Prediction rule ensembles

or a Japanese gardening approach to tree ensembles

Psychoco 2017

Marjolein Fokkema, Universiteit Leiden

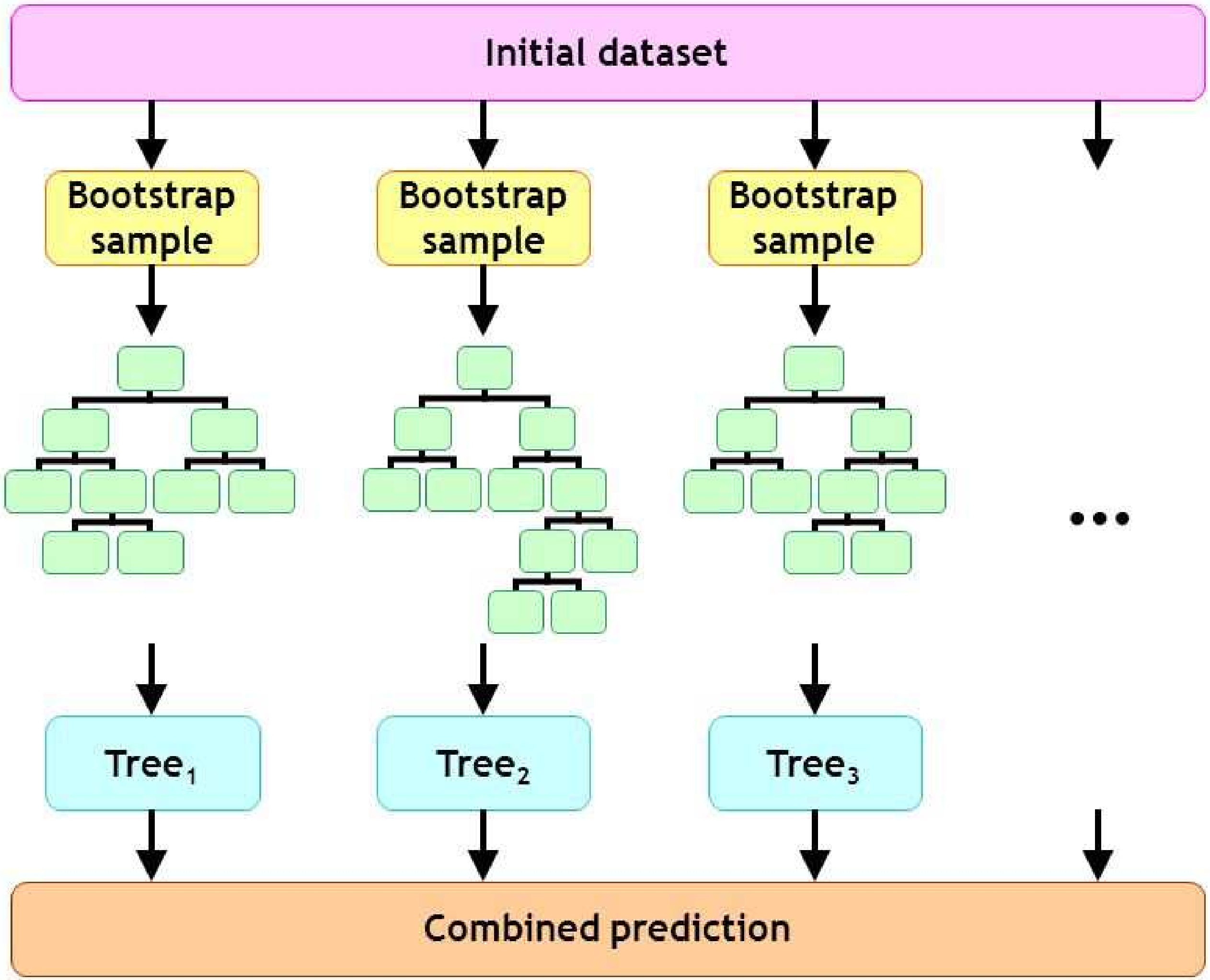
Trees

Good: Easily interpretable and applicable

Bad: Not most accurate method

Ugly: Unstable

Random forest



Random forests

\+ (One of the) most accurate methods for prediction

\- Difficult to interpret and apply:

