Variable Selection Using Random Forests
The VSURF R package

Robin Genuer\textsuperscript{a}, Jean-Michel Poggi\textsuperscript{b}, Christine Tuleau-Malot\textsuperscript{c}

\textsuperscript{a}Université de Bordeaux, \textsuperscript{b}Université d’Orsay, \textsuperscript{c}Université de Nice

27 juin 2014
Rencontres R 2014, Montpellier
Random Forests

- introduced by Breiman (2001)
- very efficient algorithm of statistical learning, for both classification and regression problems.

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

\[ X = (X^1, \ldots, X^p) \in \mathbb{R}^p \text{ (input variables)} \]
\[ Y \in \mathcal{Y} \text{ (response variable)} \]
- \( \mathcal{Y} = \mathbb{R} : \text{regression} \)
- \( \mathcal{Y} = \{1, \ldots, L\} : \text{classification} \)

**Goal**: build a predictor \( \hat{h} : \mathbb{R}^p \rightarrow \mathcal{Y} \).
Definition: Random Forests (Breiman 2001)

\[ \{ \hat{h}(., \Theta_\ell), 1 \leq \ell \leq q \} \] tree-predictor collection, \((\Theta_\ell)_{1 \leq \ell \leq q}\) i.i.d. r.v. independent with \(L_n\).

Random forests predictor \(\hat{h}\) obtained by aggregating the collection of trees.

Agregation:

- \(\hat{h}(x) = \frac{1}{q} \sum_{\ell=1}^{q} \hat{h}(x, \Theta_\ell)\) regression

- \(\hat{h}(x) = \arg\max_{1 \leq c \leq L} \sum_{\ell=1}^{q} 1_{\hat{h}(x, \Theta_\ell) = c}\) 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 $\mathcal{L}_n$.


Figure: Regression tree
Bagging (Breiman 1996)

CART instability $\Rightarrow$ 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

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

\[ \text{OOB} = \text{Out Of Bag} \ (\approx \ "\text{Out Of Bootstrap}\") \]

**OOB error**

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

\[ \Rightarrow \text{OOB error} : \]

- \[ \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \] regression

- \[ \frac{1}{n} \sum_{i=1}^{n} 1_{Y_i \neq \hat{Y}_i} \] classification
**Variable importance**


**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 $= \text{mean increase of the error of a tree after permutation.}$

*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).
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)
```
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)
```r
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:

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

```r
summary(vSRBCT.nmin10)
```

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

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

```r
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
```
Or on a cluster of several computers:

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

\[ n = 366 \quad p = 12 \]

**Input variables**

<table>
<thead>
<tr>
<th>V1</th>
<th>Month</th>
<th>V8</th>
<th>Temperature (Sandburg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>V2</td>
<td>Day of month</td>
<td>V9</td>
<td>Temperature (El Monte)</td>
</tr>
<tr>
<td>V3</td>
<td>Day of week</td>
<td>V10</td>
<td>Inversion base height</td>
</tr>
<tr>
<td>V5</td>
<td>Pressure height</td>
<td>V11</td>
<td>Pressure gradient</td>
</tr>
<tr>
<td>V6</td>
<td>Wind speed</td>
<td>V12</td>
<td>Inversion base temperature</td>
</tr>
<tr>
<td>V7</td>
<td>Humidity</td>
<td>V13</td>
<td>Visibility</td>
</tr>
</tbody>
</table>

**Output variable**

| V4   | Daily maximum one-hour-average ozone |
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)
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:

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

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


