Stability Assessment of Tree Ensembles and Psychotrees

Using the `stablelearner`[1] package

Lennart Schneider[1,2] Achim Zeileis[3] Carolin Strobl[2]

Ludwig Maximilian University of Munich[1] University of Zurich[2] University of Innsbruck[3]

28.02.2020

[1]Philipp, Zeileis, and Strobl (2016) and Philipp et al. (2018)

Decision Trees

`stablelearner`

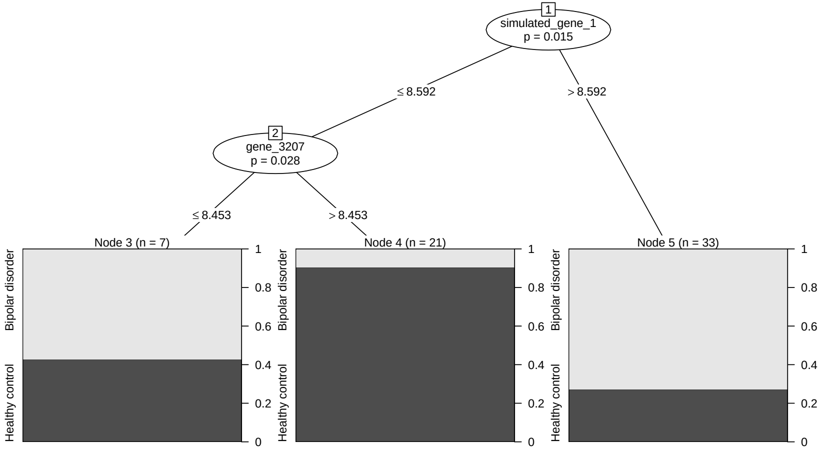
`stablelearner` and Tree Ensembles

`stablelearner` and `psychotrees`

Decision Trees

Classification, Regression and Model-Based Trees

Decision trees are supervised learners that predict the value of a target variable based on several input variables:



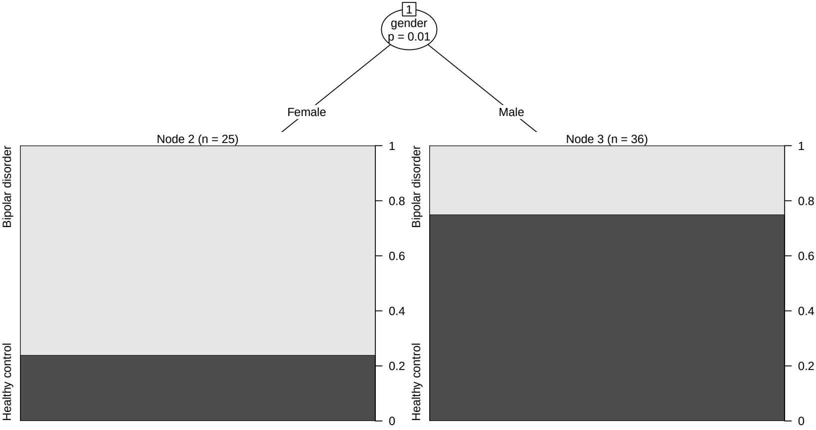
In R, e.g., `party` or `partykit` (Hothorn, Hornik, and Zeileis 2006; Zeileis, Hothorn, and Hornik 2008)

Classification, Regression and Model-Based Trees

- Easy to understand and interpret
- Handles both numerical and categorical data

- But: A single tree can be very non-robust

Classification, Regression and Model-Based Trees



stablelearner

stablelearner

`stablelearner` (Philipp, Zeileis, and Strobl 2016; Philipp et al. 2018):

- A toolkit of descriptive measures and graphical illustrations based on resampling and refitting
- Can be used to assess the stability of the variable and cutpoint selection in recursive partitioning

stablelearner - How does it work?

