left(t_z\right)$: impurity in node $t_z(j,\theta)$, for $s \in \{L, R\}$,
  - p_{t_z}: proportion of observations in t that fall into $t_z(j,\theta)$, for $z \in \{L, R\}$.

  ➜ Several impurity functions (Gini/entropy for classification, variance for regression).

- **A stopping rule**: no stopping rule, grow the maximal tree and then select of the *best* subtree.

Principle of random forests

- A random forest = aggregation of many *random* decision trees.
- Random forests algorithm = bagging + features sampling.

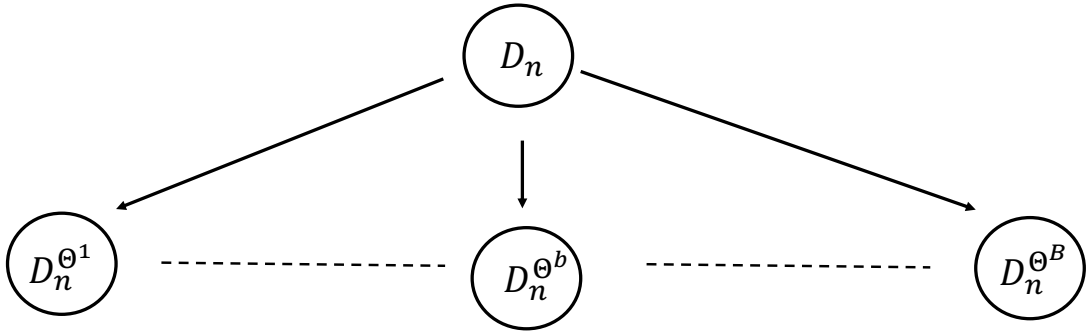
Phil Cutler

Principle of random forests

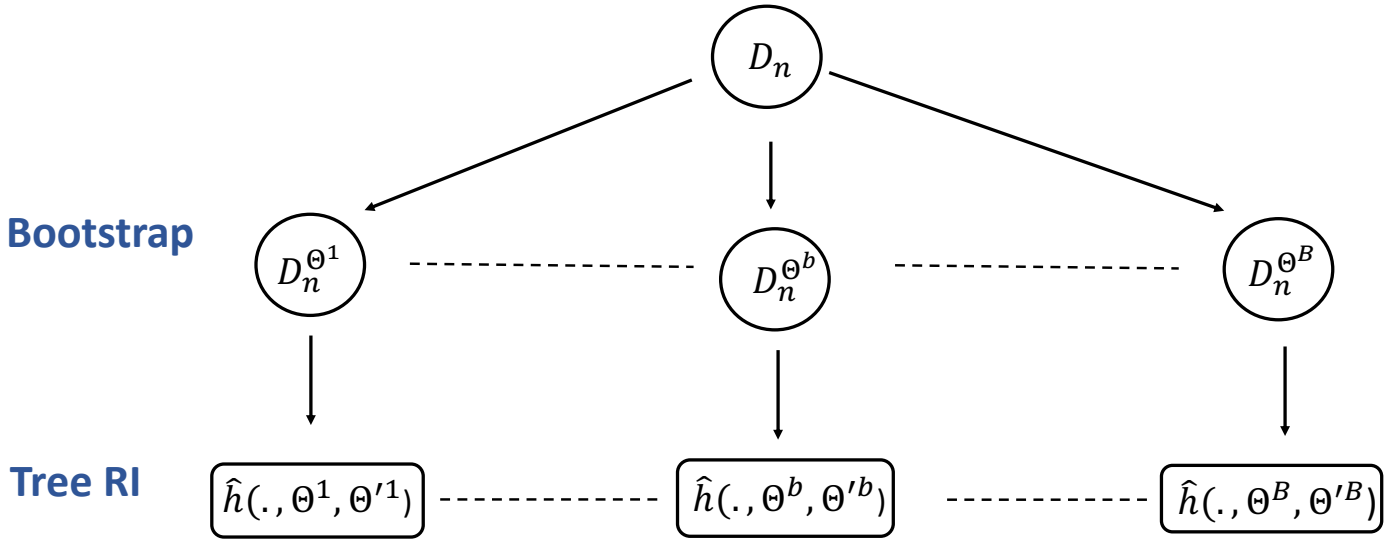
D_n

Principle of random forests

Bootstrap

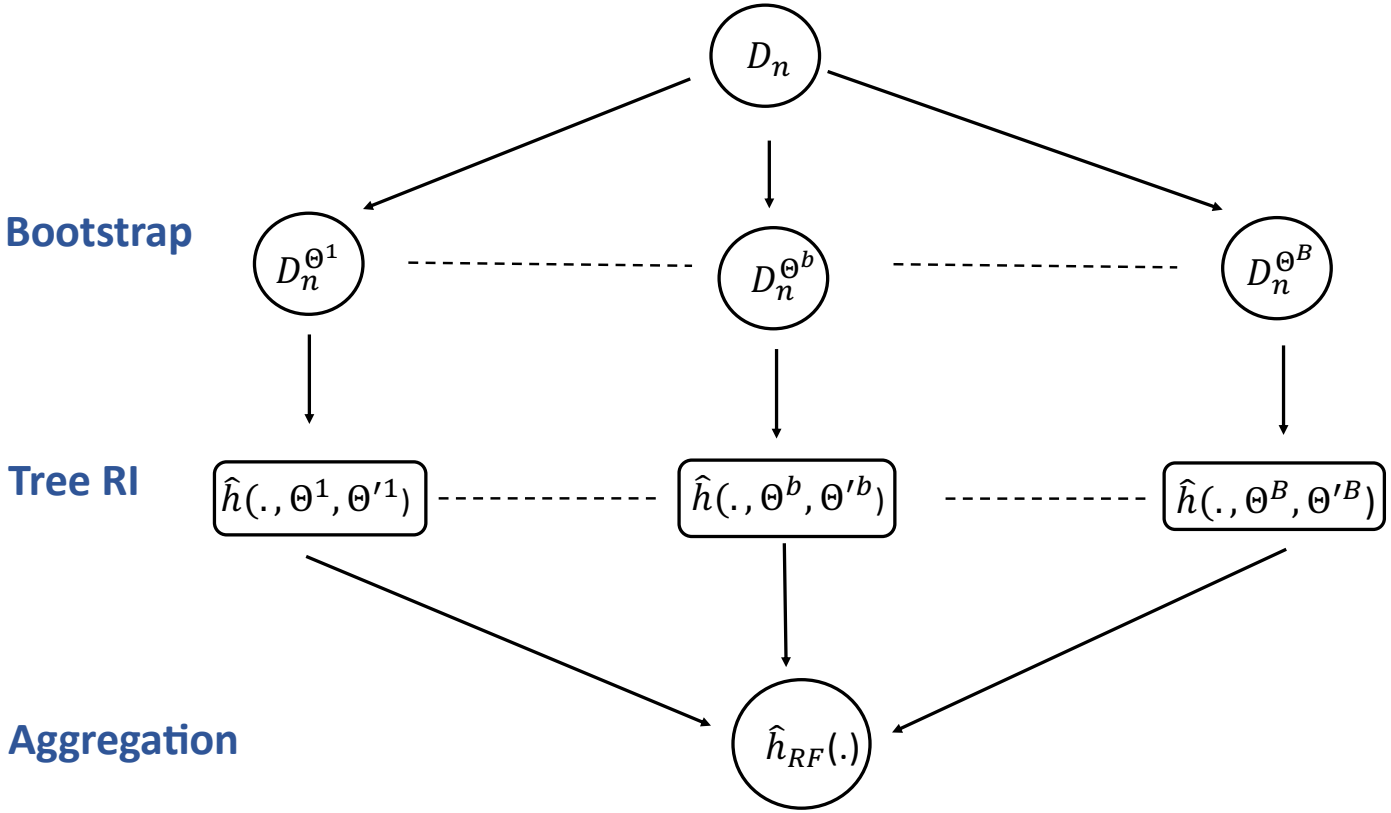


Principle of random forests



- At each node, preselect a subset of *mtry* variables eligible for splitting.

Principle of random forests



Objective

Develop a random forest method to supervised problems in which $\mathbf{X}$ has a known group structure.

Decision trees for grouped inputs

Decision trees for grouped inputs

Decision tree are defined by

- **A splitting criterion**: maximize the impurity decrease

