CART Trees and Random Forests

Jean-Michel Poggi (U. Paris and LMO, U. Paris-Saclay, France)

Master 2 Course in Statistics
Universidad de la República – Facultad de Ingeniería,
Montevideo, Uruguay

February 2020

Acknowledgments and references

ROBIN GENUER
ET JEAN-MICHEL POGGI
PRATIQUE DE LA STATISTIQUE
Les forêts aléatoires avec R
PUR
Presses Universitaires de Rennes

- Text and slides (except Section 5) written in collaboration with Robin Genuer
- Acknowledgments: S. Arlot, S. Gey, C. Tuleau-Malot and N. Villa-Vialaneix
- A freely accessible reference, in French but with full of references:
  Robin Genuer, Jean-Michel Poggi, *Arbres CART et Forêts aléatoires, Importance et sélection de variables*, 45 pages, 2017 [a]
  http://up5.fr/hal-01387654v2
- *Les forêts aléatoires avec R*
  Genuer, Poggi (2019)
  Presses Universitaires de Rennes (PUR)

[a] book chapter of "Apprentissage Statistique et Données Massives", Technip, p. 295-342, 2018

Outline

1 Introduction

2 CART Trees

3 Random Forests

4 Variable Selection

5 An industrial application

6 Scaling RF for Big Data

Leo Breiman

- From CART to Random Forests: 20 years of a scientific trajectory
- Olshen, Breiman (2001) et Cutler (2010)
- First in probability from a perspective very close to pure mathematics, then he hugely impacted applied statistics and statistical learning
- A series or papers published in the *Annals of Statistics* and in *Machine Learning*

General framework

$\mathcal{L}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ random variables i.i.d. from the same distribution as (X, Y)

$X \in \mathbb{R}^p$ (explanatory variables); we could also have $X \in \mathbb{R}^{p'} \otimes \mathcal{Q}$ mixing numerical and nominal variables

$Y \in \mathcal{Y}$ (response variable):

- $\mathcal{Y} = \mathbb{R}$: regression
- $\mathcal{Y} = \{1, \ldots, L\}$: classification

Aim: to build a predictor $\widehat{h} : \mathbb{R}^p \to \mathcal{Y}$

CART Trees Breiman et al. (1984)

- part of the family of decision tree methods
- algorithm which is the basis of very effective methods

Random Forests Breiman (2001)

- part of the family of ensemble methods
- algorithm of statistical learning, extremely efficient, both for problems of classification and of regression

Outline

1 Introduction

2 CART Trees

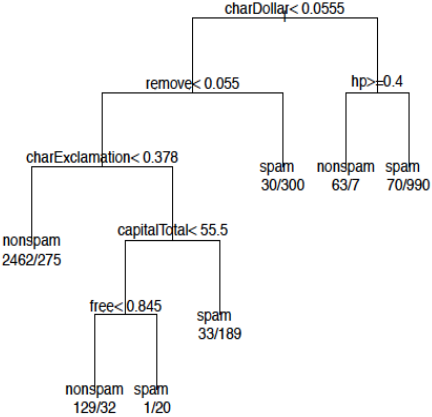
3 Random Forests

4 Variable Selection

5 An industrial application

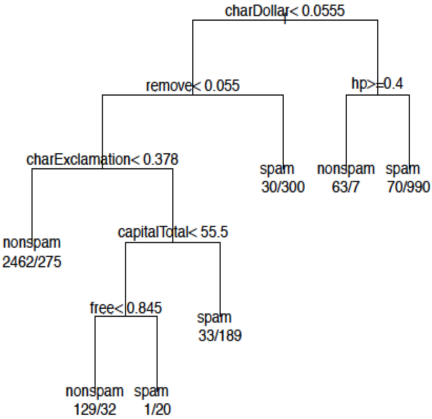
6 Scaling RF for Big Data

spam dataset



- Design an automatic spam detector (supervised learning problem)
- n=4601 email messages (1813 spams, 40%)
- p=57 predictors:
  - 54 are the % of words in the email matching a given word or character like "$", "!", "remove", "free"
  - 3 related to the lengths of uninterrupted sequences of capital letters (average, maximum, sum)

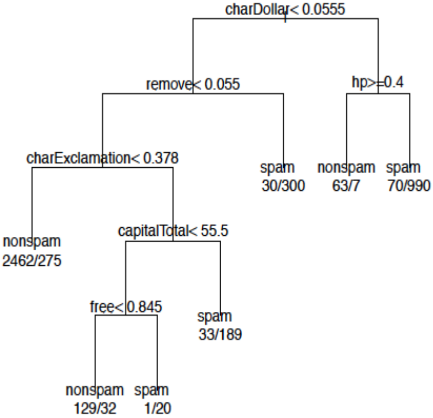
CART tree on *spam* dataset



- CART tree structure: 5 internal nodes and 7 leaves, splits involve *charDollar*, *remove*, *hp*, *free*, *charExclamation* and *capitalTotal*

- CART tree prediction: leaf labels give the prediction (spam or nonspam) and conditional distribution of Y

CART tree on *spam* dataset: prediction and interpretation



- How to get the prediction: start at the root and answer questions on x sequentially (*if* condition *then* LEFT *else* RIGHT) until a leaf is reached. The label gives the predicted value of $\hat{y}$

- Interpretation: path root-3rd leaf: an email with many "$", "!", "remove", capital letters and "free" is almost always a spam

CART *vs* decision trees

- Sometimes introduced before CART, other methods for building decision trees are available:
    - CHAID see Kass (1980)
    - $C4.5$ see Quinlan (1993)
- The decision tree method suffered from strong justified criticisms and CART offers them a conceptual framework of model selection, which gives them both broad applicability, ease of interpretation and theoretical guarantees
- The actuality of decision trees is still important, see the two recent surveys:
    - Patil et Bichkar (2012) in computer science
    - Loh (2014) in statistics

Construction

Tree: piecewise constant predictor, obtained by recursive dyadic partitioning of $\mathbb{R}^p$

Restriction: splits parallel to axes

Typically, at each step of the binary partitioning, we seek the "best" split to purify the resulting nodes.
We aim at separating the data of the current node, by looking for the "best" split leading to the maximal decrease in heterogeneity of the two child nodes

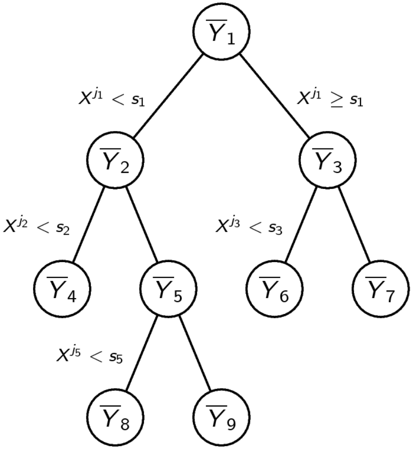
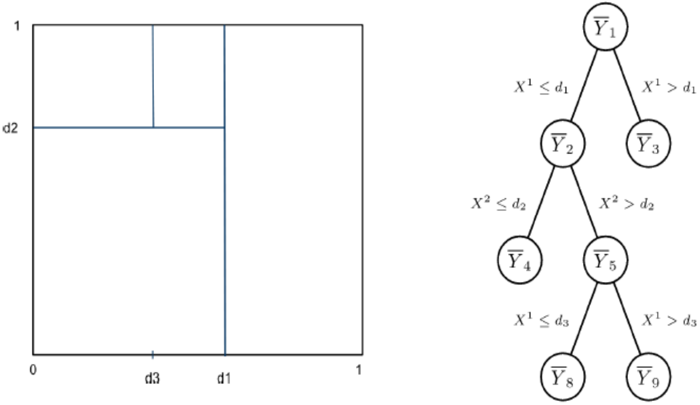


Figure: Regression tree

CART tree and piecewise constant function



Regression tree *vs* Classification tree

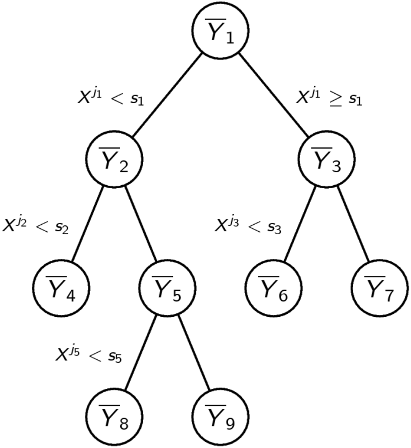


Figure: Regression tree

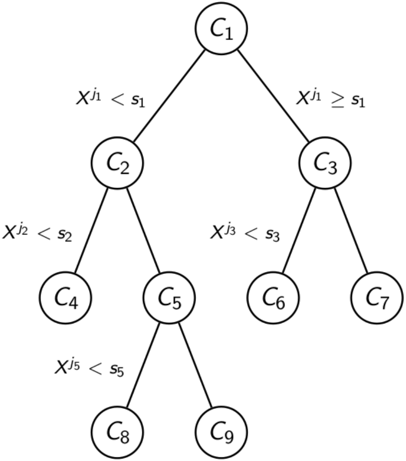


Figure: Classification tree

Construction

- Split:
  $\{X^j \leq d\} \cup \{X^j > d\}$ or $\{X^j \in d\} \cup \{X^j \in \bar{d}\}$
- Regression. Denoting the variance of the node t by $V(t) = \frac{1}{\#t} \sum_{i:x_i \in t} (y_i - \overline{y}_t)^2$, we minimize the intra-group (internal) variance after the split of t in 2 children t_L and t_R:
  $$\frac{\#t_L}{n} V(t_L) + \frac{\#t_R}{n} V(t_R)$$
- Classification. We define the impurity of nodes most often through the Gini index. The Gini index of a node t: $\Phi(t) = \sum_{c=1}^{L} \hat{p}_t^c (1 - \hat{p}_t^c)$, where $\hat{p}_t^c$ is the proportion of observations of class c in the node t. We maximize:
  $$\Phi(t) - (\frac{\#t_L}{n} \Phi(t_L) + \frac{\#t_R}{n} \Phi(t_R))$$