	Single Tree	Tree Ensemble
1.	Original Tree	
2.	Resampling & Refitting	
3.	Aggregating & Visualizing	

stablelearner

```r
library("partykit")
library("stablelearner")
```

```r
data("Bipolar2009", package = "stablelearner")
Bipolar2009$simulated_gene_2 <- cut(Bipolar2009$simulated_gene_2, breaks = 3,
                                    ordered_result = TRUE)
```

```r
str(Bipolar2009, list.len = 6)
```

```
## 'data.frame':    61 obs. of  106 variables:
##  $ age             : int  41 51 29 45 45 29 33 56 48 42 ...
##  $ brain_pH        : num  6.6 6.67 6.7 6.03 6.35 6.39 6.51 6.07 6.5 6.65 ...
##  $ status          : Factor w/ 2 levels "Bipolar disorder",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ gender          : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 1 2 1 2 ...
##  $ gene_921        : num  8.33 7.99 8.01 7.83 8.51 ...
##  $ gene_4211       : num  6.25 7.02 6.54 6.14 6.65 ...
##   [list output truncated]
```

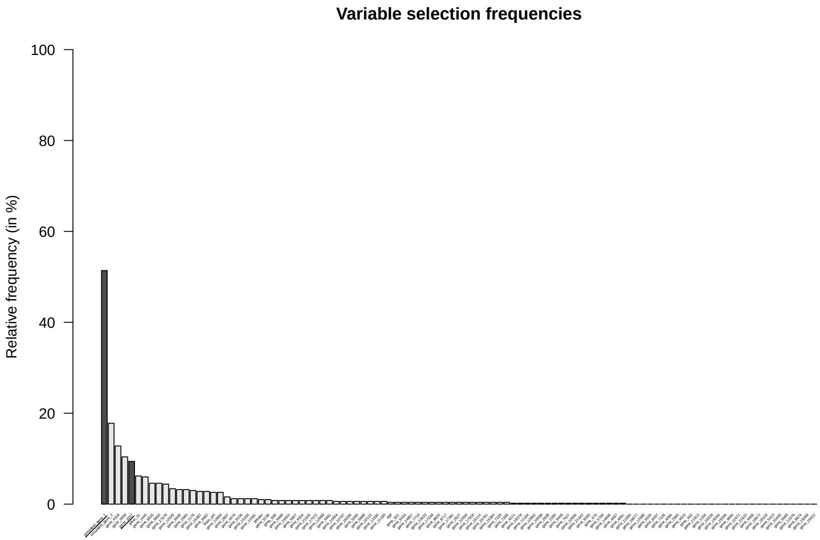
```r
ct <- ctree(status ~ ., data = Bipolar2009)
ct_stable <- stabletree(ct)
```

stablelearner - summary

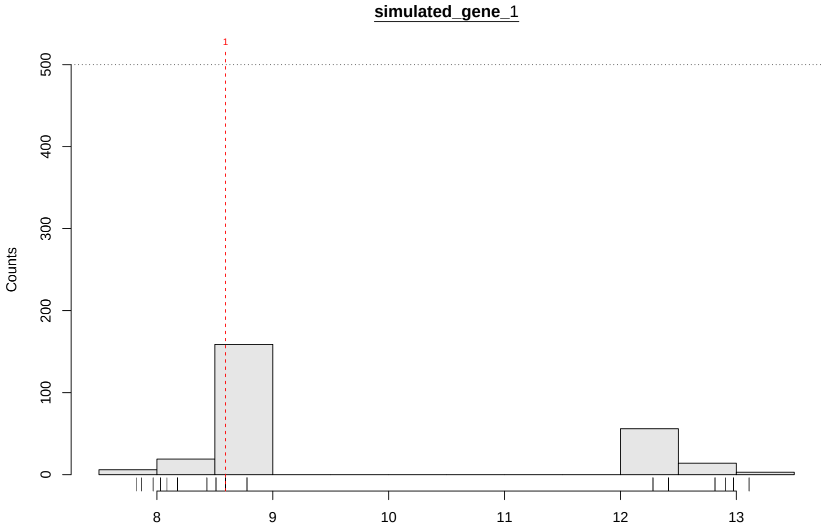
```r
summary(ct_stable)
```

```
##
## Call:
## partykit::ctree(formula = status ~ ., data = Bipolar2009)
##
## Sampler:
## B = 500
## Method = Bootstrap sampling with 100.0% data
##
## Variable selection overview:
##
##                    freq *  mean *
## simulated_gene_1 0.514 1 0.514 1
## simulated_gene_2 0.178 0 0.178 0
## gene_4318        0.128 0 0.128 0
## gene_3069        0.104 0 0.104 0
## gene_3207        0.094 1 0.094 1
## gene_31          0.062 0 0.062 0
## gene_1440        0.060 0 0.060 0
## gene_6935        0.046 0 0.048 0
## gene_9850        0.046 0 0.046 0
...
```

