

Why and how to use random forest variable importance measures (and how you shouldn't)

Carolin Strobl (LMU München) and Achim Zeileis (WU Wien)

carolin.strobl@stat.uni-muenchen.de

useR! 2008, Dortmund

Introduction

Construction

R functions

Variable
importance

Tests for variable
importance

Conditional
importance

Summary

References

Introduction

Random forests

Introduction

Construction

R functions

Variable importance

Tests for variable
importance

Conditional
importance

Summary

References

Introduction

Random forests

- ▶ have become increasingly popular in, e.g., genetics and the neurosciences

Introduction

Construction

R functions

Variable importance

Tests for variable
importance

Conditional
importance

Summary

References

Introduction

Random forests

- ▶ have become increasingly popular in, e.g., genetics and the neurosciences [imagine a long list of references here]

Introduction

Construction

R functions

Variable importance

Tests for variable
importance

Conditional
importance

Summary

References

Introduction

Random forests

- ▶ have become increasingly popular in, e.g., genetics and the neurosciences [imagine a long list of references here]
- ▶ can deal with “small n large p”-problems, high-order interactions, correlated predictor variables

Introduction

Construction

R functions

Variable importance

Tests for variable importance

Conditional importance

Summary

References

Introduction

Random forests

- ▶ have become increasingly popular in, e.g., genetics and the neurosciences [imagine a long list of references here]
- ▶ can deal with “small n large p”-problems, high-order interactions, correlated predictor variables
- ▶ are used not only for prediction, but also to assess variable importance

Introduction

Construction

R functions

Variable importance

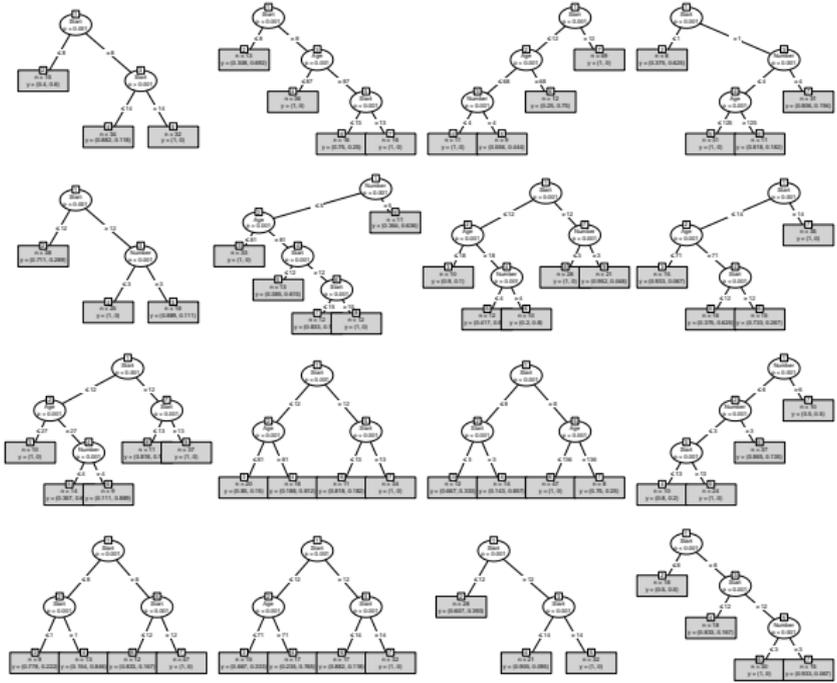
Tests for variable
importance

Conditional
importance

Summary

References

(Small) random forest



Introduction

Construction

R functions

Variable importance

Tests for variable importance

Conditional importance

Summary

References

Construction of a random forest

Introduction

Construction

R functions

Variable

importance

Tests for variable
importance

Conditional
importance

Summary

References

Construction of a random forest

- ▶ draw `ntree` bootstrap samples from original sample

Introduction

Construction

R functions

Variable
importance

Tests for variable
importance

Conditional
importance

Summary

References

Construction of a random forest

- ▶ draw `ntree` bootstrap samples from original sample
- ▶ fit a classification tree to each bootstrap sample
⇒ `ntree` trees

Introduction

Construction

R functions

Variable
importance

Tests for variable
importance

Conditional
importance

Summary

References

Construction of a random forest

- ▶ draw `ntree` bootstrap samples from original sample
- ▶ fit a classification tree to each bootstrap sample
⇒ `ntree` trees
- ▶ creates diverse set of trees because
 - ▶ trees are instable w.r.t. changes in learning data
⇒ `ntree` different looking trees (bagging)
 - ▶ randomly preselect `mtry` splitting variables in each split
⇒ `ntree` more different looking trees (random forest)