Maximal Tree and Pruning

Maximal tree, Stop rule:

- Recursive partitioning by local maximization of the decay of heterogeneity
- Do not split a pure node or a node containing too little data

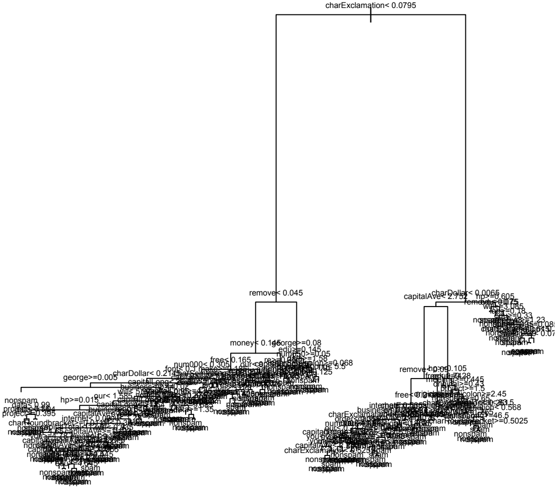
Pruning:

- The maximal tree overfits the data
- The optimal tree is a pruned subtree of the maximal tree minimizing the prediction error penalized by the complexity of the model
- Penalized criterion

$$crit_{\alpha}(T) = R_n(f, \hat{f}_{|T}, \mathcal{L}_n) + \alpha \frac{|\tilde{T}|}{n}$$

$R_n(f, \hat{f}_{|T}, \mathcal{L}_n)$ the error term (MSE for regression or the misclassification rate) and $|\tilde{T}|$ the number of leaves of T

Maximal Tree on *spam* dataset



Pruning algorithm

Proposition

The sequence of parameters $(0 = \alpha_1; \ldots; \alpha_K)$ *is strictly increasing, and for all* $1 \leq d \leq K$

$$\begin{aligned} \forall \alpha \in [\alpha_d, \alpha_{d+1}[\; T_d &= argmin_{\{T \text{ sub-tree of } T_{max}\}} crit_{\alpha}(T) \\ &= argmin_{\{T \text{ sub-tree of } T_{max}\}} crit_{\alpha_d}(T) \end{aligned}$$

So we have the following facts:

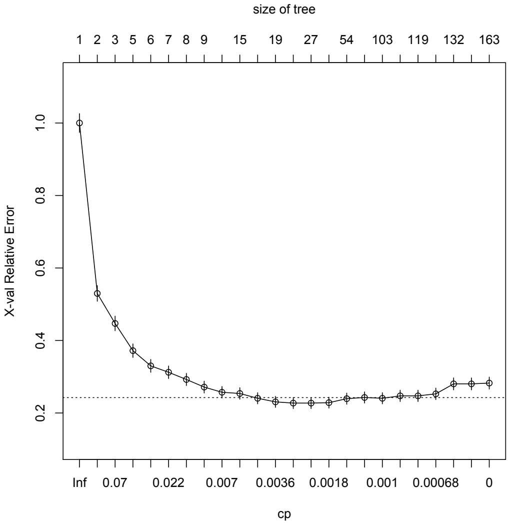
- The sequence $T_1, \ldots, T_K$ contains all the statistical information
- For any $\alpha \geqslant 0$, the subtree minimizing $crit_{\alpha}$ is a subtree of the considered sequence
- Iterative pruning algorithm does require few operations

Pruning algorithm

Input	Maximal tree T_{max}
Initialization	$\alpha_1 = 0,\ T_1 = T_{\alpha_1} = \text{argmin}_{T \text{ pruned from } T_{max}} \overline{err}(T)$ initialize $T = T_1$ and $k = 1$
Iteration	While $\lvert T \rvert > 1$, Compute $\alpha_{k+1} = \min_{\{t \text{ internal node of } T\}} \dfrac{\overline{err}(t) - \overline{err}(T_t)}{\lvert T_t \rvert - 1}$ Prune all the branches T_t of T such that $\overline{err}(T_t) + \alpha_{k+1} \lvert T_t \rvert = \overline{err}(t) + \alpha_{k+1}$ Consider T_{k+1} the obtained pruned subtree. Loop with $T = T_{k+1}$ and $k = k + 1$
Output	Trees $T_1 \succ \ldots \succ T_K = \{t_1\}$ Parameters $(0 = \alpha_1; \ldots; \alpha_K)$

Table: Informally, *start at $\alpha = 0$ and increase α continuously until the most fragile branch of the tree breaks, repeat until reaching the root*

spam dataset: the sequence of pruned subtrees



Remark about the best trees with k leaves

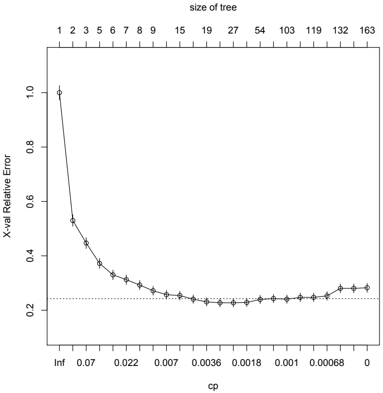


Figure: If a tree in this sequence contains k leaves, it is the best tree with k leaves. But the sequence does not contain all the best trees with k leaves for $1 \leq k \leq |T_{max}|$

A theoretical result for regression

Risk of a tree T: $R_n(f, \hat{f}_{|T}, \mathcal{L}_n) = \frac{1}{n} \sum_{(X_i, Y_i) \in \mathcal{L}_n} (Y_i - \hat{f}_{|T}(X_i))^2.$

Penalized criterion:

$$crit_\alpha(T) = R_n(f, \hat{f}_{|T}, \mathcal{L}_n) + \alpha \frac{|\tilde{T}|}{n}$$

where

- $|\tilde{T}|$ is the number of leaves of the tree T
- $\tilde{f}$ is the final estimator given by CART
- $\|.\|$ the $\mathbb{L}^2(\mathbb{R}^p, \mu)$-norm with μ the marginal distribution of X

A theoretical result for regression

Theorem (Gey, Nedelec 2005)

It exists C_1, C_2, C_3 positive constants such that:

$$\mathrm{E}\left[\|\tilde{f} - f\|^2 | \mathcal{L}_1\right] \leq C_1 \inf_{T \preceq T_{max}} \left[\inf_{u \in S_T} \|u - f\|^2 + \sigma^2 \frac{|\tilde{T}|}{n_1}\right] + \frac{C_2}{n_1} + C_3 \frac{\ln n_1}{n_2}$$

where S_T is the set of piecewise constant functions defined on the partition induced by the set of the leaves of T

- The performance of the selected tree is, at first order, of the same order of magnitude as the performance of the best predictor up the additive penalty term, thus justifying its form
- The quality of the selection of the estimator is assessed conditionally to the sample $\mathcal{L}_1$, the family of models inside which one searches being dependent on the data

A theoretical result for binary classification

- Penalized criterion : $\hat{R}_{pen}(T) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{\hat{f}_T(X_i)\neq Y_i} + \alpha|T|$
- When T_{opt} is chosen thanks to the Hold-out method with a first sample $\mathcal{L}_1$ for building and pruning T_{max} and a second sample $\mathcal{L}_2$ to choose the tree minimizing the prediction error

Theorem (Gey 2012)

Under a condition on the margin h, it exists C_1, C_2, C_3 such that :

$$\mathrm{E}\left[l(f^*, \hat{f}_{T_{opt}})|\mathcal{L}_1\right] \leq C_1 \inf_{T\preceq T_{max}} \left[\inf_{f\in S_T} l(f^*, f) + h^{-1}\frac{|T|}{n_1}\right] + \frac{C_2}{n_1} + C_3\frac{\ln n_1}{n_2}$$

where S_T is the set of classifiers defined on the partition induced by the set of the leaves of T, and
$l(f^*, f) = \mathcal{P}\left(f(X) \neq Y\right) - \mathcal{P}\left(f^*(X) \neq Y\right)$