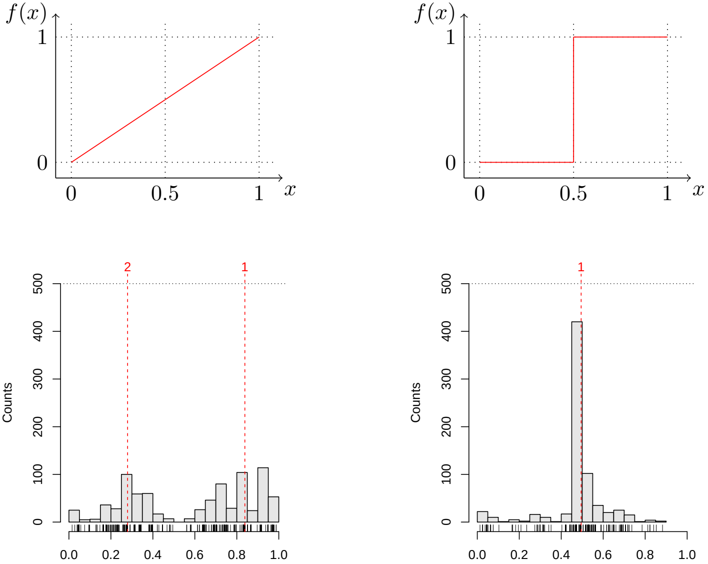
stablelearner - barplot



stablelearner - plot



stablelearner - plot



`stablelearner` and Tree Ensembles

What About Tree Ensembles

e.g., random forests?

	Single Tree	Tree Ensemble
1.	Original Tree	Base Learner
2.	Resampling & Refitting	Resampling & Refitting
3.	Aggregating & Visualizing	Aggregating & Visualizing

Two possibilities:

1. Fit a random forest in `stablelearner` using, e.g., `ctrees` as a base learner
2. Fit a random forest using the `randomForest` function of the `randomForest` package (Liaw and Wiener 2002), or the `cforest` function (of the `party` or `partykit` package) and coerce the forest to a `stabletree` object using the `as.stabletree` function

Random Forests in stablelearner

Possibility 1:

Use an appropriately specified `ctree` as a base learner and mimic a `cforest` of the `partykit` package:

```
ct_base <- ctree(status ~ ., data = Bipolar2009,
  control = ctree_control(mtry = 11, teststat = "quadratic",
                          testtype = "Univariate", mincriterion = 0,
                          saveinfo = FALSE))
```

```
cf_stable <- stabletree(ct_base, sampler = subsampling, savetrees = TRUE,
                        B = 500, v = 0.632)
```

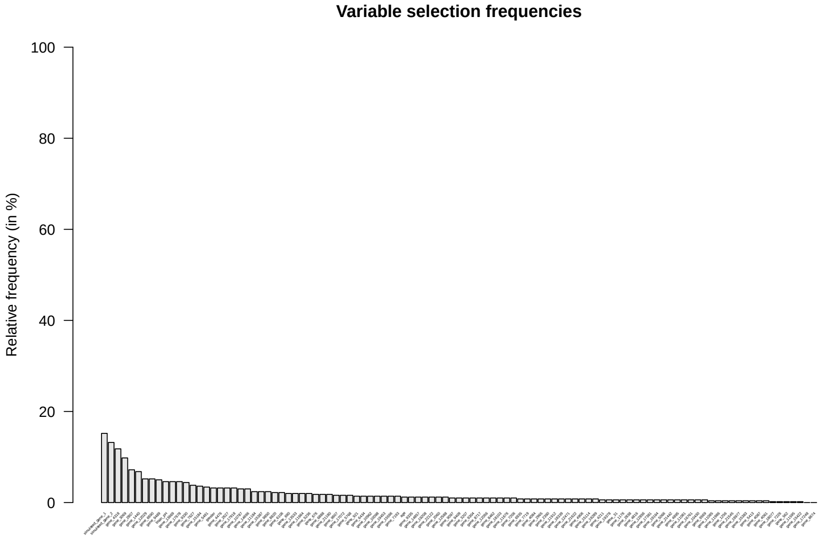
Note that this allows for **custom** builds, e.g., with respect to the resampling method (`bootstrap`, `subsampling`, `samplesplitting`, `jackknife`, `splithalf` or own sampling functions).