> 100 trees in a single forest

Trees may be large/complex

Computer needs to combine predictions

Solution: Prediction rule ensembles

Cut the trees into small parts and use only most accurate parts

Rulefit (Friedman & Popescu, 2008)

Node Harvest (Meinshausen, 2010)

Example dataset

European Journal of Psychological Assessment, Vol. 17, Issue 2, pp. 130–136

Openness to Experience and Depression*

J.M. Carrillo, N. Rojo, M.L. Sánchez-Bernardos, and M.D. Avia

Departamento de Personalidad, Evaluación y Psicología Clínica, Facultad de Psicología, Universidad Complutense de Madrid, Spain

Keywords: Openness to experience, depression, openness to fantasy, openness to actions, PB theory of depression, gender and depression, five factor model

Summary: The present study examines, in the context of the Five Factor Model, the contradictory role played by the *Openness to Fantasy* and *Openness to Actions* facets (of the *Openness to Experience* factor) in the prediction of depression. The fact that our data are taken from a sample of the Spanish general population is also a cross-cultural contribution that must be emphasized. 112 participants – 50% females and 50% males – filled out the NEO-PI and the BDI depression questionnaires. A stepwise regression shows that the *Fantasy* facet

Example dataset

Aim: Predict depression based on personality scales

Predictor variables:

- Neuroticism:	n1-n6, ntot
- Extraversion:	e1-e6, etot
- Openness to exp.:	open1-open5, opentot
- Altruism:	altot
- Conscientiousness:	contot
- Covariates:	sex (sexo), age (edad)

Response variable:

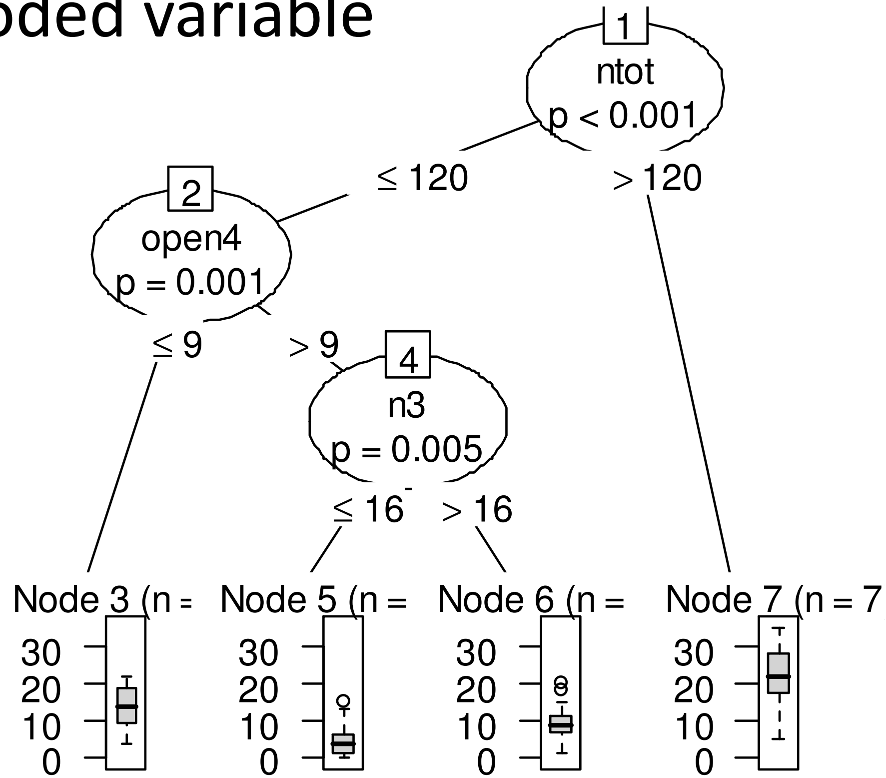
- Beck Depression Inventory total score (bdi)

