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
Random forests
Random forests

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

Random forests

- have become increasingly popular in, e.g., genetics and the neurosciences [imagine a long list of references here]
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
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
(Small) random forest
Construction of a random forest

Construction of a random forest involves the following steps:

1. Draw `ntree` bootstrap samples from the original sample.
2. Fit a classification tree to each bootstrap sample.
3. Create a diverse set of trees because the trees are unstable with respect to changes in the learning data.
4. Randomly preselect `mtry` splitting variables in each split.
5. Create `ntree` more different looking trees (random forest).

The process results in a set of trees that are combined to make predictions and interpret variable importance.
Construction of a random forest

- draw \texttt{ntree} bootstrap samples from original sample
Construction of a random forest

- draw $n_{tree}$ bootstrap samples from original sample
- fit a classification tree to each bootstrap sample
  \[ \Rightarrow n_{tree} \text{ trees} \]
Construction of a random forest

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

- Gini importance
  mean Gini gain produced by $X_j$ over all trees
  
  ```r
  obj <- randomForest(..., importance=TRUE)
  obj$importance
  ```
  column: MeanDecreaseGini
  
  ```r
  importance(obj, type=2)
  ```

for variables of different types: biased in favor of continuous variables and variables with many categories
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())`
The permutation importance within each tree \( t \)

\[
VI^{(t)}(x_j) = \frac{\sum_{i \in \mathcal{B}^{(t)}} I \left( y_i = \hat{y}_i^{(t)} \right)}{|\mathcal{B}^{(t)}|} - \frac{\sum_{i \in \mathcal{B}^{(t)}} I \left( y_i = \hat{y}_{i,\pi_j}^{(t)} \right)}{|\mathcal{B}^{(t)}|}
\]

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

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

\( x_{i,\pi_j} = (x_{i,1}, \ldots, x_{i,j-1}, x_{\pi_j(i),j}, x_{i,j+1}, \ldots, x_{i,p}) \)

Note: \( VI^{(t)}(x_j) = 0 \) by definition, if \( X_j \) is not in tree \( t \)
The permutation importance

over all trees:

1. raw importance

\[ \text{VI}(x_j) = \frac{\sum_{t=1}^{n_{\text{tree}}} \text{VI}^{(t)}(x_j)}{n_{\text{tree}}} \]

▶ \text{obj} <- \text{randomForest(..., importance=TRUE)}

\text{importance(obj, type=1, scale=FALSE)}
The permutation importance

over all trees:

2. scaled importance (z-score)

\[
\frac{\hat{Vl}(x_j)}{\hat{\sigma}} = z_j \sqrt{\frac{\sigma}{ntree}}
\]

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

for variable selection purposes

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

$\text{randomForest}$, scale=TRUE + $\alpha$-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)

$\text{varSelRF}$ (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)
Tests for variable importance

for variable selection purposes

- Breiman and Cutler (2008): simple significance test based on normality of z-score
  \texttt{randomForest, scale=TRUE } + \alpha\text{-quantile of } \mathcal{N}(0,1)
Tests for variable importance

for variable selection purposes

- Breiman and Cutler (2008): simple significance test based on normality of z-score
  \texttt{randomForest}, \texttt{scale=TRUE} + $\alpha$-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)
  \texttt{varSelRF} (pkg: \textit{varSelRF}), dep. on \textit{randomForest}
Tests for variable importance

for variable selection purposes

- Breiman and Cutler (2008): simple significance test based on normality of z-score
  \texttt{randomForest, scale=TRUE} + \alpha\text{-quantile of } \text{N}(0,1)

- Diaz-Uriarte and Alvarez de Andrés (2006): backward elimination (throw out least important variables until out-of-bag prediction accuracy drops)
  \texttt{varSelRF (pkg: varSelRF)}, dep. on \texttt{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 \(\equiv\) baseline)
Tests for variable importance

problems of these approaches:
Tests for variable importance

problems of these approaches:

▶ (at least) Breiman and Cutler (2008): strange statistical properties (Strobl and Zeileis, 2008)
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 Shugart, 2007; Archer and Kimes, 2008)
Breiman and Cutler’s test

under the null hypothesis of zero importance:

\[ z_j \overset{as.}{\sim} N(0, 1) \]