Introduction

Construction

R functions

Variable
importance

Tests for variable
importance

Conditional
importance

Summary

References

Random forests in R

- ▶ `randomForest` (pkg: `randomForest`)
 - ▶ reference implementation based on CART trees (Breiman, 2001; Liaw and Wiener, 2008)
 - for variables of different types: biased in favor of continuous variables and variables with many categories (Strobl, Boulesteix, Zeileis, and Hothorn, 2007)
- ▶ `cforest` (pkg: `party`)
 - ▶ based on unbiased conditional inference trees (Hothorn, Hornik, and Zeileis, 2006)
 - + for variables of different types: unbiased when subsampling, instead of bootstrap sampling, is used (Strobl, Boulesteix, Zeileis, and Hothorn, 2007)

Introduction

Construction

R functions

Variable

importance

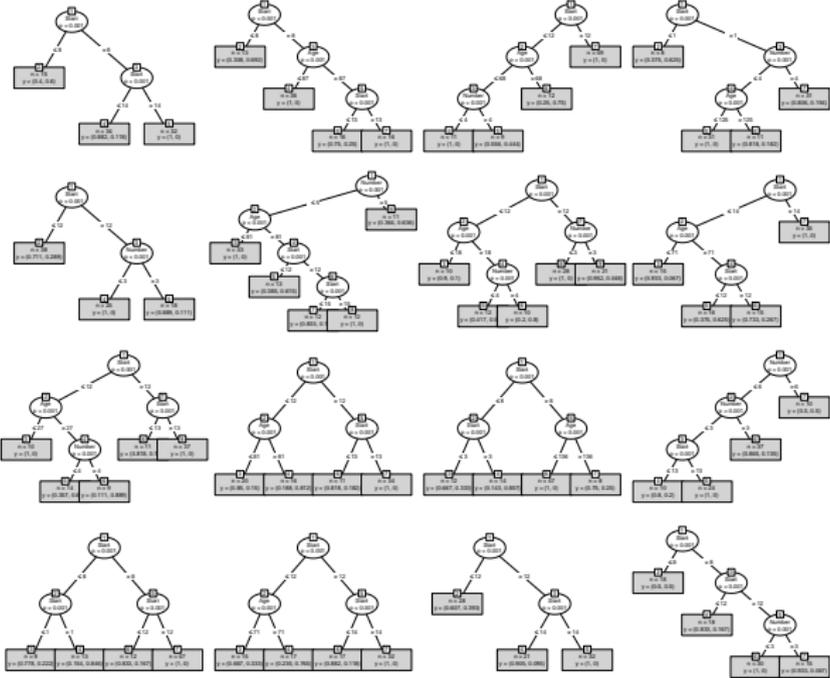
Tests for variable
importance

Conditional
importance

Summary

References

(Small) random forest



Introduction

Construction

R functions

Variable importance

Tests for variable importance

Conditional importance

Summary

References

Measuring variable importance

- ▶ Gini importance

mean Gini gain produced by X_j over all trees

- ▶

```
obj <- randomForest(..., importance=TRUE)
obj$importance      column: MeanDecreaseGini
importance(obj, type=2)
```

for variables of different types: biased in favor of continuous variables and variables with many categories

Introduction

Construction

R functions

**Variable
importance**

Tests for variable
importance

Conditional
importance

Summary

References

Measuring variable importance

- ▶ permutation importance

mean decrease in classification accuracy after permuting X_j over all trees

- ▶ `obj <- randomForest(..., importance=TRUE)`
`obj$importance` column: MeanDecreaseAccuracy
`importance(obj, type=1)`
- ▶ `obj <- cforest(...)`
`varimp(obj)`

for variables of different types: unbiased only when subsampling is used as in `cforest(..., controls = cforest_unbiased())`

Introduction

Construction

R functions

Variable importance

Tests for variable importance

Conditional importance

Summary

References

The permutation importance

within each tree t

$$VI^{(t)}(\mathbf{x}_j) = \frac{\sum_{i \in \overline{\mathfrak{B}}^{(t)}} I(y_i = \hat{y}_i^{(t)})}{|\overline{\mathfrak{B}}^{(t)}|} - \frac{\sum_{i \in \overline{\mathfrak{B}}^{(t)}} I(y_i = \hat{y}_{i,\pi_j}^{(t)})}{|\overline{\mathfrak{B}}^{(t)}|}$$