Sample size: N = 112

From trees to rules

Every node from every tree is transformed into a rule

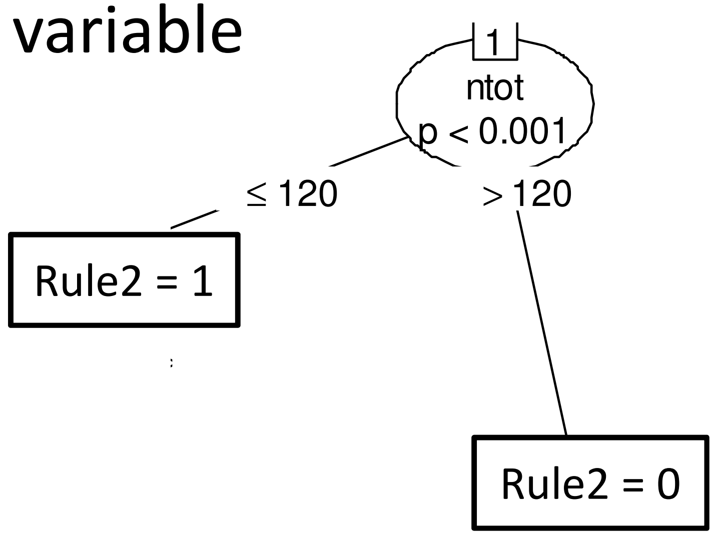
Each rule becomes a 0-1 coded variable



From trees to rules

Every node from every tree is transformed into a rule

Each rule becomes a 0-1 coded variable

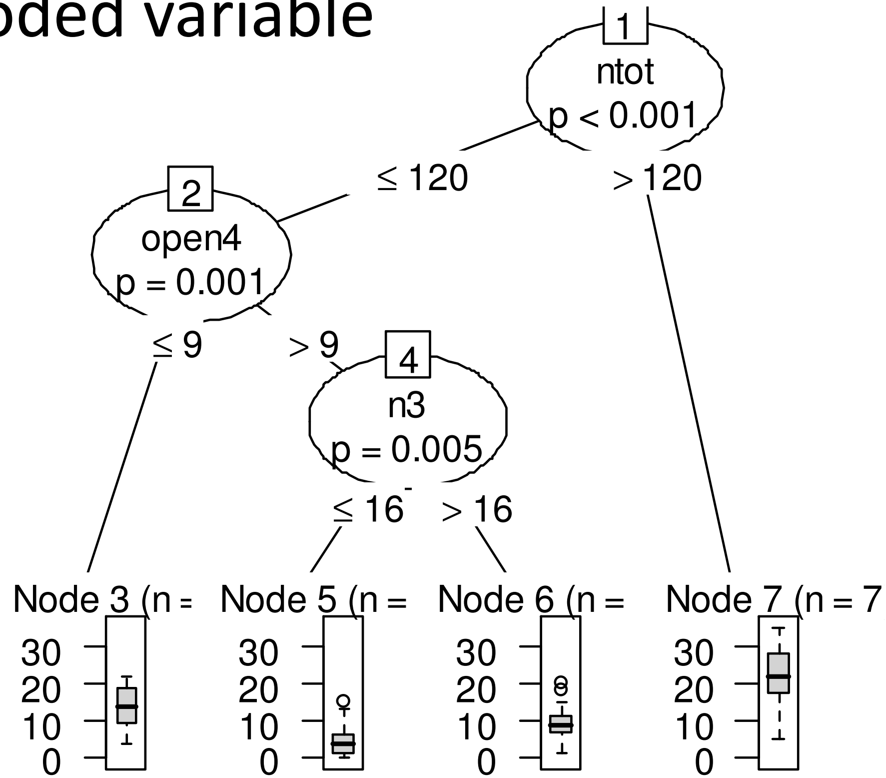


Rule2: ntot ≤ 120

From trees to rules

Every node from every tree is transformed into a rule

Each rule becomes a 0-1 coded variable

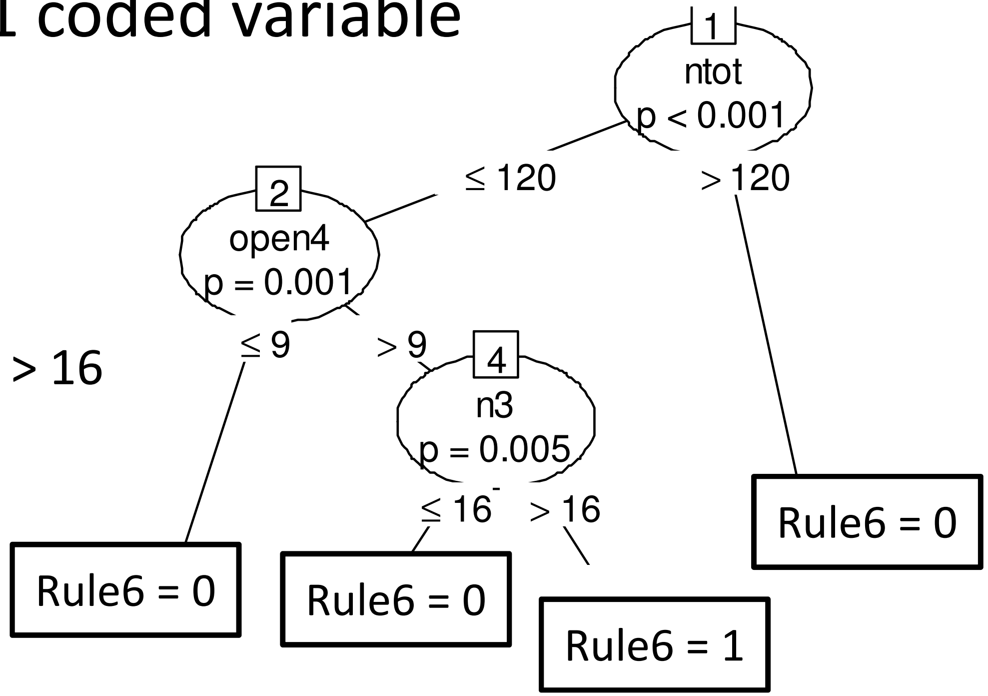


From trees to rules

Every node from every tree is transformed into a rule

Each rule becomes a 0-1 coded variable

Rule6: ntot ≤ 120 & open4 > 9 & n3 > 16



Rule ensemble algorithm

1) Take sub- or bootstrap samples from training data
2) Grow tree on each sample
   - Boosting: Set learning rate > 0 when growing trees, and/or
   - Random forest: Select random subset of predictor variables for selection of each split
3) Create initial ensemble
   - Include each node from each tree as a rule, and/or
   - Include predictor variables as linear functions
4) Select final ensemble using penalized regression

Code example

Psychoco2017.Rpres file on

https://github.com/marjoleinF/misc/

pre

Prediction Rule Ensembles

Psychoco 2017

Marjolein Fokkema, Leiden University

Load library and data

```
library(devtools)
install_github("marjoleinF/pre")
library(pre)
library(foreign)
cardata <- read.spss(
  "https://github.com/marjoleinF/misc/raw/master/data Carillo et al.sav",
  to.data.frame = TRUE)
set.seed(42)
train <- sample(1:112, 80)
```

Fitting a PRE

```
carpre <- pre(formula = bdi ~ ., data = cardata[train,], type = "both",
maxdepth = 3, learnrate = .01, mtry = Inf, sampfrac = .5)
```