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

sample size

<table>
<thead>
<tr>
<th>mean importance</th>
<th>mean importance</th>
<th>mean importance</th>
</tr>
</thead>
<tbody>
<tr>
<td>ntree = 100</td>
<td>ntree = 200</td>
<td>ntree = 500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>relevance</th>
<th>relevance</th>
<th>relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>0.3</td>
<td>0.4</td>
<td>0.5</td>
</tr>
</tbody>
</table>

sample size

<table>
<thead>
<tr>
<th>mean importance</th>
<th>mean importance</th>
<th>mean importance</th>
</tr>
</thead>
<tbody>
<tr>
<td>ntree = 100</td>
<td>ntree = 200</td>
<td>ntree = 500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>relevance</th>
<th>relevance</th>
<th>relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>0.3</td>
<td>0.4</td>
<td>0.5</td>
</tr>
</tbody>
</table>
z-score and power

<table>
<thead>
<tr>
<th>Sample size</th>
<th>z-score</th>
<th>z-score</th>
<th>z-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ntree = 100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ntree = 200</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ntree = 500</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Power</th>
<th>Power</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>ntree = 100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ntree = 200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ntree = 500</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Findings

z-score and power

- increase in \texttt{ntree}
- decrease in sample size

⇒ rather use raw, unscaled permutation importance!

\texttt{importance(obj, type=1, scale=FALSE)}

\texttt{varimp(obj)}
What null hypothesis were we testing in the first place?

<table>
<thead>
<tr>
<th>obs</th>
<th>Y</th>
<th>$X_j$</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$y_1$</td>
<td>$x_{\pi_j(1).j}$</td>
<td>$z_1$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$i$</td>
<td>$y_i$</td>
<td>$x_{\pi_j(i).j}$</td>
<td>$z_i$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$n$</td>
<td>$y_n$</td>
<td>$x_{\pi_j(n).j}$</td>
<td>$z_n$</td>
</tr>
</tbody>
</table>

$H_0: X_j \perp Y, Z$ or $X_j \perp Y \land X_j \perp Z$

$$P(Y, X_j, Z) \overset{H_0}{=} P(Y, Z) \cdot P(X_j)$$
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 $Z$
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!
Suggestion: Conditional permutation scheme

<table>
<thead>
<tr>
<th>obs</th>
<th>Y</th>
<th>$X_j$</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$y_1$</td>
<td>$x_{\pi j</td>
<td>Z = a(1), j}$</td>
</tr>
<tr>
<td>3</td>
<td>$y_3$</td>
<td>$x_{\pi j</td>
<td>Z = a(3), j}$</td>
</tr>
<tr>
<td>27</td>
<td>$y_{27}$</td>
<td>$x_{\pi j</td>
<td>Z = a(27), j}$</td>
</tr>
<tr>
<td>6</td>
<td>$y_6$</td>
<td>$x_{\pi j</td>
<td>Z = b(6), j}$</td>
</tr>
<tr>
<td>14</td>
<td>$y_{14}$</td>
<td>$x_{\pi j</td>
<td>Z = b(14), j}$</td>
</tr>
<tr>
<td>33</td>
<td>$y_{33}$</td>
<td>$x_{\pi j</td>
<td>Z = b(33), j}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

$H_0 : X_j \perp Y | Z$

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

or $P(Y | X_j, Z) \overset{H_0}{=} P(Y | Z)$
Technically

▶ use any partition of the feature space for conditioning
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)
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
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)
Simulation study

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

\[ \Sigma = \begin{pmatrix}
1 & 0.9 & 0.9 & 0.9 & 0 & \ldots & 0 \\
0.9 & 1 & 0.9 & 0.9 & 0 & \ldots & 0 \\
0.9 & 0.9 & 1 & 0.9 & 0 & \ldots & 0 \\
0.9 & 0.9 & 0.9 & 1 & 0 & \ldots & 0 \\
0 & 0 & 0 & 0 & 1 & \ldots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix} \]

<table>
<thead>
<tr>
<th>( X_j )</th>
<th>( X_1 )</th>
<th>( X_2 )</th>
<th>( X_3 )</th>
<th>( X_4 )</th>
<th>( X_5 )</th>
<th>( X_6 )</th>
<th>( X_7 )</th>
<th>( X_8 )</th>
<th>\ldots</th>
<th>( X_{12} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta_j )</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>-5</td>
<td>-5</td>
<td>-2</td>
<td>0</td>
<td>\ldots</td>
<td>0</td>
</tr>
</tbody>
</table>
Results

- **mtry = 1**
  - Variable importance
  - Scores range from 0 to 25

- **mtry = 3**
  - Variable importance
  - Scores range from 0 to 30

- **mtry = 8**
  - Variable importance
  - Scores range from 0 to 80
Peptide-binding data
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)
Summary

if your predictor variables are of different types:
use `cforest (pkg: party)` with default option `controls = cforest_unbiased()`
with permutation importance `varimp(obj)`
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`)
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`)