$\hat{y}_i^{(t)} = f^{(t)}(\mathbf{x}_i)$ = predicted class before permuting

$\hat{y}_{i,\pi_j}^{(t)} = f^{(t)}(\mathbf{x}_{i,\pi_j})$ = predicted class after permuting X_j

$\mathbf{x}_{i,\pi_j} = (x_{i,1}, \dots, x_{i,j-1}, x_{\pi_j(i),j}, x_{i,j+1}, \dots, x_{i,p})$

Note: $VI^{(t)}(\mathbf{x}_j) = 0$ by definition, if X_j is not in tree t

Introduction

Construction

R functions

Variable
importance

Tests for variable
importance

Conditional
importance

Summary

References

The permutation importance

over all trees:

1. raw importance

$$VI(\mathbf{x}_j) = \frac{\sum_{t=1}^{ntree} VI^{(t)}(\mathbf{x}_j)}{ntree}$$

```
▶ obj <- randomForest(..., importance=TRUE)
  importance(obj, type=1, scale=FALSE)
```

Introduction

Construction

R functions

**Variable
importance**

Tests for variable
importance

Conditional
importance

Summary

References

The permutation importance

over all trees:

2. scaled importance (z-score)

$$\frac{VI(\mathbf{x}_j)}{\frac{\hat{\sigma}}{\sqrt{ntree}}} = z_j$$

```
▶ obj <- randomForest(..., importance=TRUE)
  importance(obj, type=1, scale=TRUE) (default)
```

Introduction

Construction

R functions

**Variable
importance**

Tests for variable
importance

Conditional
importance

Summary

References

Tests for variable importance

for variable selection purposes

Introduction

Construction

R functions

Variable
importance

**Tests for variable
importance**

Conditional
importance

Summary

References

Tests for variable importance

for variable selection purposes

- ▶ Breiman and Cutler (2008): simple significance test based on normality of z-score

`randomForest, scale=TRUE` + α -quantile of $N(0,1)$

Introduction

Construction

R functions

Variable
importance

Tests for variable
importance

Conditional
importance

Summary

References

Tests for variable importance

for variable selection purposes

- ▶ Breiman and Cutler (2008): simple significance test based on normality of z-score

`randomForest`, `scale=TRUE` + α -quantile of $N(0,1)$

- ▶ Diaz-Uriarte and Alvarez de Andrés (2006): backward elimination (throw out least important variables until out-of-bag prediction accuracy drops)

`varSelRF` (pkg: `varSelRF`), dep. on `randomForest`

Introduction

Construction

R functions

Variable
importance

Tests for variable
importance

Conditional
importance

Summary

References

Tests for variable importance

for variable selection purposes

- ▶ Breiman and Cutler (2008): simple significance test based on normality of z-score
`randomForest`, `scale=TRUE` + α -quantile of $N(0,1)$
- ▶ Diaz-Uriarte and Alvarez de Andrés (2006): backward elimination (throw out least important variables until out-of-bag prediction accuracy drops)
`varSelRF` (pkg: `varSelRF`), dep. on `randomForest`
- ▶ Diaz-Uriarte (2007) and Rodenburg et al. (2008): plots and significance test (randomly permute response values to mimic the overall null hypothesis that none of the predictor variables is relevant = baseline)

Introduction

Construction

R functions

Variable
importance

Tests for variable
importance

Conditional
importance

Summary

References

Tests for variable importance

problems of these approaches:

Introduction

Construction

R functions

Variable
importance

**Tests for variable
importance**

Conditional
importance

Summary

References

Tests for variable importance

problems of these approaches:

- ▶ (at least) Breiman and Cutler (2008): strange statistical properties (Strobl and Zeileis, 2008)

Introduction

Construction

R functions

Variable
importance

Tests for variable
importance

Conditional
importance

Summary

References

Tests for variable importance

problems of these approaches:

- ▶ (at least) Breiman and Cutler (2008): strange statistical properties (Strobl and Zeileis, 2008)
- ▶ all: preference of correlated predictor variables (see also Nicodemus and Shugar, 2007; Archer and Kimes, 2008)