Random Forests in stablelearner

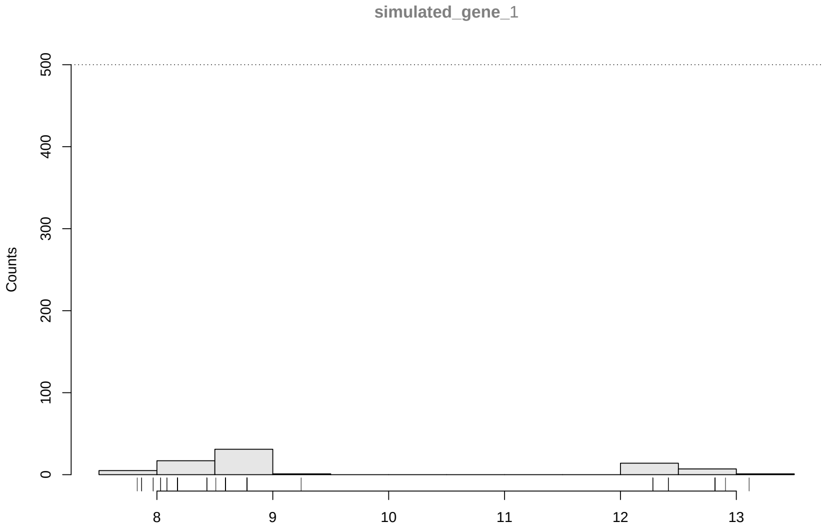
```r
summary(cf_stable, original = FALSE)
```

```
##
## Call:
## ctree(formula = status ~ ., data = Bipolar2009, control = ctree_control(mtry = 11,
##     teststat = "quadratic", testtype = "Univariate", mincriterion = 0,
##     saveinfo = FALSE))
##
## Sampler:
## B = 500
## Method = Subsampling with 63.2% data
##
## Variable selection overview:
##
##                   freq  mean
## simulated_gene_1 0.152 0.152
## simulated_gene_2 0.132 0.134
## gene_4318        0.118 0.118
## gene_3069        0.098 0.098
## gene_2807        0.072 0.072
## gene_1440        0.068 0.068
## gene_12029       0.052 0.052
...
```

stablelearner - barplot



stablelearner - plot



Random Forests in stablelearner

Possibility 2:

Fit a random forest externally, e.g., using the `cforest` function of the `partykit` package and coerce the forest.

```
cf_partykit <- cforest(status ~ ., data = Bipolar2009, mtry = 11)
cf_partykit_stable <- as.stabletree(cf_partykit)
```

```
summary(cf_partykit_stable)
#.
#.
#.
```

stablelearner and psychotrees

raschtrees

The `psychotree` package provides functionality for model based trees of, e.g., the Rasch model (Strobl, Kopf, and Zeileis 2015). This allows for a global test of Differential Item Functioning (DIF).