CART in practice

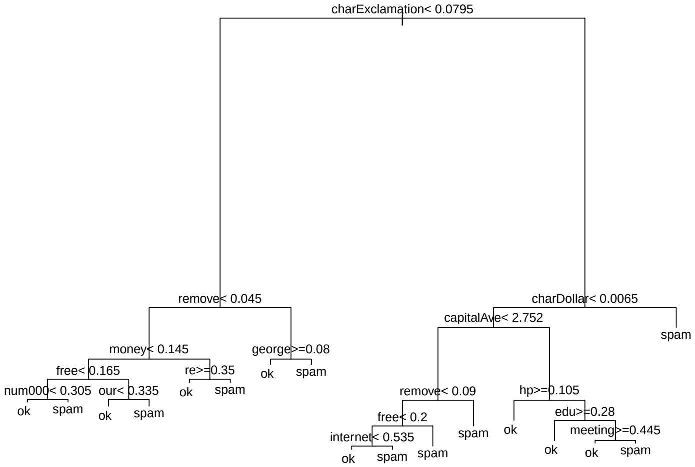
The CART trees displayed in this section are obtained by:

- *R* package *rpart*, see Therneau et al. (2015)
- with default settings (Gini index of heterogeneity of for the construction of the maximal tree and pruning by 10-fold CV)

Four trees are considered:

- the tree obtained with the default parameters (including the suboptimal tree selected using $\alpha = 0.01R(T_1)$)
- the optimal tree obtained with default parameters and using the 1-SE rule of Breiman
- an optimal stump (2-leaf tree)
- the maximal tree

spam dataset: optimal tree with 1 SE rule



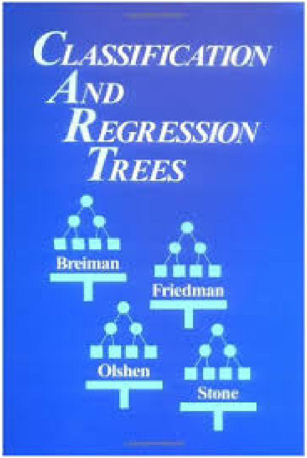
spam dataset: optimal tree with 1 SE rule

- The best pruned subtree of the maximal tree (up to 1 SE)
  - 17 leaves
  - Only 14 variables (among the 57 initial ones) appear in the splits of the 16 internal nodes: `charExclamation`, `charDollar`, `remove`, `capitalAve`, `money`, `george`, `hp`, `free`, `re`, `num000`, `our`, `edu`, `internet meeting`
- Two paths interpreted:
  - From the root to the rightmost leaf: A mail that contains a lot of `$` and of `!` is almost always a spam
  - From the root to the fifth leaf to the right: A mail that contains a lot of `!`, of capital letters and of `hp` but little of `$` is almost never a spam

Tree	2 leaves	1 SE	maximal	optimal
Empirical Error	0.208	0.073	0.000	0.062
Test Error	0.209	0.096	0.096	0.086

Table: Errors (empirical and test) of the four trees

The reference



- CART Classification And Regression Trees, Breiman et al. (1984)
- A compact and clear introduction of the CART method in the regression case, can be found in Chapter 2 of the PhD thesis S. Gey (2002), but in French ...
- See also Zhang, Singer (2010) and of course the book Hastie, Tibshirani, Friedman (2009)

Pros and cons

- Nonparametric model + data partition
- A single and versatile framework for regression and binary or multiclass classification
- Models easy to interpret
- Data do not need to be normally distributed, predictor variables are not supposed to be independent
- Numerical predictors can be mixed with nominal ones
- Competing primary splits: manual growing of the maximal tree
- Clever way to consider missing values in prediction: surrogate splits
- Main but huge drawback: lack of stability
- CART is a base predictor for: bagging, boosting, random forests

Extensions and variants

- Variants
  - In regression, more regular predictors than piecewise constants functions, e.g. MARS introduced by Friedman (1991)
  - Ortho-CART Donoho et al. (1997), in image processing, dyadic splits + pruning using a classical algorithm for the choice to the wavelet packets best basis
  - Dyadic-CART, ideas generalized by Blanchard et al. (2007)
- Extensions
  - One of the most widely used extensions: CART for survival data, LeBlanc, Crowley (1993), Molinaro et al. (2004) and the recent survey paper Bou-Hamad et al. (2011)
  - Extension to spatial data with kriging type ideas see Bel et al. (2009) and more recently Bar-Hen et al. (2019)
  - In Zhang, Singer (2010) variants for longitudinal data or for functional data
  - CART for chemometrics in Questier et al. (2005)

Outline

1 Introduction

2 CART Trees

3 Random Forests

4 Variable Selection

5 An industrial application

6 Scaling RF for Big Data

Random Forests

- Introduced by Breiman (2001), they are part of the family of ensemble methods, see Dietterich (1999,2000), one can cite *Bagging, Boosting, Randomizing Outputs, Random Subspace*

- Machine learning algorithm, extremely powerful and successful, both for classification and regression problems. Increasingly used to process many real data in a wide range of applications:
  - Biochips Díaz-Uriarte and Alvarez De Andres (2006)
  - Ecology Prasad et al. (2006)
  - Forecasting pollution data Ghattas (1999)
  - Genomics Goldstein et al. (2010) and Boulesteix et al. (2012)
  - and for a larger survey, see Verikas et al. (2011)

- "Crowned" in Fernández-Delgado et al. (2014), they were absent from Wu et al. (2008) which mentions CART

Random Forests definition

$\mathcal{L}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ i.i.d. r.v. with the same distribution as (X, Y). $X \in \mathbb{R}^p$ (input variables), $Y \in \mathcal{Y}$ (response variable)
$\mathcal{Y} = \mathbb{R}$ regression and $\mathcal{Y} = \{1, \ldots, L\}$ classification
Aim: build a predictor $\widehat{h} : \mathbb{R}^p \to \mathcal{Y}$

Definition: Random Forests (Breiman 2001)

$\left\{\widehat{h}(., \Theta_\ell), 1 \leq \ell \leq q\right\}$ tree-predictor collection, $(\Theta_\ell)_{1 \leq \ell \leq q}$ i.i.d. r.v. independent with $\mathcal{L}_n$.

Random forests predictor $\widehat{h}$ obtained by aggregating the collection of trees.

- $\widehat{h}(x) = \dfrac{1}{q} \displaystyle\sum_{\ell=1}^{q} \widehat{h}(x, \Theta_\ell)$ regression
- $\widehat{h}(x) = \underset{1 \leq c \leq L}{\operatorname{argmax}} \displaystyle\sum_{\ell=1}^{q} 1_{\widehat{h}(x, \Theta_\ell) = c}$ classification

Random Forests: a schema

$$\begin{tikzcd}
 & & \mathcal{L}_n \arrow[dll, "\text{Arbre aléatoire}"'] \arrow[d] \arrow[drr] & & \\
\widehat{h}(., \Theta_1) \arrow[drr, "\text{Agrégation}"'] & \cdots\cdots\cdots\cdots\cdots & \widehat{h}(., \Theta_l) \arrow[d] & \cdots\cdots\cdots\cdots\cdots & \widehat{h}(., \Theta_q) \arrow[dll] \\
 & & \widehat{h}(.) & &
\end{tikzcd}$$

Bagging (Breiman 1996)

$$\begin{tikzcd}
 & & \mathcal{L}_n \arrow[dll, "\text{Bootstrap}"'] \arrow[d] \arrow[drr] & & \\
\mathcal{L}_n^{\Theta_1} \arrow[d, "\text{CART}"'] & \cdots\cdots\cdots\cdots & \mathcal{L}_n^{\Theta_l} \arrow[d] & \cdots\cdots\cdots\cdots & \mathcal{L}_n^{\Theta_q} \arrow[d] \\
\widehat{h}(.,\Theta_1) \arrow[drr, "\text{Agrégation}"'] & \cdots\cdots\cdots & \widehat{h}(.,\Theta_l) \arrow[d] & \cdots\cdots\cdots & \widehat{h}(.,\Theta_q) \arrow[dll] \\
 & & \widehat{h}_{BAG}(.) & &
\end{tikzcd}$$

Instability of CART ⇒ performance improvement

Random Forests-Random Inputs (Breiman 2001)

Definition: RI-tree

We define a RI-tree as the variant of CART consisting to select at random, at each node, `mtry` variables, and split using only the selected variables.

`mtry` is the same for all nodes of all trees in the forest.

Definition: Random Forests-RI

A Random Forests-RI is obtained by doing Bagging with RI-trees.