$$\Delta I(j,\theta) = \mathrm{I}\left(t\right) - \left[p_{t_L(j,\theta)}\mathrm{I}\left(t_L(j,\theta)\right) + p_{t_R(j,\theta)}\mathrm{I}\left(t_R(j,\theta)\right)\right].$$

- **A stopping rule**: no stopping rule, grow the maximal tree and then select of the *best* subtree.

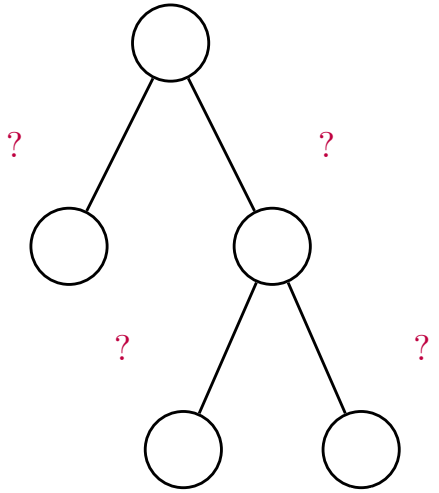
Decision trees for grouped inputs

Decision tree are defined by

- **A splitting criterion**: maximize the impurity decrease

$$\Delta I(j,\theta) = \mathrm{I}\left(t\right) - \left[p_{t_L(j,\theta)}\mathrm{I}\left(t_L(j,\theta)\right) + p_{t_R(j,\theta)}\mathrm{I}\left(t_R(j,\theta)\right)\right].$$

- **A stopping rule**: no stopping rule, grow the maximal tree and then select of the *best* subtree.



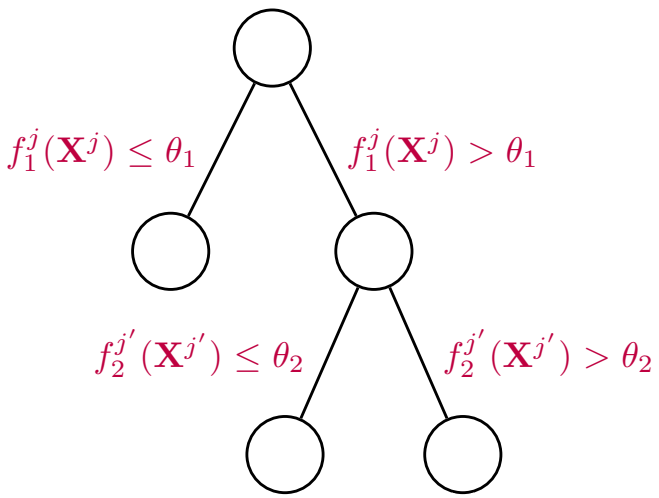
Decision trees for grouped inputs

Decision tree are defined by

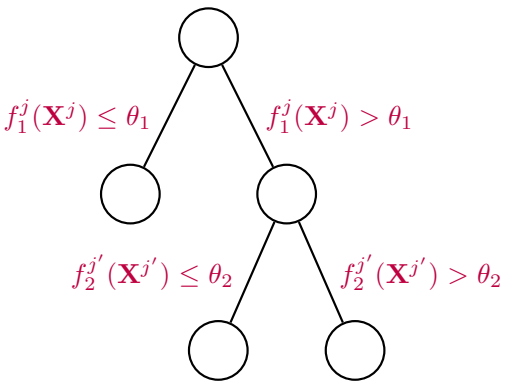
- **A splitting criterion**: maximize the impurity decrease

$$\Delta I(j,\theta) = \mathrm{I}\left(t\right) - \left[p_{t_L(j,\theta)}\mathrm{I}\left(t_L(j,\theta)\right) + p_{t_R(j,\theta)}\mathrm{I}\left(t_R(j,\theta)\right)\right].$$

- **A stopping rule**: no stopping rule, grow the maximal tree and then select of the *best* subtree.