Stability of raschtrees

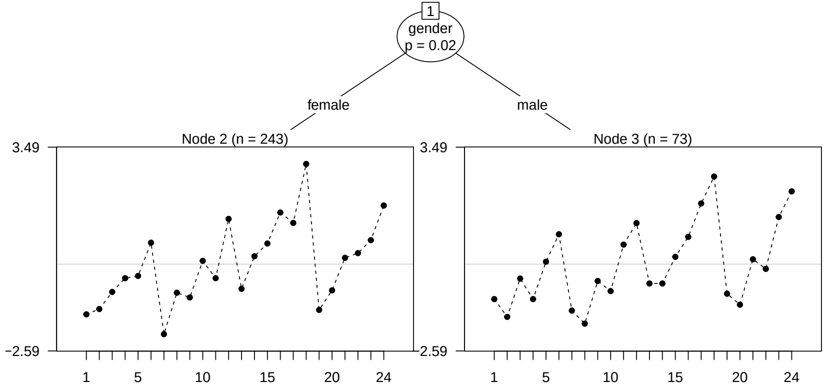
```r
library("psychotree")
```

```r
data("VerbalAggression", package = "psychotools")
str(VerbalAggression)
```

```
## 'data.frame':    316 obs. of  4 variables:
##  $ resp  : num [1:316, 1:24] 0 0 1 1 1 2 2 0 0 2 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr  "S1WantCurse" "S1DoCurse" "S1WantScold" "S1DoScold" ...
##  $ resp2 : num [1:316, 1:24] 0 0 1 1 1 1 1 0 0 1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr  "S1WantCurse" "S1DoCurse" "S1WantScold" "S1DoScold" ...
##  $ gender: Factor w/ 2 levels "female","male": 2 2 1 1 1 1 1 1 1 1 ...
##  $ anger : int  20 11 17 21 17 21 39 21 24 16 ...
```

```r
rt <- raschtree(resp2 ~ gender + anger, data = VerbalAggression, minsize = 30)
```

Stability of raschtrees



Stability of raschtrees

```r
rt_stable <- stabletree(rt, sampler = subsampling, v = 0.632)
```

```r
summary(rt_stable)
```

```
##
## Call:
## raschtree(formula = resp2 ~ gender + anger, data = VerbalAggression,
##     minsize = 30L)
##
## Sampler:
## B = 500
## Method = Subsampling with 63.2% data
##
## Variable selection overview:
##
##         freq *  mean *
## gender 0.270 1 0.270 1
## anger  0.004 0 0.004 0
## (* = original tree)
```

Some Observations

- Setting `minsize` too small results in very unstable item parameter estimates of the Rasch models that are fitted during fitting of the `raschtree`
- Stability results do vary strongly with respect to the sampler (`bootstrap` vs. `subsampling` vs. strata sampling), `subsampling` appears to perform well, see also Strobl et al. (2007)

Conclusion

- `stablelearner` can now be used to assess the stability of tree ensembles by either growing the ensemble using a base learner or by coercing an externally fitted ensemble
- stability assessment of psychotrees is technically straightforward

Thanks!

If anyone has experience with IRT resampling, please share your knowledge with me!

References I

Hothorn, T., K. Hornik, and A. Zeileis. 2006. "Unbiased Recursive Partitioning: A Conditional Inference Framework." *Journal of Computational and Graphical Statistics* 15 (3): 651–74. https://doi.org/10.1198/106186006x133933.

Liaw, A., and M. Wiener. 2002. "Classification and Regression by randomForest." *R News* 2 (3): 18–22.

Philipp, M., T. Rusch, K. Hornik, and C. Strobl. 2018. "Measuring the Stability of Results from Supervised Statistical Learning." *Journal of Computational and Graphical Statistics* 27 (4): 685–700. https://doi.org/10.1080/10618600.2018.1473779.

Philipp, M., A. Zeileis, and C. Strobl. 2016. "A Toolkit for Stability Assessment of Tree-Based Learners." In *Proceedings of COMPSTAT 2016 – 22nd International Conference on Computational Statistics*, edited by A. Colubi, A. Blanco, and C. Gatu, 315–25. The International Statistical Institute/International Association for Statistical Computing.

References II

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). https://doi.org/10.1186/1471-2105-8-25.

Strobl, C., J. Kopf, and A. Zeileis. 2015. "Rasch Trees: A New Method for Detecting Differential Item Functioning in the Rasch Model." *Psychometrika* 80 (2): 289–316. https://doi.org/10.1007/s11336-013-9388-3.

Zeileis, A., T. Hothorn, and K. Hornik. 2008. "Model-Based Recursive Partitioning." *Journal of Computational and Graphical Statistics* 17 (2): 492–514. https://doi.org/10.1198/106186008x319331.