Introduction

Construction

R functions

Variable
importance

Tests for variable
importance

Conditional
importance

Summary

References

Breiman and Cutler's test

under the null hypothesis of zero importance:

$$z_j \stackrel{as.}{\sim} N(0,1)$$

if z_j exceeds the α -quantile of $N(0,1) \Rightarrow$ reject the null hypothesis of zero importance for variable X_j

Introduction

Construction

R functions

Variable

importance

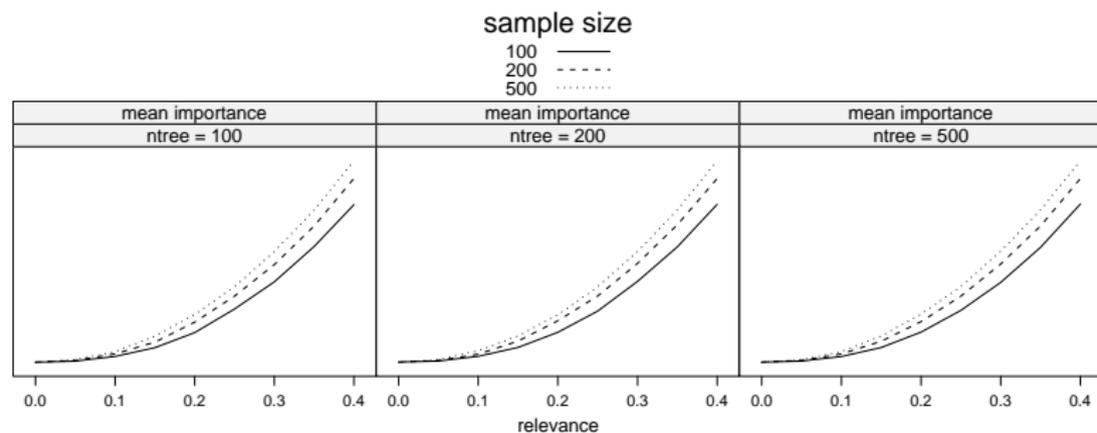
Tests for variable
importance

Conditional
importance

Summary

References

Raw importance



Introduction

Construction

R functions

Variable
importance

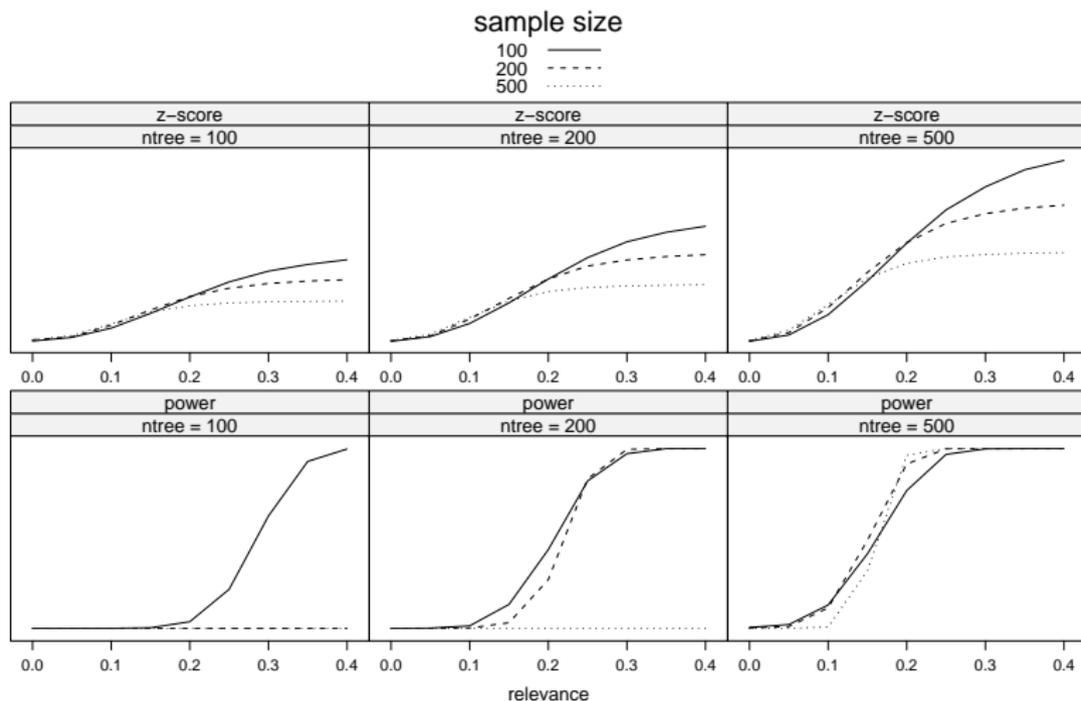
Tests for variable
importance

Conditional
importance

Summary

References

z-score and power



Introduction

Construction

R functions

Variable
importance

Tests for variable
importance

Conditional
importance

Summary

References

Findings

z-score and power

- ▶ increase in `n`
- ▶ decrease in sample size

⇒ rather use raw, unscaled permutation importance!

```
importance(obj, type=1, scale=FALSE)
```

```
varimp(obj)
```