Decision trees for grouped inputs



Definition of a split based on a group j:

- ▷ Via a regularized LDA ➜ Developement of Tree Penalized Linear Discriminant Analysis (TPLDA) [Poterie et al., 2019]
- ▷ Via a CART tree ➜ Extension of CART for Grouped Inputs (CARTGI). [Poterie, 2018, Poterie et al., 202X]

TPLDA

We want to split t into

$$t_L(j,\theta) = \{\mathbf{X} \in t | f_t^j(\mathbf{X}^j) \leq \theta\} \quad \text{and} \quad t_R(j,\theta) = \{\mathbf{X} \in t | f_t^j(\mathbf{X}^j) > \theta\}$$

with $j \in \{1, \ldots, J\}$ and $\theta \in \mathbb{R}$.

➜ Geometric defintion of a split: a linear combination of the variables in group $\mathbf{X}^j$

$$(\beta^j)^\top \mathbf{X}^j = f_t(\mathbf{X}^j)$$

Estimation method

- For group j, find $\beta^j = (\beta_1^j, \ldots, \beta_{d_j}^j) \in \mathbb{R}^{d_j}$ by maximizing the Fisher's criterion in node t:

$$\max_{\beta^j \in \mathbb{R}^{d_j}} \left\{ (\beta^j)^\top \widehat{B}_t^j \, \beta^j - \lambda_j \sum_{\ell=1}^{d_j} |\widehat{\sigma}_{t,\ell}^j \beta_\ell^j| \right\} \quad \text{subject to} \quad (\beta^j)^\top \widehat{\Sigma}_t^j \, \beta^j \leq 1,$$

with

- $\hat{\sigma}_{t,\ell}^j$: within-class standard deviation estimate of $\mathbf{X}_\ell^j$,
- $\widehat{B}_t^j$: standard estimate of the between-class covariance of $\mathbf{X}^j$,
- $\widehat{\Sigma}_t^j$: diagonal estimate of the within-class covariance of $\mathbf{X}^j$
- $\lambda_j \in \mathbb{R}^+$: regularization parameter.

Estimation method

- For group j, find $\beta^j = (\beta_1^j, \ldots, \beta_{d_j}^j) \in \mathbb{R}^{d_j}$ by maximizing the Fisher's criterion in node t:

$$\max_{\beta^j \in \mathbb{R}^{d_j}} \left\{ (\beta^j)^\top \widehat{B}_t^j \, \beta^j - \lambda_j \sum_{\ell=1}^{d_j} |\widehat{\sigma}_{t,\ell}^j \beta_\ell^j| \right\} \quad \text{subject to} \quad (\beta^j)^\top \widehat{\Sigma}_t^j \, \beta^j \leq 1,$$

  with
  - $\hat{\sigma}_{t,\ell}^j$: within-class standard deviation estimate of $\mathbf{X}_\ell^j$,
  - $\widehat{B}_t^j$: standard estimate of the between-class covariance of $\mathbf{X}^j$,
  - $\widehat{\Sigma}_t^j$: diagonal estimate of the within-class covariance of $\mathbf{X}^j$
  - $\lambda_j \in \mathbb{R}^+$: regularization parameter.

- Find a split for each group, then select the one that maximizes

$$\Delta I(j, \theta) = \text{pen}(d_j) \left\{ \mathrm{I}(t) - [p_{t_L(j,\theta)} \mathrm{I}(t_L(j,\theta)) + p_{t_R(j,\theta)} \mathrm{I}(t_R(j,\theta))] \right\},$$

  where $\text{pen}(d_j)$ is decreasing function of the group size d_j.

Non linear approach: CARTGI

We want to split t into several disjoint nodes using information from a group $\mathbf{X}^j$, ($j \in \{1, \ldots, J\}$).

Geometric defintion of a split: a partition of the input space defined according to variables of a group $\mathbf{X}^j$.

