Variable Selection Using Random Forests
The VSURF R package

Robin Genuer[a], Jean-Michel Poggi[b], Christine Tuleau-Malot[c]

[a]Université de Bordeaux, [b]Université d'Orsay, [c]Université de Nice

27 juin 2014
Rencontres R 2014, Montpellier

Random Forests

- introduced by Breiman (2001)
- ensemble methods family Dietterich (1999, 2000)
- very efficient algorithm of statistical learning, for both classification and regression problems.

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

$X = (X^1, ..., X^p) \in \mathbb{R}^p$ (input variables)
$Y \in \mathcal{Y}$ (response variable)

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

Goal : 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 agreggating the collection of trees.

Agreggation :

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

Tree : piece-wise constant predictor, obtained by a recursive dyadic partitioning of $\mathbb{R}^p$.

Restriction : splits parallel to axes.

Typically, at each step of the partitioning, we seek the "best" split of the data $\mathcal{L}_n$.

Example : CART, Breiman et.al. (1984).

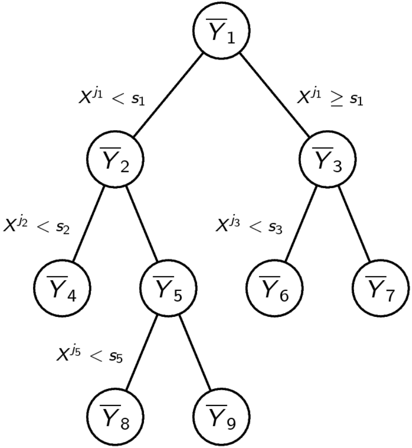


Figure : Regression tree

Bagging (Breiman 1996)

$$\begin{tikzcd}
& \mathcal{L}_n \arrow[ld, "\text{Bootstrap}"'] \arrow[d] \arrow[rd] & \\
\mathcal{L}_n^{\Theta_1} \arrow[r, dotted, no head] \arrow[d, "\text{CART}"'] & \mathcal{L}_n^{\Theta_l} \arrow[r, dotted, no head] \arrow[d] & \mathcal{L}_n^{\Theta_q} \arrow[d] \\
\widehat{h}(.,\Theta_1) \arrow[r, dotted, no head] \arrow[rd, "\text{Agrégation}"'] & \widehat{h}(.,\Theta_l) \arrow[r, dotted, no head] \arrow[d] & \widehat{h}(.,\Theta_q) \arrow[ld] \\
& \widehat{h}_{BAG}(.) &
\end{tikzcd}$$

CART instability ⇒ increase of efficiency

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

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

Additional randomness ⇒ increase of efficiency

Random Forests-RI

R package `randomForest`:

- based on the initial 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 (default = $\sqrt{p}$)

Prediction estimator error

OOB = Out Of Bag (≈ "Out Of Bootstrap")

OOB error

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

⇒ OOB error :

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

Variable importance

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

Definition: Variable importance (VI)

Let $j \in \{1, \ldots, p\}$. For each OOB sample we permute at random the j-th variable values of the data.

Variable importance of the j-th variable $=$ mean increase of the error of a tree after permutation.

The more the error increases, the more important is the variable.

1 Introduction
- Definition
- Examples

2 Variable selection
- Procedure
- Application

Variable Selection

Genuer, Poggi, Tuleau (2010)

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

One earlier work must be cited: Díaz-Uriarte, Alvarez de Andrés (2006).

SRBCT

A high dimensional classification dataset, available in `mixOmics` package

$$n = 63 \quad p = 2308$$

- Input variables : gene expressions
- Output variable : class tumour of each sample (4 classes)

```r
library(VSURF)
library(mixOmics)
data(srbct)
```

Application

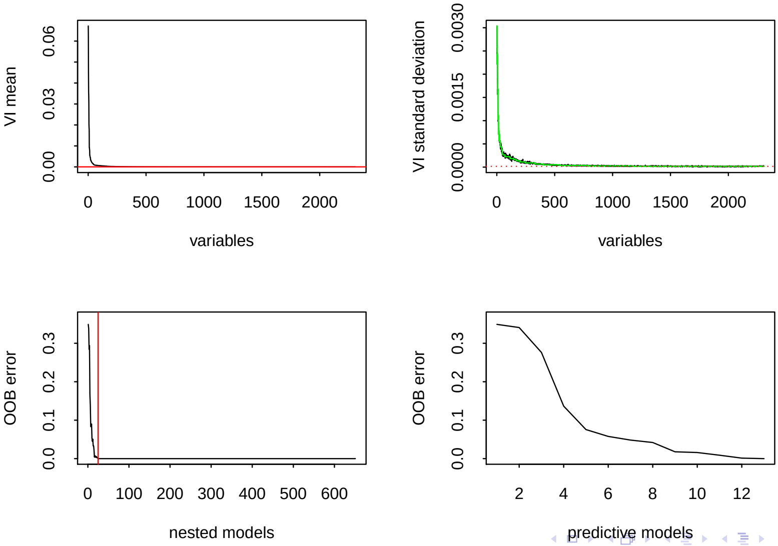
```
vSRBCT <- VSURF(x = srbct$gene, y = srbct$class)
```

```
summary(vSRBCT)

##
##  VSURF computation time: 2.7 hours
##
##  VSURF selected:
##  651 variables at thresholding step (in 7.2 mins)
##  25 variables at interpretation step (in 2.6 hours)
##  13 variables at prediction step (in 14.6 secs)
```