Introduction

Construction

R functions

Variable

importance

Tests for variable
importance

Conditional
importance

Summary

References

What null hypothesis were we testing in the first place?

<i>obs</i>	<i>Y</i>	<i>X_j</i>	<i>Z</i>
1	<i>y</i> ₁	<i>x</i> _{$\pi_j(1),j$}	<i>z</i> ₁
⋮	⋮	⋮	⋮
<i>i</i>	<i>y</i> _{<i>i</i>}	<i>x</i> _{$\pi_j(i),j$}	<i>z</i> _{<i>i</i>}
⋮	⋮	⋮	⋮
<i>n</i>	<i>y</i> _{<i>n</i>}	<i>x</i> _{$\pi_j(n),j$}	<i>z</i> _{<i>n</i>}

$$H_0 : X_j \perp Y, Z \text{ or } X_j \perp Y \wedge X_j \perp Z$$

$$P(Y, X_j, Z) \stackrel{H_0}{=} P(Y, Z) \cdot P(X_j)$$

Introduction

Construction

R functions

Variable
importance

Tests for variable
importance

Conditional
importance

Summary

References

What null hypothesis were we testing in the first place?

the current null hypothesis reflects independence of X_j from both Y and the remaining predictor variables Z

Introduction

Construction

R functions

Variable

importance

Tests for variable
importance

Conditional
importance

Summary

References

What null hypothesis were we testing in the first place?

the current null hypothesis reflects independence of X_j from both Y and the remaining predictor variables Z

⇒ a high variable importance can result from violation of either one!

Introduction

Construction

R functions

Variable importance

Tests for variable importance

Conditional importance

Summary

References

Suggestion: Conditional permutation scheme

<i>obs</i>	<i>Y</i>	<i>X_j</i>	<i>Z</i>
1	<i>y</i> ₁	$X_{\pi_{j Z=a}(1),j}$	$z_1 = a$
3	<i>y</i> ₃	$X_{\pi_{j Z=a}(3),j}$	$z_3 = a$
27	<i>y</i> ₂₇	$X_{\pi_{j Z=a}(27),j}$	$z_{27} = a$
6	<i>y</i> ₆	$X_{\pi_{j Z=b}(6),j}$	$z_6 = b$
14	<i>y</i> ₁₄	$X_{\pi_{j Z=b}(14),j}$	$z_{14} = b$
33	<i>y</i> ₃₃	$X_{\pi_{j Z=b}(33),j}$	$z_{33} = b$
⋮	⋮	⋮	⋮

$$H_0 : X_j \perp Y | Z$$

$$P(Y, X_j | Z) \stackrel{H_0}{=} P(Y | Z) \cdot P(X_j | Z)$$

$$\text{or } P(Y | X_j, Z) \stackrel{H_0}{=} P(Y | Z)$$

Introduction

Construction

R functions

Variable
importance

Tests for variable
importance

Conditional
importance

Summary

References

Technically

- ▶ use any partition of the feature space for conditioning

Introduction

Construction

R functions

Variable
importance

Tests for variable
importance

**Conditional
importance**

Summary

References

Technically

- ▶ use any partition of the feature space for conditioning
- ▶ here: use binary partition already learned by tree
(use cutpoints as bisectors of feature space)

Introduction

Construction

R functions

Variable
importance

Tests for variable
importance

Conditional
importance

Summary

References

Technically

- ▶ use any partition of the feature space for conditioning
- ▶ here: use binary partition already learned by tree
(use cutpoints as bisectors of feature space)
- ▶ condition on correlated variables or select some

Introduction

Construction

R functions

Variable
importance

Tests for variable
importance

Conditional
importance

Summary

References

Technically

- ▶ use any partition of the feature space for conditioning
- ▶ here: use binary partition already learned by tree
(use cutpoints as bisectors of feature space)
- ▶ condition on correlated variables or select some