CARTGI

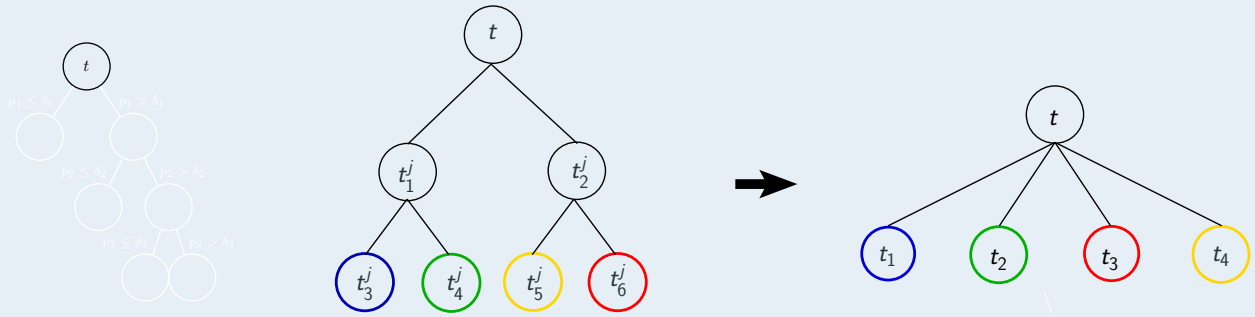
Estimation method

- For group j, build a CART tree with root t and depth $\mathbf{D_j}$

Estimation method

- For group j, build a CART tree with root t and depth $\mathbf{D_j}$

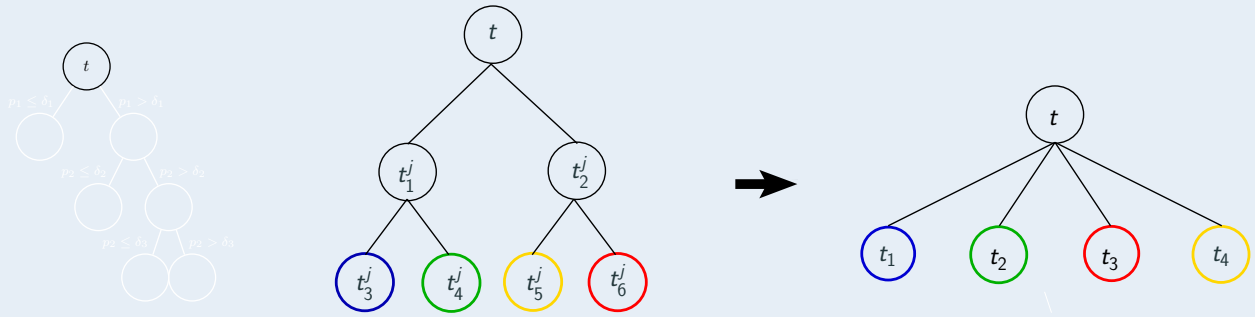
Node t **Tree on node t** **Split of t**



Estimation method

- For group j, build a CART tree with root t and depth $\mathbf{D_j}$

Node t **Tree on node t** **Split of t**



- Find a split for each group, then select the one that maximizes

$$\Delta I(j) = \text{pen}(d_j)\left[n_t I(t) - \sum_{l=1}^{L} n_{t_l(j)} I(t_l(j))\right]$$

where $\text{pen}(d_j)$ is decreasing function of the group size d_j.

Pruning strategy

Find a pruned subtree of $T_{\max}$ that minimizes

$$\mathcal{R}_\alpha(T) \;=\; \mathcal{R}(T, \mathcal{D}_n) \;+\; \alpha|\widetilde{T}|, \quad \alpha \in \mathbb{R}^+,$$

where

- $\mathcal{R}(T, \mathcal{D}_n)$: the prediction error,
- $|\widetilde{T}|$: the number of leaves of T,
- α: a tuning parameter which controls the complexity of the tree.

Pruning strategy

Theorem: Generalized cost-complexity pruning [Poterie, 2018]

For any non-trivial and non-binary tree T with root t_1, there exist a unique sequence

$$0 = \alpha_1 < \ldots < \alpha_K = \infty$$

and a unique sequence of nested subtrees of T

$$T \succeq T_1 \succ \ldots \succ T_K = \{t_1\}$$

such that for every $1 \leq k < K$,

$$\forall\, \alpha \in [\alpha_k; \alpha_{k+1}[, \quad T_k \;=\; \underset{T' \preceq T}{\operatorname{argmin}}\, \mathcal{R}_\alpha(T'), \text{ and}$$

$$\forall\, \alpha \geq \alpha_K, \quad T_K \;=\; \underset{T' \preceq T}{\operatorname{argmin}}\, \mathcal{R}_\alpha(T').$$

➜ In practice, crossvalidation and train-test split can be used to select the best subtree in the sequence.

Random forests for grouped inputs (RFGI)

Principle of random forests for groups of inputs (RFGI)

RFGI algorithm = bagging + **features and groups** sampling.

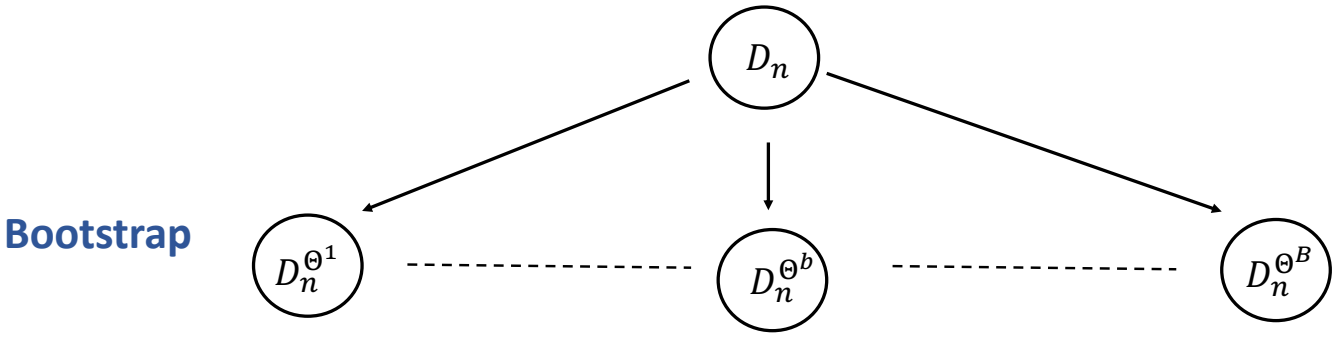
Principle of random forests for groups of inputs (RFGI)

RFGI algorithm = bagging + **features and groups** sampling.

D_n

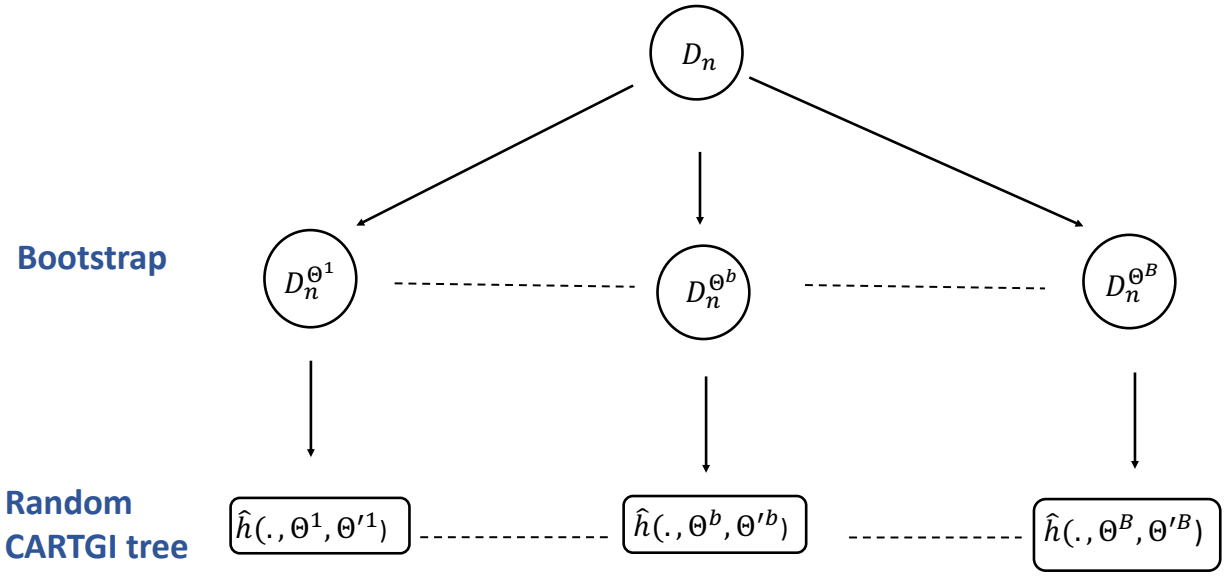
Principle of random forests for groups of inputs (RFGI)