Application

```
plot(vSRBCT)
```



Computational remarks

The first step keeps obviously too many variables for this example. Increasing the `nmin` parameter significantly reduces the overall computation time:

```
vSRBCT.nmin10 <- VSURF(x = srbct$gene, y = srbct$class,
    nmin = 10)
```

```
summary(vSRBCT.nmin10)

##
##  VSURF computation time: 25.3 mins
##
##  VSURF selected:
##  233 variables at thresholding step (in 6.3 mins)
##  25 variables at interpretation step (in 18.7 mins)
##  14 variables at prediction step (in 13.6 secs)
```

Computational remarks

To reduce computational time, we can also use the parallel version of VSURF on one computer (my laptop):

```
vSRBCT.laptop <- VSURF.parallel(x = srbct$gene, y = srbct$class,
    nmin = 10)
```

```
summary(vSRBCT.laptop)

##
##  VSURF computation time: 11.3 mins
##
##  VSURF selected:
##  233 variables at thresholding step (in 3 mins)
##  25 variables at interpretation step (in 8.1 mins)
##  14 variables at prediction step (in 14 secs)
##
##  VSURF ran in parallel on a PSOCK cluster and used 3 cores
```

Computational remarks

Or on a cluster of several computers:

```
vSRBCT.cluster <- VSURF.parallel(x = srbct$gene, y = srbct$class,
    nmin = 10, clusterType = "MPI", ncores = 50)
```

```
summary(vSRBCT.cluster)

##
##  VSURF computation time: 1.4 mins
##
##  VSURF selected:
##  228 variables at thresholding step (in 32.2 secs)
##  25 variables at interpretation step (in 38.3 secs)
##  16 variables at prediction step (in 16 secs)
##
##  VSURF ran in parallel on a MPI cluster and used 50 cores
```

Ozone

`Ozone` : A standard regression dataset, from `mlbench` package.

$$n = 366 \quad p = 12$$

Input variables

`V1`	Month	`V8`	Temperature (Sandburg)
`V2`	Day of month	`V9`	Temperature (El Monte)
`V3`	Day of week	`V10`	Inversion base height
`V5`	Pressure height	`V11`	Pressure gradient
`V6`	Wind speed	`V12`	Inversion base temperature
`V7`	Humidity	`V13`	Visibility

Output variable

`V4` Daily maximum one-hour-average ozone

Application

```
library(VSURF)
library(mlbench)
data(Ozone)
```

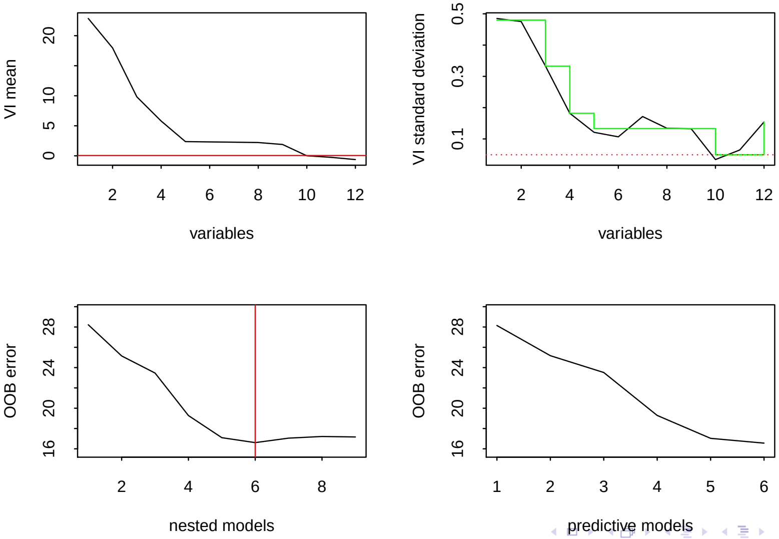
```
vozone <- VSURF(formula = V4~., data = Ozone,
                     na.action = na.omit)
```

```
summary(vozone)

##
##  VSURF computation time: 1.7 mins
##
##  VSURF selected:
##  9 variables at thresholding step (in 57.3 secs)
##  6 variables at interpretation step (in 28 secs)
##  6 variables at prediction step (in 17.4 secs)
```

Application

```
plot(vozone)
```



Concluding Remarks

- Variable selection procedure fully data-driven
- Can be applied for both classification and regression problems involving both standard and high-dimensional datasets
- Handles mixed data (categorical and continuous input variables) and missing data
- R package available on CRAN (still in development). We invite you to test it:

```r
install.packages("VSURF")
```

(mailto:Robin.Genuer@isped.u-bordeaux2.fr)

Short bibliography

- Breiman, L., Friedman J., Olshen R., Stone C. *Classification And Regression Trees*. Chapman & Hall (1984)

- Breiman, L. *Bagging*. Machine Learning (1996)

- Breiman, L. *Random Forests*. Machine Learning (2001)

- Díaz-Uriarte R., Alvarez de Andrés S. *Gene Selection and classification of microarray data using random forest*. BMC Bioinformatics (2006)

- Genuer R., Poggi J.-M. and Tuleau-Malot C. *Variable selection using random forests*. Pattern Recognition Letters (2010)

- Genuer R., Poggi J.-M. and Tuleau-Malot C. *VSURF: An R Package for Variable Selection Using Random Forests* (submitted)