Random Forests-RI: a schema

$$\begin{tikzcd}
& \mathcal{L}_n \arrow[dl, "\text{Bootstrap}"'] \arrow[d] \arrow[dr] & \\
\mathcal{L}_n^{\Theta_1} \arrow[r, dotted, no head] \arrow[d, "\text{RI Tree}"'] & \mathcal{L}_n^{\Theta_l} \arrow[r, dotted, no head] \arrow[d] & \mathcal{L}_n^{\Theta_q} \arrow[d] \\
\widehat{h}(.,\Theta_1,\Theta'_1) \arrow[r, dotted, no head] \arrow[dr, "\text{Agreggation}"'] & \widehat{h}(.,\Theta_l,\Theta'_l) \arrow[r, dotted, no head] \arrow[d] & \widehat{h}(.,\Theta_q,\Theta'_q) \arrow[dl] \\
& \widehat{h}_{RF-RI}(.) &
\end{tikzcd}$$

Additional randomness ⇒ increase of efficiency

Random Forest-RI in practice

R package randomForest:

- based on the initial FORTRAN code of Breiman, Cutler (2000)
- well described in Liaw, Wiener (2002)

Main parameters of the randomForest procedure:

- ntree: number of trees in the forest (default = 500)
- mtry: number of variables randomly selected at each node
  - by default: $\sqrt{p}$ for classification, $p/3$ for regression
  - the empirical study Genuer et al. (2008) points out:
    - In regression, except for calculation time, no significant improvement compared to unpruned Bagging ($mtry = p$)
    - For standard classification problem, the default value is correct
    - but for high-dimensional classification ones, larger values for *mtry* sometimes give much better results

spam dataset: optimal tree, Bagging, RF

Predictor	optimal tree	bagging	RF
Tets error	0.086	0.060	0.052

Table: Bagging and random forest test errors, compared to optimal tree error on `spam` dataset

- Bagging using also the package `randomForest` and by constructing a Bagging predictor with as a basic rule an unpruned CART tree (the package does not allow to prune the trees of a forest)
- RF built using the package `randomForest` with default values

Family of Random Forests

Examples of additional randomness:

- resampling prior to the construction of the trees,
- random selection of the split variable at each node,
- random selection of the cut-off point at each node.

Two main families of random forests:

- Classical: partition optimized on the learning data $\mathcal{L}_n$
- Purely random: randomly chosen partition, independently of $\mathcal{L}_n$

Variants and ... theoretical results

Definition: Purely Random Forests (PRF)

A PRF is an aggregation of purely random trees, if the partition associated with each of these trees is drawn randomly independently of $\mathcal{L}_n$

- PRF in theory:
  - Breiman (2000), Biau et al. (2008), Zhu et al. (2015), Ishwaran, Kogalur (2010), Denil et al. (2014): consistency
  - Genuer (2012): variance reduction result, convergence rate in dim. 1. And then Arlot, Genuer (2014) in dim. d
  - Biau (2012): result of reduction of variance and bias in a context of reduction of dimension
  - Mentch, Hooker (2014), Wager (2014): asymptotic normality
- PRF in practice:
  - Cutler, Zhao (2001), Geurts et al. (2006), Duroux et al. (2016)

Variants and ... theoretical results (2)

- Scornet, Biau, Vert (2015): consistency for the Breiman's RF (for additive models)
- Recent paper Biau, Scornet (2016): excellent survey of theoretical work + discussion

- In this discussion, Arlot, Genuer (2016) study the contribution of RF randomness ingredients, theoretically for a simple variant of RF and by simulation for a variant close to RF-RI
  - It appears that the randomization of partitions (obtained by the bootstrap, the drawing of m variables at each node or the drawing of the cut-point) would be the most crucial
  - This explains why the Bagging (which does not randomize the selection of the cut-point) and Extra Random Trees of Geurts et al. (2006) (which does not use bootstrap) give very satisfactory results in practice, although very different in the choice of the additional hazard Θ

Extensions and variants

- Extensions for various objectives:
  - Ranking Forests Clemençon et al. (2013)
  - Survival Forests Hothorn et al. (2006), Ishwaran et al. (2008)
  - Quantile regression Meinshausen (2006)
  - Cluster forests Yan et al. (2013), Afanador et al. (2016)
- Variants:
  - LOFB-DRF aims to improve the diversity of the trees of the RF, Fawagreh et al. (2015) use Local Outlier Factor (LOF) to identify the diverse trees and select those corresponding to largest LOF-score
  - Reweighting *a posteriori* the trees to improve predictive performance, Winham et al. (2013)
  - Random Forests-RC (RC for ”random combination”) use splits non necessarily parallel to the axes, already introduced in Breiman (2001), and more recently considered in Blaser, Frizlewicz (2015), Menze et al. (2011)
  - A recent neuronal variant of RF, see Biau et al. 2016

OOB Error and Prediction error estimation

OOB Error, Out Of Bag (≈ "Out Of Bootstrap")

To predict Y_i, we only aggregate the predictors $\widehat{h}(., \Theta_\ell)$ built on bootstrap samples not containing (X_i, Y_i)

- OOB Error $= \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} \left(Y_i - \widehat{Y}_i \right)^2$ in regression
- OOB Error $= \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} \mathbb{1}_{Y_i \neq \widehat{Y}_i}$ in classification

- Estimation similar to the classical estimators of generalization error, using test sample or cross-validation
- No prior splitting of the learning sample is needed, included in the generation of bootstrap samples
- But attention: it is indeed a different sub-forest (in general) that is used to calculate each $\widehat{Y}_i$

Variable Importance

- Beyond the performance and the quasi-automatic tuning of RF, one of the most important aspects from the applied side is the quantification of variable importance
- Azen et Budescu (2003): for a general discussion about this notion
- Notion sparsely studied by statisticians and mainly in linear models, Grömping (2015) or the recent PhD thesis Wallard (2015)
- RF provide an ideal framework for estimating it:
  - a fully nonparametric method, without prescribing any particular form for the relation between Y and the components of X
  - a bootstrap resampling

  to have an effective and convenient definition of such indices

Variable Importance (2)

Breiman (2001), Strobl *et al.* (2007, 2008), Ishwaran (2007), Archer *et al.* (2008), Genuer et al. (2010), Gregorutti et al. (2013, 2015), Louppe et al. (2013)

Variable Importance (VI)

Soit $j \in \{1, \ldots, p\}$. For each OOB sample, we randomly permute the values of the jth variable from the data

Importance of the j-th variable

=

mean increase of the error of a tree after permutation

The greater the error increase, the more important the variable

“Toys data”, Weston *et al.* (2003)

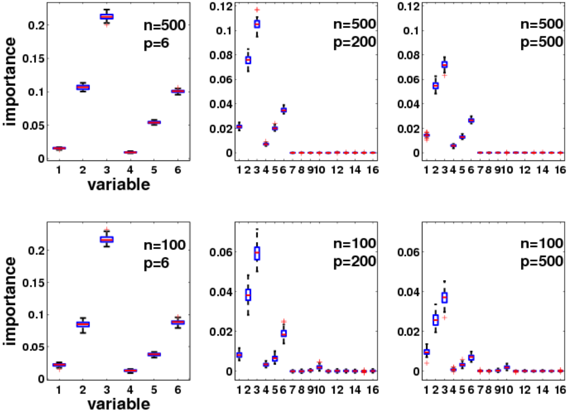
Two-class problem, $Y \in \{-1, 1\}$
6 true variables + noise variables:

- two independent groups of 3 significant variables (strongly, moderately and weakly correlated with the response), related to Y
- a group of noise variables, independent with Y

Model defined through the conditional distributions of the X^i conditionnally to $Y = y$:

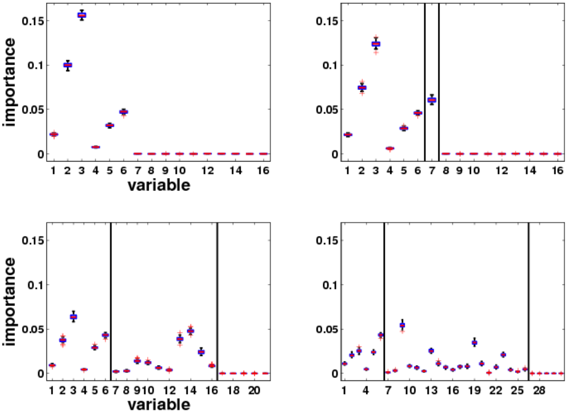
- for 70% of data, $X^i \sim y\mathcal{N}(i, 1)$ for $i = 1, 2, 3$ and $X^i \sim y\mathcal{N}(0, 1)$ for $i = 4, 5, 6$
- for the 30% left, $X^i \sim y\mathcal{N}(0, 1)$ for $i = 1, 2, 3$ and $X^i \sim y\mathcal{N}(i - 3, 1)$ for $i = 4, 5, 6$
- the other variables are noise, $X^i \sim \mathcal{N}(0, 1)$ for $i = 7, \dots, p$

VI sensitivity to *n* and *p*



Variability of VI is large for true variables with respect to useless ones

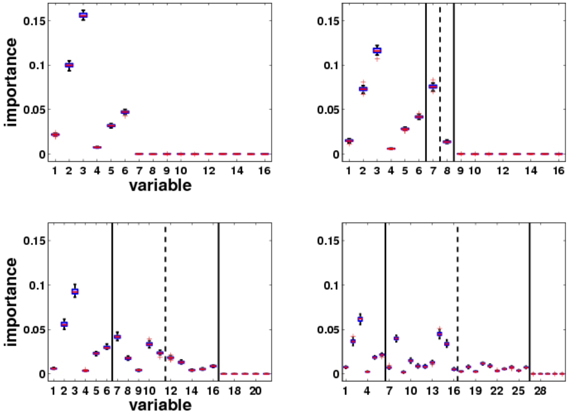
VI of a group of correlated variables



$\{1, 2, 3\}$ decreases with the number of replications of 3, $\{4, 5, 6\}$ unchanged