RFGI algorithm = bagging + **features and groups** sampling.



Principle of random forests for groups of inputs (RFGI)

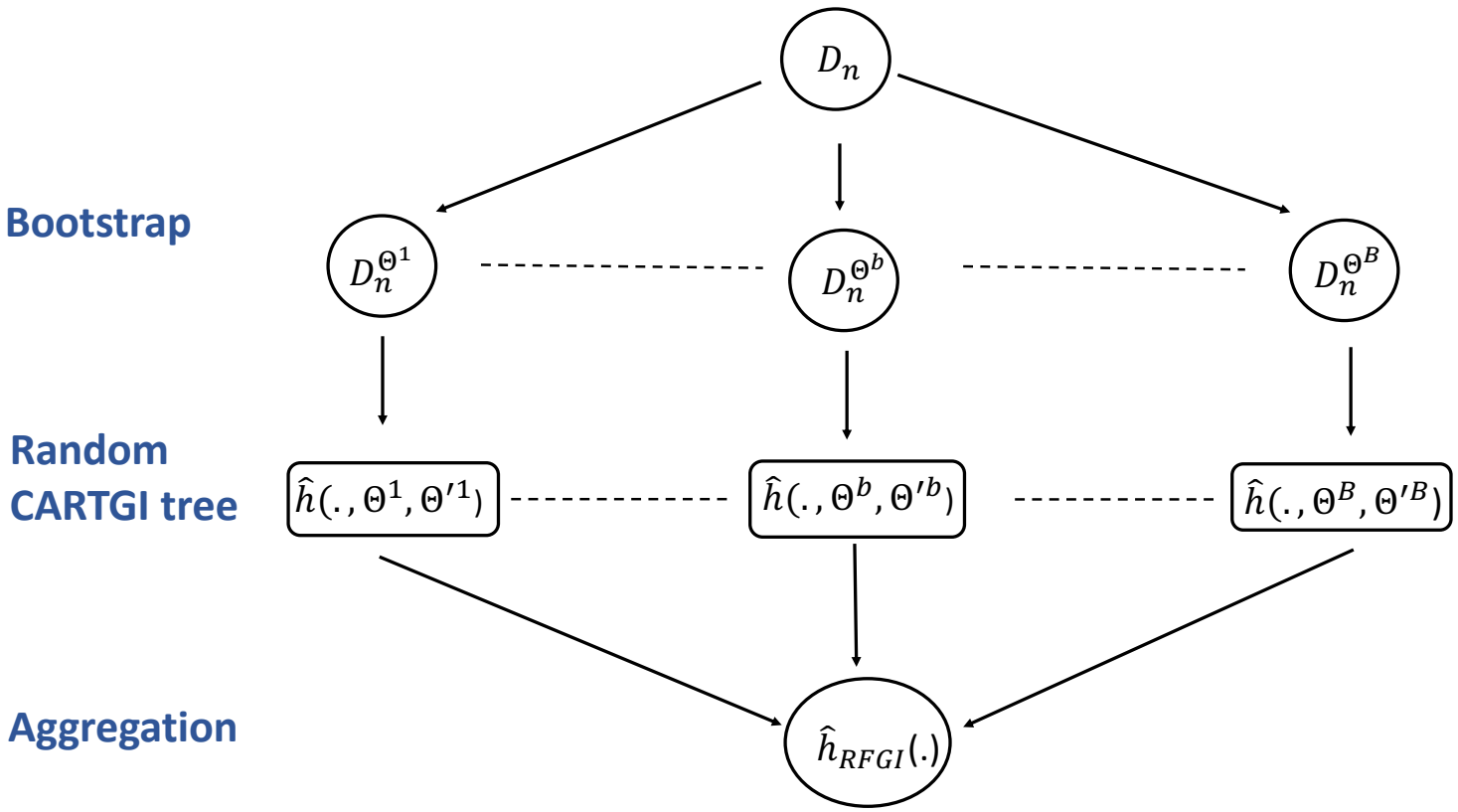
RFGI algorithm = bagging + **features and groups** sampling.



- At each node, preselect randomly `mgrp` groups eligible for splitting.
- Build a tree RI on each selected groups by preselecting `mvar` inputs eligible for splitting at each step.

Principle of random forests for groups of inputs (RFGI)

RFGI algorithm = bagging + **features and groups** sampling.



RFGI

RFGI algorithm

Input: an observation $\mathbf{x} \in \mathbb{R}^d$, $\mathcal{D}_n$, $\texttt{ntree}$, $\texttt{mgrp}$, $\texttt{mvar}_j$, D_j, for all $j = 1, \ldots, J$.