Strobl et al. (2008)

available in `cforest` from version 0.9-994: `varimp(obj, conditional = TRUE)`

Introduction

Construction

R functions

Variable
importance

Tests for variable
importance

Conditional
importance

Summary

References

Simulation study

- ▶ dgp: $y_i = \beta_1 \cdot x_{i,1} + \dots + \beta_{12} \cdot x_{i,12} + \varepsilon_i$, $\varepsilon_i \stackrel{i.i.d.}{\sim} N(0, 0.5)$
- ▶ $X_1, \dots, X_{12} \sim N(0, \Sigma)$

$$\Sigma = \begin{pmatrix} 1 & 0.9 & 0.9 & 0.9 & 0 & \dots & 0 \\ 0.9 & 1 & 0.9 & 0.9 & 0 & \dots & 0 \\ 0.9 & 0.9 & 1 & 0.9 & 0 & \dots & 0 \\ 0.9 & 0.9 & 0.9 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

X_j	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	\dots	X_{12}
β_j	5	5	2	0	-5	-5	-2	0	\dots	0

Introduction

Construction

R functions

Variable
importance

Tests for variable
importance

Conditional
importance

Summary

References

Summary

Introduction

Construction

R functions

Variable

importance

Tests for variable
importance

Conditional
importance

Summary

References

Summary

if your predictor variables are of different types:

use `cforest` (pkg: `party`) with default option `controls = cforest_unbiased()`

with permutation importance `varimp(obj)`

Introduction

Construction

R functions

Variable

importance

Tests for variable
importance

Conditional
importance

Summary

References

Summary

if your predictor variables are of different types:

use `cforest` (pkg: `party`) with default option `controls = cforest_unbiased()`

with permutation importance `varimp(obj)`

otherwise: feel free to use `cforest` (pkg: `party`)

with permutation importance `varimp(obj)`

or `randomForest` (pkg: `randomForest`)

with permutation importance `importance(obj, type=1)`

or Gini importance `importance(obj, type=2)`

but don't fall for the z-score! (i.e. set `scale=FALSE`)

Introduction

Construction

R functions

Variable

importance

Tests for variable
importance

Conditional
importance

Summary

References

Summary

if your predictor variables are of different types:

use `cforest` (pkg: `party`) with default option `controls = cforest_unbiased()`

with permutation importance `varimp(obj)`

otherwise: feel free to use `cforest` (pkg: `party`)

with permutation importance `varimp(obj)`

or `randomForest` (pkg: `randomForest`)

with permutation importance `importance(obj, type=1)`

or Gini importance `importance(obj, type=2)`

but don't fall for the z-score! (i.e. set `scale=FALSE`)

if your predictor variables are highly correlated: use the

conditional importance in `cforest` (pkg: `party`)

Introduction

Construction

R functions

Variable

importance

Tests for variable
importance

Conditional
importance

Summary

References



Introduction

Construction

R functions

Variable importance

Tests for variable importance

Conditional importance

Summary

References

Archer, K. J. and R. V. Kimes (2008). Empirical characterization of random forest variable importance measures. *Computational Statistics & Data Analysis* 52(4), 2249–2260.

Breiman, L. (2001). Random forests. *Machine Learning* 45(1), 5–32.

Breiman, L. and A. Cutler (2008). Random forests – Classification manual. Website accessed in 1/2008;
<http://www.math.usu.edu/~adele/forests>.

Breiman, L., A. Cutler, A. Liaw, and M. Wiener (2006). *Breiman and Cutler's Random Forests for Classification and Regression*. R package version 4.5-16.

Diaz-Uriarte, R. (2007). GeneSrf and varselrf: A web-based tool and R package for gene selection and classification using random forest. *BMC Bioinformatics* 8:328.

Introduction

Construction

R functions

Variable importance

Tests for variable importance

Conditional importance

Summary

References

- Hothorn, T., K. Hornik, and A. Zeileis (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics* 15(3), 651–674.
- Strobl, C., A.-L. Boulesteix, A. Zeileis, and T. Hothorn (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics* 8:25.
- Strobl, C. and A. Zeileis (2008). Danger: High power! – exploring the statistical properties of a test for random forest variable importance. In *Proceedings of the 18th International Conference on Computational Statistics, Porto, Portugal*.
- Strobl, C., A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis (2008). Conditional variable importance for random forests. *BMC Bioinformatics* 9:307.

Introduction

Construction

R functions

Variable importance

Tests for variable importance

Conditional importance

Summary

References