VI is not divided by the number of replications

VI of two groups of correlated variables



Two groups decrease when adding more replications of 3 and 6

Relative importance between two groups preserved

Variable Importance, *spam* dataset

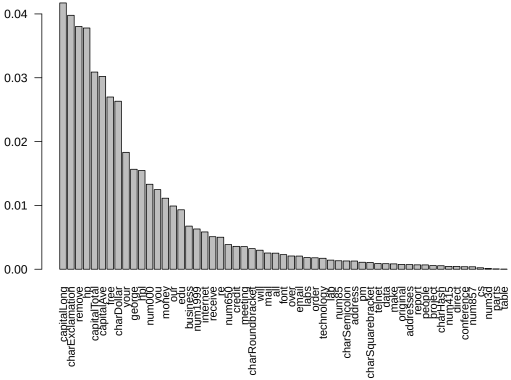


Figure: The 8 most important: The proportions of occurrences of the words or characters *remove*, *hp*, *$*, *!*, *free* as well as the 3 variables related to the lengths of the series of uppercase letters

Variable Importance, *spam* dataset

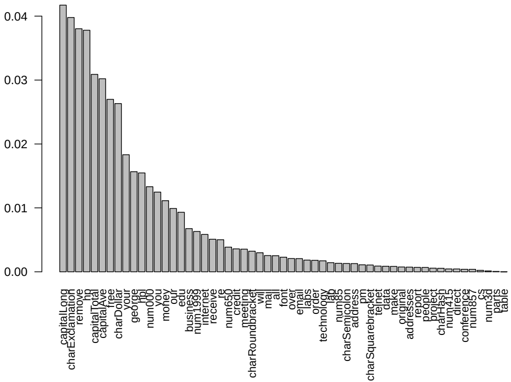


Figure: The variables of the first splits of the optimal CART tree are not at the top and the most important: *capitalLong* is not included

Outline

1 Introduction

2 CART Trees

3 Random Forests

4 Variable Selection

5 An industrial application

6 Scaling RF for Big Data

Variable Selection Procedure

Genuer, Poggi, Tuleau (2010), PRL et (2015), R Journal

We distinguish two different objectives:

1. to select all important variables, even with high redundancy, for interpretation purpose
2. to find a sufficient parsimonious set of important variables for prediction

Our aim is to build an automatic procedure, which fulfills these two objectives

Let us simply mention two previous works which have inspired our proposal:

- Díaz-Uriarte, Alvarez de Andrés (2006)
- Ben Ishak, Ghattas (2008)

Selection(s) on *toys*

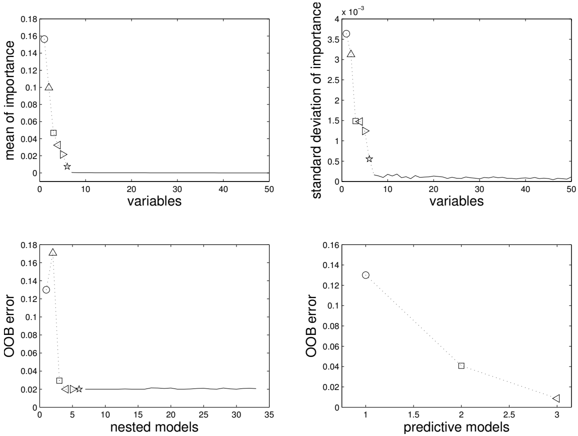


Figure: Variable selection procedure for interpretation and prediction: toys data $n = 100$, $p = 200$

- True variables (1 to 6) represented by $(\triangleright, \triangle, \circ, \star, \triangleleft, \Box)$
- VI based on 50 forests with $ntree = 2000$, $mtry = 100$

Variable selection procedure: Ranking

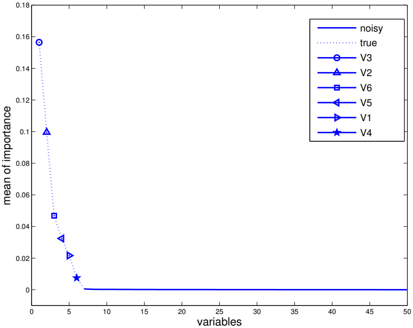


Figure: *Ranking by sorting the VI in descending order*

- Graph for the 50 most important variables (the other noisy variables having an importance very close to zero too)

- True variables are significantly more important than the noisy ones

Variable selection procedure: Elimination

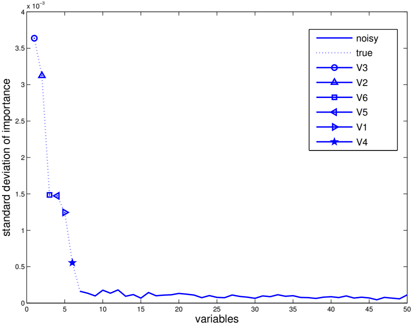


Figure: *Consider corresponding standard deviations of VI to estimate a threshold and keep variables of importance exceeding this level*

- Threshold = min of the prediction value given by a CART model fitting this curve (conservative in general)

- True variables standard deviation large w.r.t. the noisy variables one, which is close to zero

- The selected threshold leads to retain 33 variables

Variable selection for interpretation

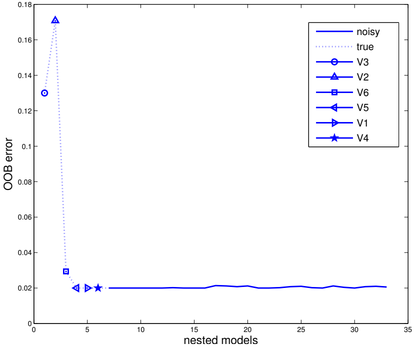


Figure: *Compute OOB error rates of RF for the nested models and select the variables of the model leading to the smallest OOB error*

- Error decreases quickly and reaches its minimum when the first 4 true variables are included in the model, then it remains *almost* constant

- The model containing 4 of the 6 true variables is selected. In fact, the actual minimum is reached for 24 variables but we use a rule similar to the 1 SE rule of Breiman *et al.* (1984) used for cost-complexity selection

Variable selection for prediction

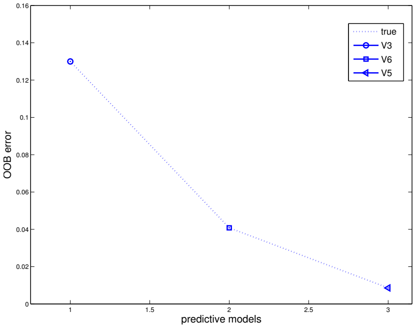
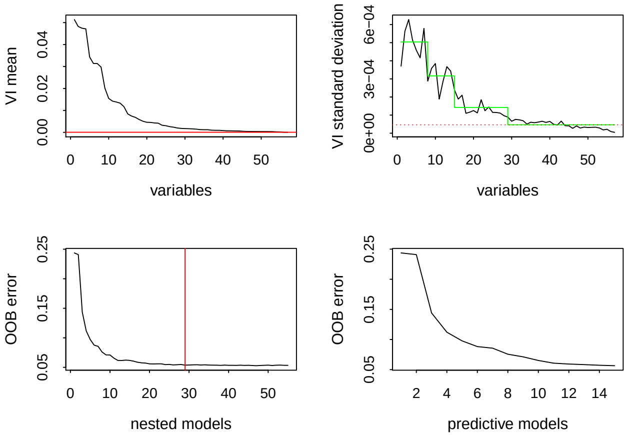


Figure: *Sequential variable introduction with testing*

- A variable is added only if the error gain exceeds a threshold since the error decrease must be significantly greater than the average variation obtained by adding noisy variables

- Final prediction model involves only variables 3, 6 and 5

VSURF applied to spam dataset



Forest	Initial	interpretation	Prediction
Test error	0.052	0.056	0.060

An application: Brain fMRI data

Genuer, Michel, Eger, Thirion (2010)

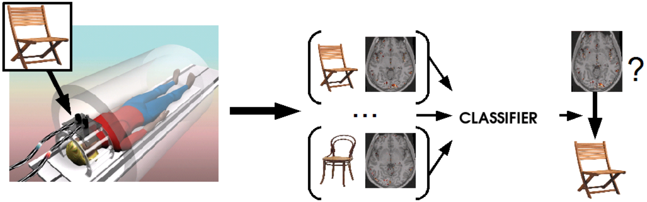


Figure: Experimental framework, fMRI

12 individuals: 4 kinds of chair (4 classes), raw data are made of 100 000 voxels (variables), 72 observations.
Preliminary step: A parcellisation obtained by Ward algorithm reduces to 1000 parcels.