Repeat $\texttt{ntree}$ **times the following steps:**

1. Build the bootstrap sample $\mathcal{D}_n^b$.
2. Grow a maximal "random" CARTGI tree $\hat{h}(., \Theta^b, \Theta'^b)$.
   - Use only a random subset of $\texttt{mgrp}$ groups to select a split.
   - Grow random CART trees
     $\Rightarrow$ at each step, use only a random subset of $\texttt{mvar}_j$ inputs.
   - No pruning phase

Output: prediction of the random forest for observation $\mathbf{x}$

$$\hat{h}_{RFGI}(\mathbf{x}) = \textit{aggregation}\left\{\hat{h}(\mathbf{x}, \Theta^1, \Theta'^1), \ldots, \hat{h}(\mathbf{x}, \Theta^{\texttt{ntree}}, \Theta'^{\texttt{ntree}})\right\}.$$

RFGI parameters

- $\texttt{mgrp}/\texttt{mvar}_j$ = the number of selected groups/variables
  - ➜ Important parameters, analogues of `mtry`.
  - ➜ Default values:
    - $\texttt{mgrp}= J/3$ and $\texttt{mvar}_j = d_j/3$ in regression,
    - $\texttt{mgrp}= \sqrt{J}$ and $\texttt{mvar}_j = \sqrt{d_j}$ in classification.

- D_j = the depth of each splitting tree.
  - ➜ Controls the trade-off between adjustment and complexity.
  - ➜ Suggested values: $D_j= 2, 3$ for all j.

Group importance and interpretation

Group importance measure

Mean decrease in Accuracy (MDA) for group $\mathbf{X}^j$: weighted difference between the forest error $err(\hat{h}_{RFGI})$ and the permuted forest error $\widetilde{err}^j(\hat{h}_{RFGI})$

$$MDA(\mathbf{X}^j) = \frac{1}{d_j}\left\{\widetilde{err}^j(\hat{h}_{RFGI}) - err(\hat{h}_{RFGI})\right\}.$$

➜ Adaptation of the measure of grouped importance [Gregorutti et al., 2015].

Numerical experiments

Numerical experiments

Model 1: no "real" group structure (block covariance matrix).

- $n = 500$, $J = 10$ groups including 5 predictive groups, $d_j = 10$.

Model 2: group structure, non-linear relationship + interactions in groups.

- $n = 1000$, $J = 10$ groups including 2 predictive groups, $d_j = 5$.

Model 3: group structure, linear relationship in groups.

- $n = 1000$, $J = 10$ groups including 2 predictive groups, $d_j = 5$ or $d_j = 50$.

	RFGI	RF	CARTGI	TPLDA	CART	Group-Lasso
Model 1						
AUC	**0.83 (0.01)**	**0.83 (0.01)**	0.69 (0.03)	0.76 (0.02)	0.68 (0.03)	0.67 (0.02)
Error	**0.24 (0.01)**	**0.23 (0.01)**	0.33 (0.02)	0.28 (0.02)	0.33 (0.02)	0.35 (0.03)
Model 2						
AUC	**0.94 (0.02)**	0.85 (0.04)	0.78 (0.04)	0.61 (0.09)	0.73 (0.06)	0.50 (0.03)
Error	**0.15 (0.03)**	0.24 (0.04)	0.25 (0.04)	0.41 (0.09)	0.31 (0.06)	0.49 (0.02)
Model 3						
AUC	0.86 (0.03)	0.82 (0.03)	0.73 (0.03)	0.78 (0.03)	0.71 (0.04)	**0.90 (0.02)**
Error	0.21 (0.03)	0.25 (0.03)	0.30 (0.03)	0.23 (0.03)	0.32 (0.04)	**0.15 (0.02)**

Tuning parameters are selected by using cross-validation and a validation sample.

Simulations: measure of group importance

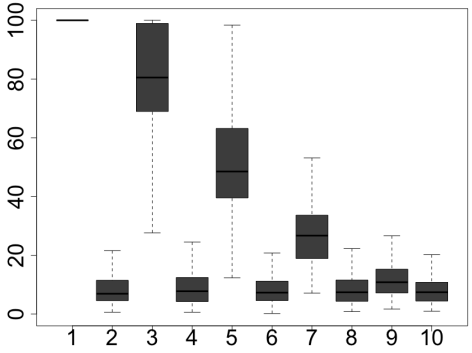
Model 1: no "real" group structure (block covariance matrix), Gaussian mixture.

- $J = 10$ groups with each $d_j = 10$ variables.
- No between-group correlation.
- 5 relevant groups = groups with ood index.
- $I(\mathbf{X}^1) > I(\mathbf{X}^3) > I(\mathbf{X}^5) > I(\mathbf{X}^7) > I(\mathbf{X}^9)$.

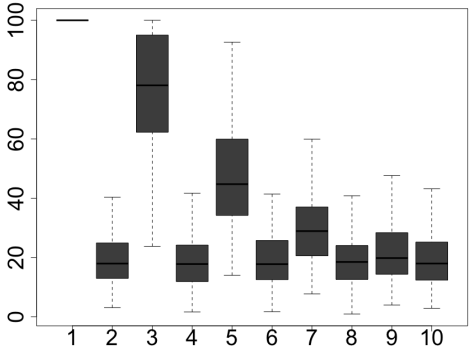
Simulations: measure of group importance

Model 1: no "real" group structure (block covariance matrix), Gaussian mixture.

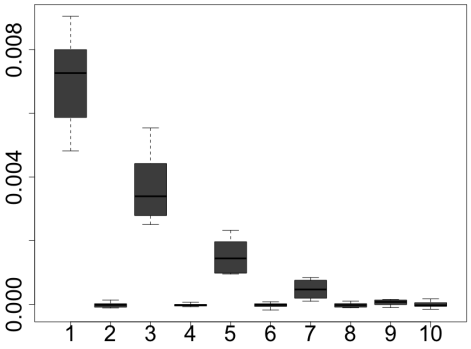
- $J = 10$ groups with each $d_j = 10$ variables.
- No between-group correlation.
- 5 relevant groups = groups with ood index.
- $I(\mathbf{X}^1) > I(\mathbf{X}^3) > I(\mathbf{X}^5) > I(\mathbf{X}^7) > I(\mathbf{X}^9)$.