Classification $n = 72$ $p = 1000$ $L = 4$

Variable selection procedures for a real subject

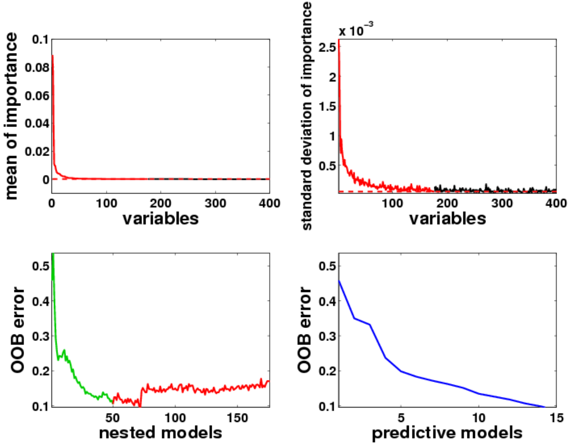


Figure: $ntree = 2000$, $mtry = p/3$
- Key point: it selects 176 variables after the threshold step, 50 variables for interpretation, and 15 variables for prediction (very much smaller than $p = 1000$)

The results

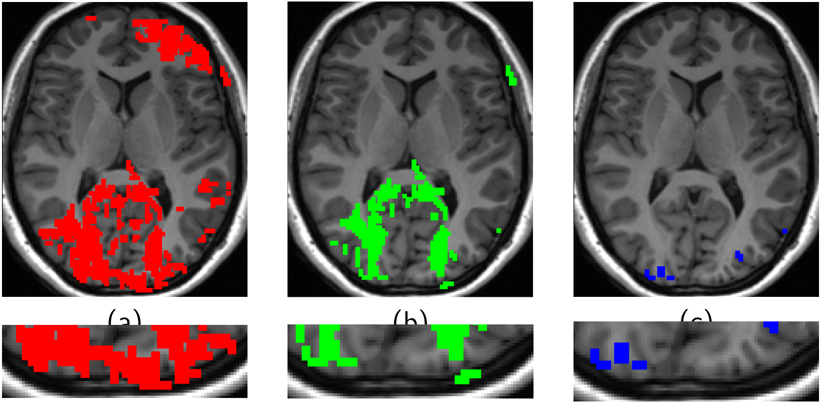


Figure: Example of the different steps of the framework on a real subject. (a) Elimination Step (b) Interpretation Step (c) Prediction Step

A 3D view of a final result

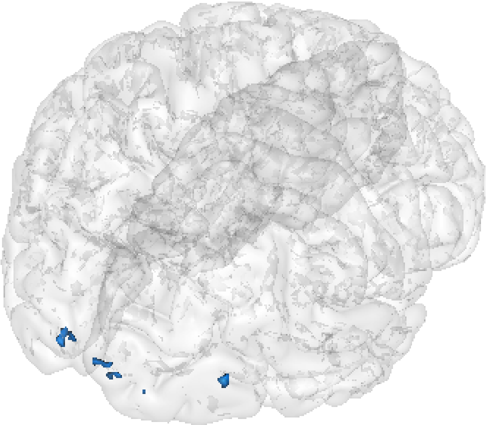


Figure: Selected regions for at least 3 subjects among 12 for the last step of the proedure

A final comparison

	Initial	Elim.	Interp.	Pred.	Reference
Erreur	34 %	29 %	27 %	30 %	31 %
Nombre var.	1000	146	23	8	350

Figure: Results on the 12 subjects of the study

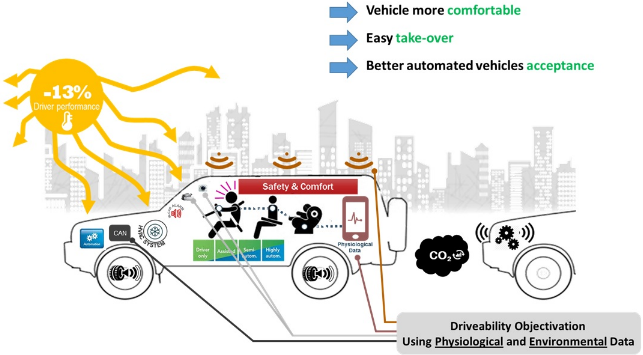
- Reference method: linear SVM (F-test + cross-validation)
- Comparable error rate
- Many fewer variables

Outline

1. Introduction
2. CART Trees
3. Random Forests
4. Variable Selection
5. **An industrial application**
6. Scaling RF for Big Data

An industrial application: Physiological variable selection for driver's stress level classification[1]

Context: Safety and end-user acceptance of road automation in smart cities



- Vehicle more comfortable
- Easy take-over
- Better automated vehicles acceptance

Driveability Objectivation
Using Physiological and Environmental Data

[1] RF-based approach for physiological functional variable selection for driver's stress level classification El Haouij, Poggi, Ghozi, Sevestre Ghalila, Jaïdane, slides from talks ENBIS16 Sheffield, JdS2017 Avignon , SIS 2017 Florence

Electrodermal Activity EDA, ECG, Heart Rate HR

- **Electrodermal Activity (EDA)** measures the autonomic nervous system changes in the electrical properties of the skin, Has been used as a **measure of stress** in anticipatory anxiety

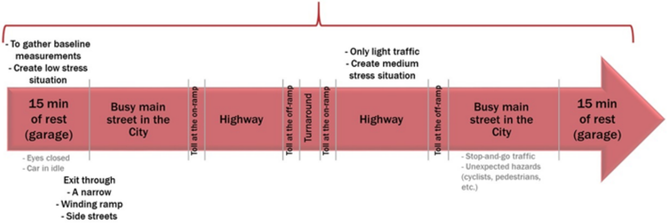
- **Blood Volume Pulse (BVP) and Electrocardiogram (ECG)** are used to measure heart activity, **heart rate (HR)** and vasoconstriction

Respiration (RESP) and Electromyogram (EMG)

- Capturing breathing activity by recording **chest cavity expansion** is a measure of **RESP**
- **EMG** measures muscle activity by detecting **surface voltages** that occur when **a muscle is contracted**

Driving protocol of "drivedb[2]" and data collection (Healey 2000)

From 50 min to 1,5h (depending on traffic conditions)
In midmorning or midafternoun



- To gather baseline measurements
- Create low stress situation

15 min of rest (garage)

- Eyes closed
- Car in idle

Exit through
- A narrow
- Winding ramp
- Side streets

Busy main street in the City

Toll at the on-ramp

Highway

Toll at the off-ramp

Turnaround

Toll at the on-ramp

- Only light traffic
- Create medium stress situation

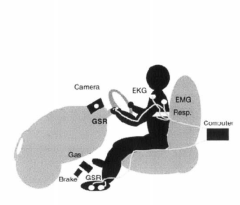
Highway

Toll at the off-ramp

Busy main street in the City

- Stop-and-go traffic
- Unexpected hazards (cyclists, pedestrians, etc.)

15 min of rest (garage)



[2] http://physionet.org/physiobank/database/drivedb/

Objectives

From 10 available driving experiences:

- Provide a *physiological variables ranking* according to their importance in the stress level classification
- Automatic *selection of the most relevant variables* in classifying driver's stress level
- Recognize the stress with an accuracy *comparable* to the results of the *Expert-Based method*

In the future: *automatic extension* to other data and to other physical and physiological signals

For details, see El Haouij, Poggi, Ghozi, Sevestre-Ghalila, Jaidane *Random Forest-Based Approach for Physiological Functional Variable Selection: Towards Driver's Stress Level Classification*, Stat. Methods & Applications, 1-29, 2018

Cohort Description

Drive	Participant label	Date (mm-dd-yy)	Duration (hh:mm:ss)
1	M-3	07-28-99	1:24:15
2		08-04-99	1:20:46
3	M-4	07-15-99	1:28:38
4		08-05-99	1:21:11
5		08-13-99	1:10:52
6	F-8	08-02-99	1:21:16
7		08-05-99	1:21:13
8		08-06-99	1:23:04
9		08-09-99	1:17:38
10	Ind 4	07-16-99	1:04:57

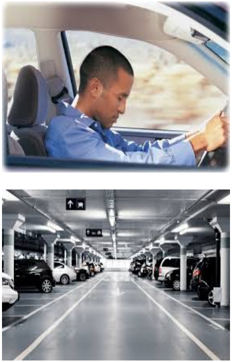
Starting Point[3]: Stress Level according to Subjective Studies

City

High stress level

Highway

Medium stress level

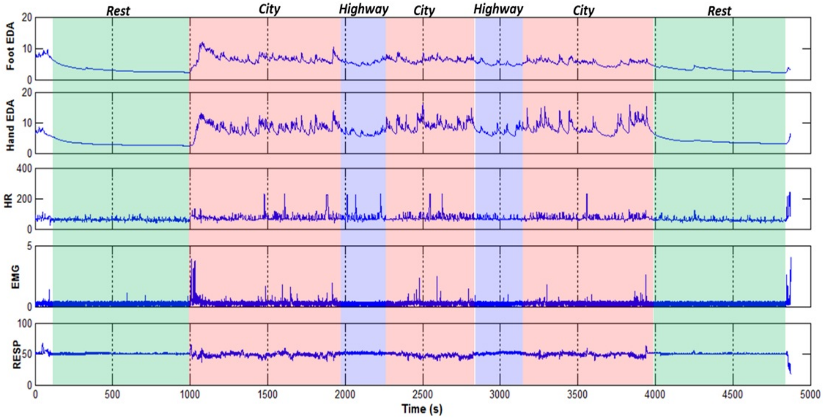


Rest

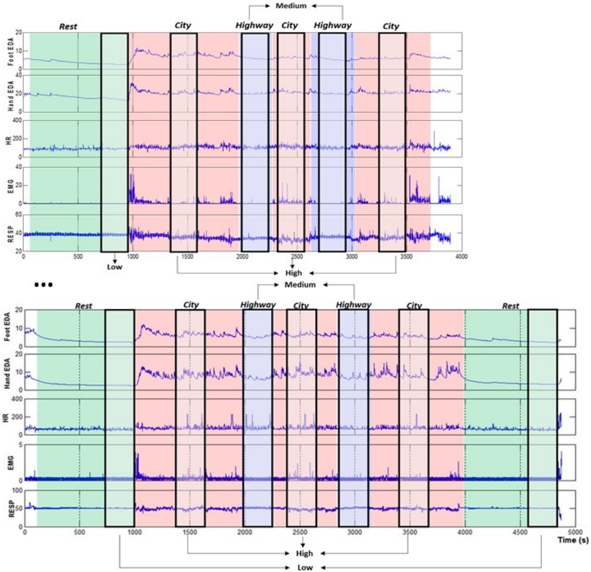
Low stress level

[3] Healey A. et Picard R. (2005). Detecting Stress During Real-World Driving Tasks Using Physiological Sensors. IEEE Trans.on Intelligent Transportation Systems

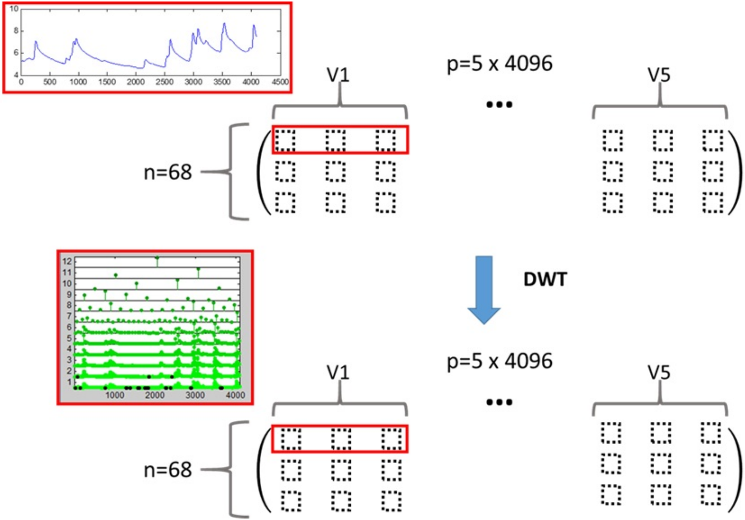
Data illustration: 1 drive



Data Description: Extraction



Data Description: Preprocessing



V1

p=5 x 4096

•••

V5

n=68

DWT

V1

p=5 x 4096

•••

V5

n=68

Model

- $\boldsymbol{S} = (S^1, ..., S^p) \in \mathcal{S}^p$: explanatory variables where $S^j = \{S^j(t) \in \mathbb{R}, t \in [0, T]\}$, $T = 4.40\ min$ where $j = 1..p, p = 5$
- Let i be the index of the drive segment, $i = 1..N, N = 68$
- For a given i, $\boldsymbol{S}_i(t)$ corresponds to the response variable, the stress level y_i

$$y_i = \left\{ \begin{array}{c} H = High\ stress\ level \\ M = Medium\ stress\ level \\ L = Low\ stress\ level \end{array} \right\}$$

- We aim to build a fully nonparametric random forests based estimator of the Bayes classifier $g : \mathcal{S}^p \to \{L, M, H\}$ minimizing the classification error $P(Y \neq g(\boldsymbol{S}))$

RF-RFE Recursive Feature Elimination algorithm[4]

Adapted from the SVM-RFE algorithm Guyon et al. (2002)

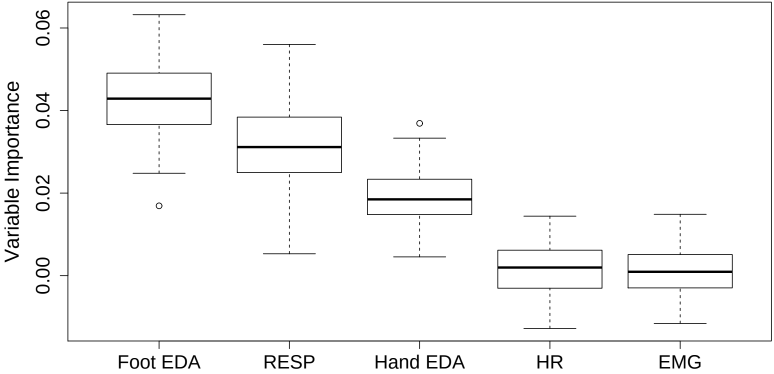
1. Split L into a training set L_T and a validation set L_V. Set $\mathcal{V}$=whole explanatory variables.
2. Fit a random forest model using L_T and considering $\mathcal{V}$
3. Compute the VI measure (respectively the grouped VI measure)
4. Compute the error using the validation sample L_V
5. Eliminate the least important variable (resp. group of variables) and update $\mathcal{V}$.
6. Repeat 2-5 until no further variables (group of variables) remain
7. Select the variables (resp. the groups of variables) involved in the model minimizing the prediction error

[4] Gregorutti et al. (2015)

Our procedure: Iterative RF-RFE

1. Wavelet decomposition of the physiological functional variables
2. Physiological Functional variable elimination: Repeat 10 times
    1. RF-RFE $(G(1), ..., G(p))$
    2. Compute a selection score for each group $G(j)$
    3. Eliminate the less relevant variables (those of a selection score below a threshold δ)
3. Wavelet Levels Selection: Repeat 10 times
    1. RF-RFE $(\{G(1, k_1), ..., G(J, k_1), ..., G(1, k_R), ..., G(J, k_R)\})$
    2. Compute a selection score for each group $G(w, k_R)$
    3. Eliminate the less relevant variables (those of a selection score below a threshold $\delta\prime$)

Physiological Variables Importance

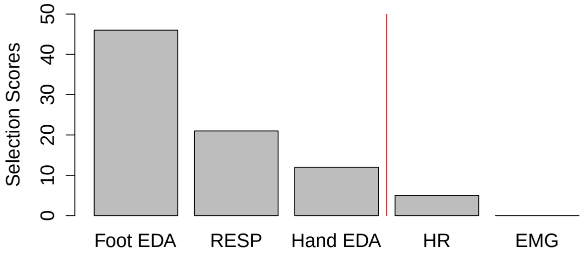


Iterative RF-RFE on the physiological functional variables

1	Foot EDA	RESP	Hand EDA	HR	EMG
2	HR	RESP	Hand EDA	Foot EDA	EMG
3	Foot EDA	Hand EDA	HR	EMG	RESP
4	Foot EDA	RESP	Hand EDA	EMG	HR
5	RESP	Foot EDA	Hand EDA	HR	EMG
6	Foot EDA	RESP	Hand EDA	EMG	HR
7	Foot EDA	Hand EDA	RESP	HR	EMG
8	Foot EDA	RESP	Hand EDA	HR	EMG
9	Foot EDA	Hand EDA	RESP	EMG	HR
10	Foot EDA	RESP	Hand EDA	HR	EMG

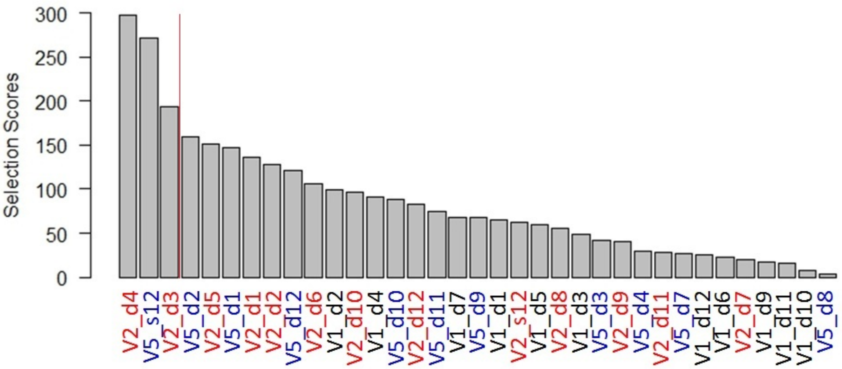
- Even if the number of selected variables varies:
- **Foot EDA is always selected**
- **EMG and HR** (except one) **never selected**