TPLDA



CARTGV



RFGV

Applications to real data

Applications to real data

Gene expression data 1: colitis data [Burczynski et al., 2006].

- $n = 127$ patients whom 85 patients with Crohn's disease or ulcerative colitis
- $\mathbf{d} = \mathbf{7818}$ genes grouped into $\mathbf{J} = \mathbf{275}$ groups including (average size: 30 genes).

Gene expression data 2: breast cancer data [Ma et al., 2004].

- $n = 60$ patients whom 28 with cancer reccurence.
- $\mathbf{d} = \mathbf{4246}$ genes grouped into $\mathbf{J} = \mathbf{268}$ groups including (average size: 18.5 genes).

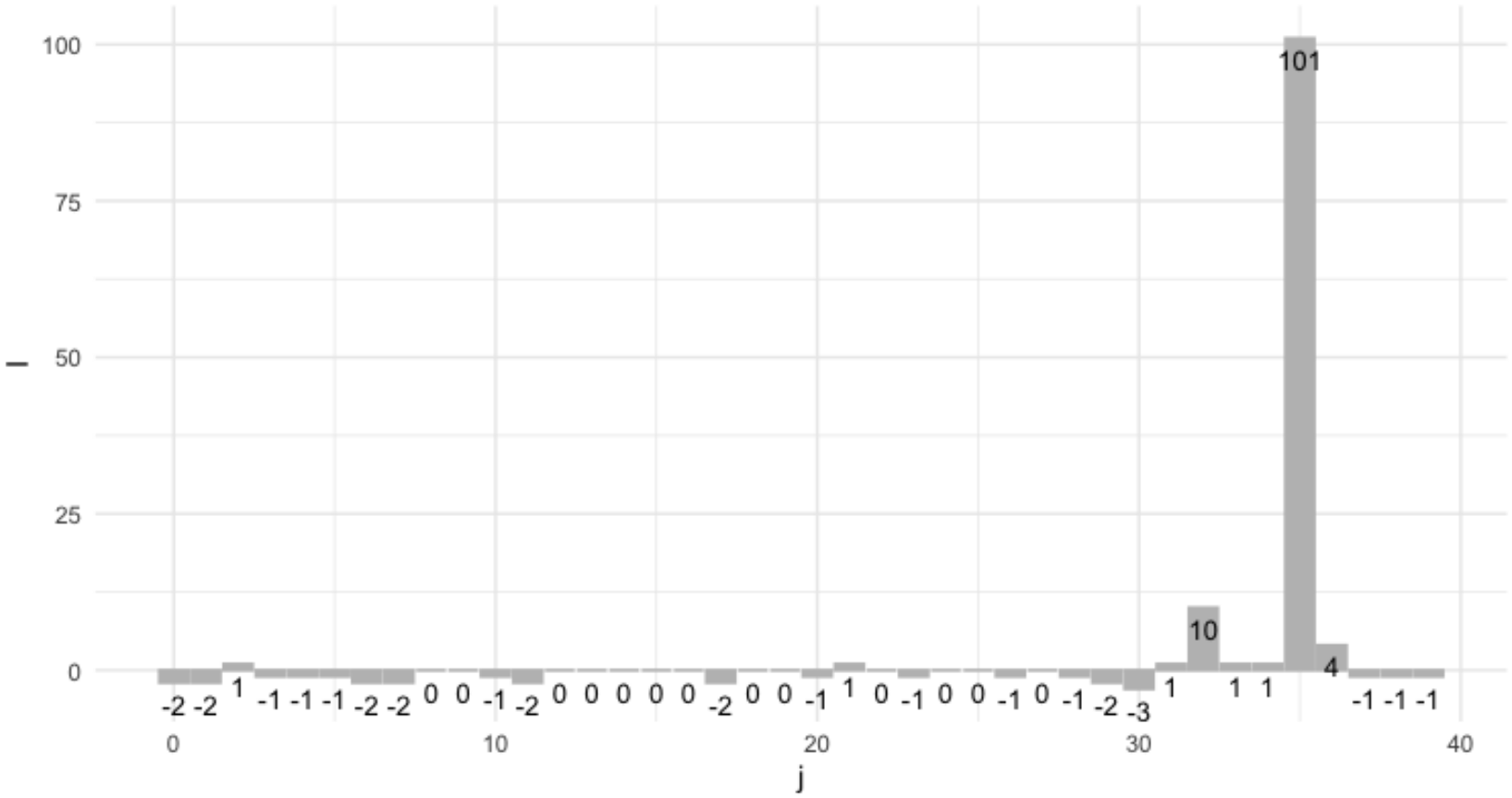
Extreme weather event data : Flash Flood Information Retrieval data [Wilkho et al., 2023]

- $\mathbf{n} = \mathbf{14420}$ webpages whom 1560 are related to flash flood **(10,8%)**.
- $d = 40$ synthetic variables grouped into $J = 8$ groups including (group size between 3 and 10 variables).

	RFGI	RF	Sparse-group lasso*	Group-Lasso*
colitis data				
Accuracy	0.93 (0.04)	0.93 (0.04)	0.87	0.84
breast cancer data				
Accuracy	0.53 (0.12)	0.48 (0.12)	0.70	0.60
flood data				
Accuracy	0.95 (2E-3)	0.95 (2E-3)	-	-