Iterative RF-RFE on the physiological functional variables

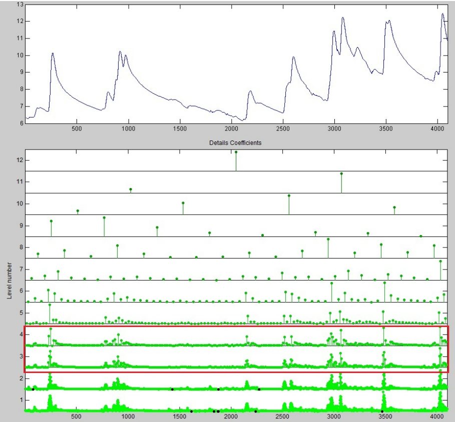


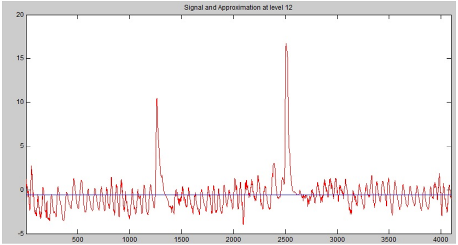
Iterative RF-RFE on the wavelet levels

V1: Hand EDA V2: Foot EDA V5:RESP



Wavelet Levels Selection, back to time domain





Outline

1. Introduction
2. CART Trees
3. Random Forests
4. Variable Selection
5. An industrial application
6. Scaling RF for Big Data

Motivation: Statistics in the Big Data World

For an introduction to statistics in Big Data, see
Jordan, *On statistics, computation and scalability*, Bernoulli, 2013

- Random Forests (RF):
  - Popular statistical machine learning method
  - Remarkable performance in a lot of applied problems
- Big Data (BD):
  - Massive, heterogeneous, streaming data
  - Major challenge to analyse such data

Our aim in Genuer, Poggi, Tuleau-Malot, Villa-Vialaneix, *Random Forests for Big Data*, published in Big Data Research (2017) is to provide a review of the different proposals of RFBD, some experiments on massive datasets, remarks and extensions

Big Data characteristics

- The three V (highlighted by Gartner, Inc.):
  - Volume: massive data
  - Velocity: data stream
  - Variety: heterogeneous data

- We focus
  - mainly on the Volume characteristic: (at least) data are so large that they cannot be stored on one single computer
  - and additionally on the Velocity issue at the end of the talk

Strategies for scaling RF to Big Data

- Subsampling: choose a tractable subset of data, perform a classical analysis on it, and repeat this several times (e.g. Bag of Little Bootstraps, Kleiner et.al. 2012)

- Divide and Conquer: split the data into a lot of tractable subsamples, apply classical analysis on each of them, and combine the collection of results (e.g. MapReduce framework)

- Sequential Updating for Stream Data: conduct an analysis in an online manner, by updating quantities along data arrival (e.g. Schifano et.al. 2014)

See Wang et.al. 2015 for an introduction

The reference scheme: standard RF (**seqRF**)

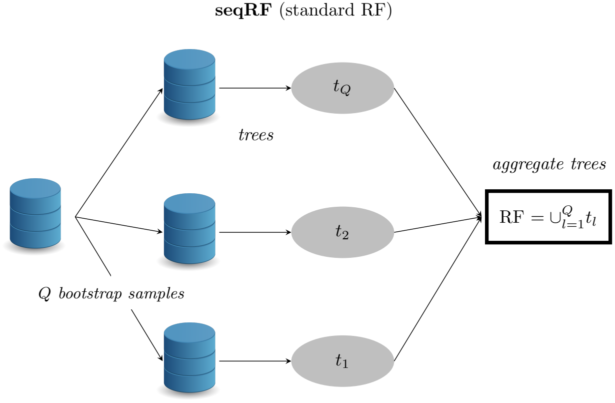
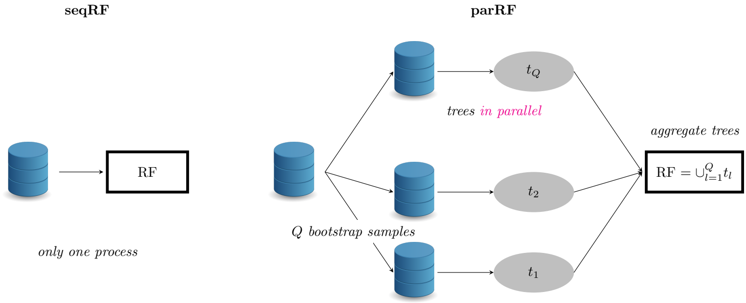


Figure: starting from the dataset, generate bootstrap samples (draw randomly n observations with replacement from L) and learn corresponding randomized binary trees. Finally aggregate them

Parallel implementations of standard RF (**parRF**)

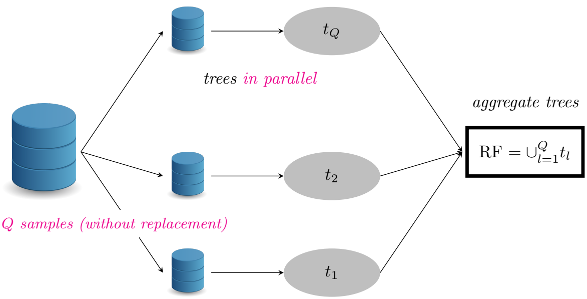


- Sequential (left) and parallel (right) implementations of the standard RF algorithm. The trees are learned in parallel (individually or by groups) and RF is the final random forest with Q trees

m-out-of-n RF (**moonRF**)

- Q samples without replacement, with m observations out of n, are randomly built in parallel and a tree is learned from each of these samples. The Q trees are then aggregated to obtain a final RF with Q trees

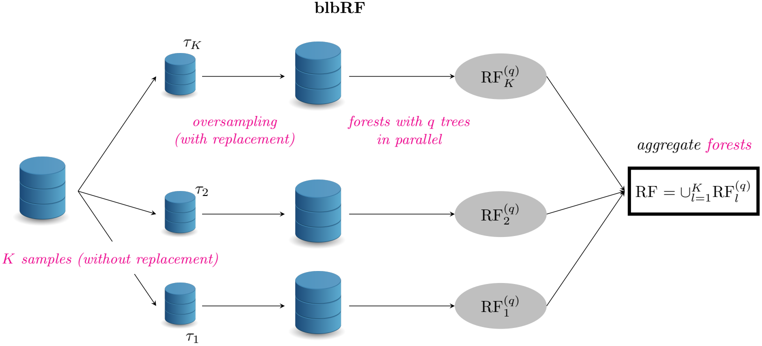
moonRF



t_Q
trees in parallel
aggregate trees
$\mathrm{RF} = \cup_{l=1}^{Q} t_l$
t_2
Q samples (without replacement)
t_1

Bag of Little Bootstraps RF (**blbRF**)

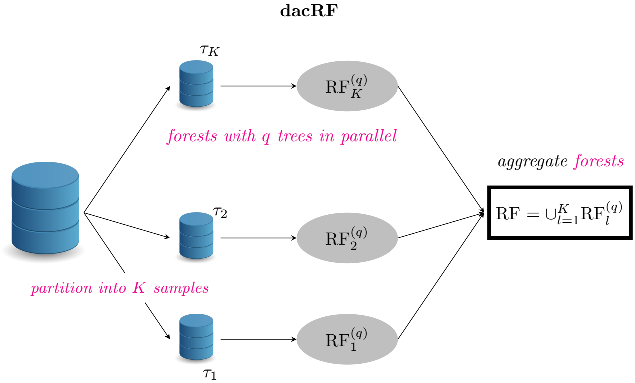
- A subsampling step, performed K times in parallel, is followed by an oversampling step which aims at building q trees for each subsample, all obtained from a bootstrap sample of size n of the original data, but each containing only $m << n$ different observations



blbRF
τ_K
τ_2
τ_1
oversampling (with replacement)
forests with q trees in parallel
aggregate forests
$\mathrm{RF}_K^{(q)}$
$\mathrm{RF}_2^{(q)}$
$\mathrm{RF}_1^{(q)}$
$\mathrm{RF} = \cup_{l=1}^K \mathrm{RF}_l^{(q)}$
K samples (without replacement)

Divide-and-conquer RF (**dacRF**)

- The original dataset is partitioned into K subsets. A RF with q trees is built from each of the partition's subsets and all the forests are finally aggregated in a final random forest, RF



Online RF (**onRF**)

- Handle data streams (data arrive sequentially) in an online manner (no memory of past data): Saffari et al. 2009
- Can deal with massive data streams (addressing both Volume and Velocity characteristics), but also massive (static) data, by running through the data sequentially
- In depth adaptation of Breiman's RF: even the tree growing mechanism is changed.
  - Main idea: think only in terms of proportions of output classes, instead of observations + online bagging
- ERT is used instead of the original Breiman's RF, because it allows for a faster update of the RF
  - candidate splits (*variable, value*) are randomly drawn for each node, and the best split is computed only among those splits
- Consistency results in Denil et al. 2013 for a variant

Thanks for your attention

- A freely accessible reference, in French but with full of references:
  Robin Genuer, Jean-Michel Poggi, *Arbres CART et Forêts aléatoires, Importance et sélection de variables*, 45 pages, 2017 [a]
  http://up5.fr/hal-01387654v2
- *Les forêts aléatoires avec R*
  Genuer, Poggi (2019)
  Presses Universitaires de Rennes (PUR)

- Left image credit: *Marc Varachaud*, original creation

[a] book chapter of "Apprentissage Statistique et Données Massives", Technip, p. 295-342, 2018