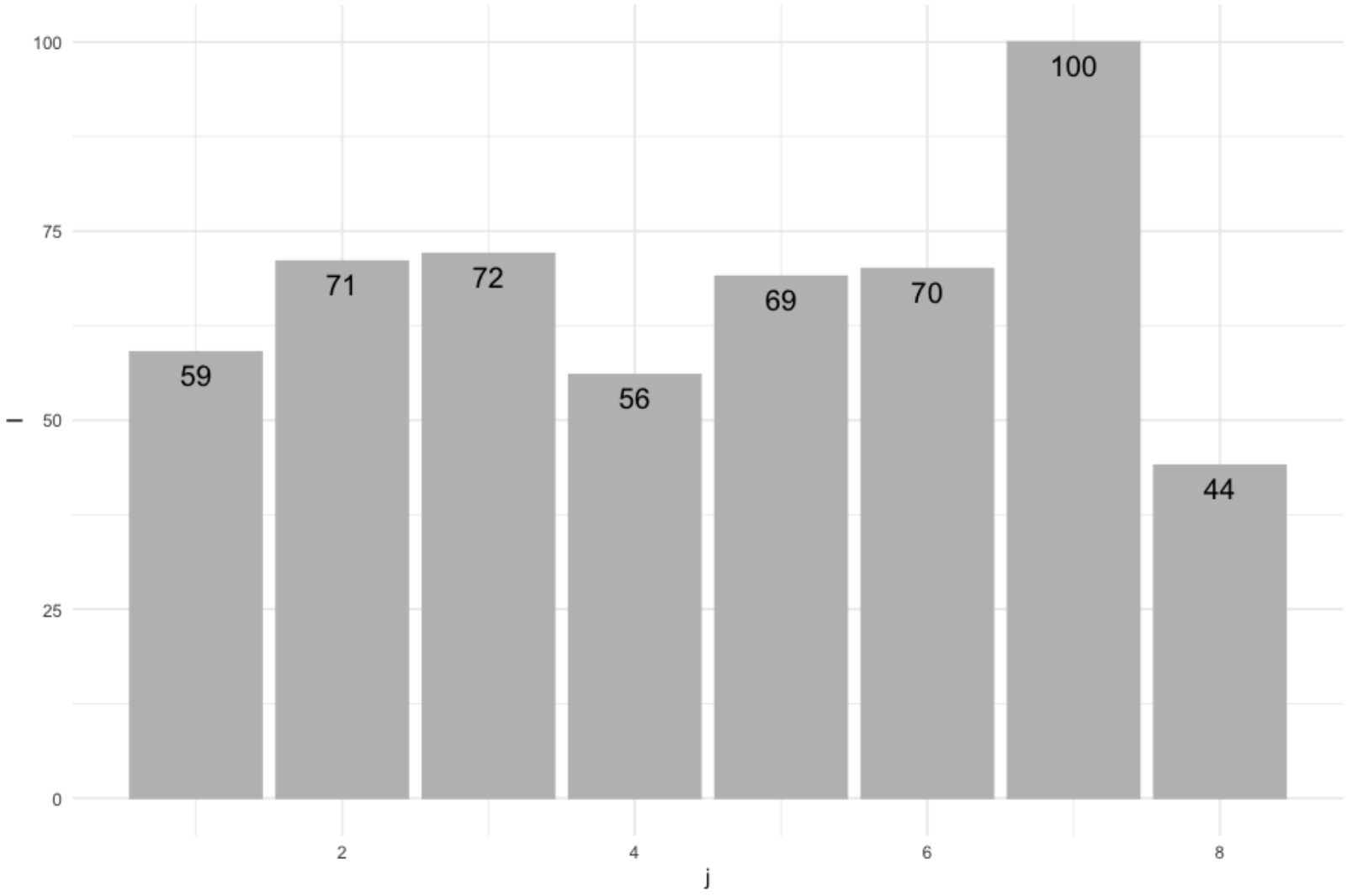
Results from [Simon et al., 2013].

Applications to real data: group/variable importance



Normalized MDA for variables in RF

Applications to real data: group/variable importance



Normalized MDA for groups in RFGV

Conclusions, perspectives

Conclusions, perspectives

- **Development of decision-tree and random forests methods for groups of variables**.
- Methods implemented in `R` and `Python`.
  - In `R`: *TPLDA* and *dtrfgv* (available on github).
  - Fast implemention in `Cython`: extension of the library scikit-learn (available on github).

Conclusions, perspectives

- **Development of decision-tree and random forests methods for groups of variables**.
- Methods implemented in `R` and `Python`.
  - In `R`: *TPLDA* and *dtrfgv* (available on github).
  - Fast implemention in `Cython`: extension of the library scikit-learn (available on github).

Actual questions:

- How to automatically define the group structure ?
- How to build random forest for data with spatio-temporal data ?

References i

Breiman, L. (2001).
Random forests.
Machine learning, 45(1):5–32.

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

Burczynski, M. E., Peterson, R. L., Twine, N. C., Zuberek, K. A., Brodeur, B. J., Casciotti, L., Maganti, V., Reddy, P. S., Strahs, A., Immermann, F., et al. (2006).
Molecular classification of crohn's disease and ulcerative colitis patients using transcriptional profiles in peripheral blood mononuclear cells.
The journal of molecular diagnostics, 8(1):51–61.

References ii

Gregorutti, B., Michel, B., and Saint-Pierre, P. (2015).
Grouped variable importance with random forests and application to multiple functional data analysis.
Computational Statistics & Data Analysis, 90:15–35.

Ma, X.-J., Wang, Z., Ryan, P. D., Isakoff, S. J., Barmettler, A., Fuller, A., Muir, B., Mohapatra, G., Salunga, R., Tuggle, J. T., et al. (2004).
A two-gene expression ratio predicts clinical outcome in breast cancer patients treated with tamoxifen.
Cancer cell, 5(6):607–616.

Poterie, A. (2018).
Arbres de décision et forêts aléatoires pour variables groupées.
PhD thesis, Université Rennes 1.

Poterie, A., Dupuy, J.-F., Monbet, V., and Rouviere, L. (2019).
Classification tree algorithm for grouped variables.
Computational Statistics, 34(4):1613–1648.

References iii

Poterie, A., Monbet, V., and Rouviere, L. (202X).
Random forests for grouped inputs.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013).
A sparse-group lasso.
Journal of computational and graphical statistics, 22(2):231–245.

Wilkho, R. S., Gharaibeh, N. G., Chang, S., and Zou, L. (2023).
Ff-ir: An information retrieval system for flash flood events developed by integrating public-domain data and machine learning.
Environmental Modelling & Software, page 105734.