- Above, default settings are specified
- To generate trees as a random forest:
  - maxdepth = Inf
  - learnrate = 0
  - mtry = srtq(p)
  - sampfrac = 1

Some methods

```
print(carpre)
```

```
Final ensemble with cv error within 1se of minimum:
  lambda = 0.9873066
  number of terms = 6
  mean cv error (se) = 32.80081 (6.512305)

        rule coefficient           description
 (Intercept)     10.4221                  <NA>
     rule104     -1.6236 n3 <= 18 & open4 > 12
      rule69     -1.5237    n6 <= 14 & e6 > 16
      rule31     -1.4096              n4 <= 15
       rule3     -0.5847              n3 <= 15
      rule66     -0.0296    n2 <= 16 & e6 > 18
       rule4      0.0000               n3 > 15
```

```
coef(carpre)
predict(carpre, newdata = cardata[-train,])
cvpre(carpre)
```

Importances

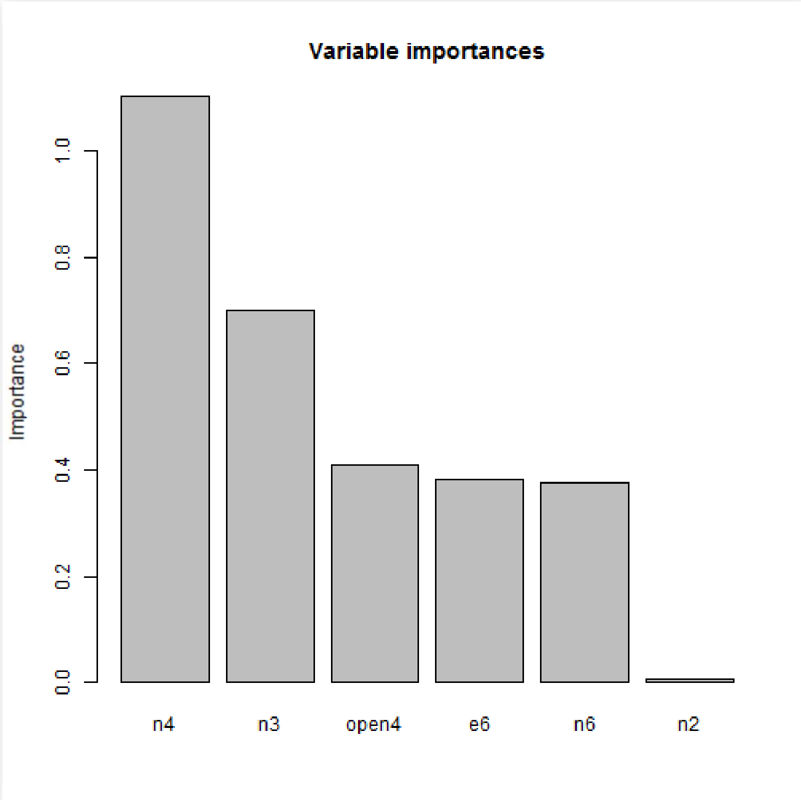
```
importance(carpre, round = 4, plot = FALSE)
```

```
$varimps
   varname    imp
4       n4 1.1034
3       n3 0.7011
18   open4 0.4084
13      e6 0.3830
6       n6 0.3756
2       n2 0.0074

$baseimps
               description    imp coefficient      sd
33  n3 <= 18 & open4 > 12 0.8169     -1.6236 0.5032
115    n6 <= 14 & e6 > 16 0.7512     -1.5237 0.4930
74                n4 <= 15 0.6949     -1.4096 0.4930
72                n3 <= 15 0.2927     -0.5847 0.5006
112    n2 <= 16 & e6 > 18 0.0148     -0.0296 0.5006
83                 n3 > 15 0.0000      0.0000 0.5006
```

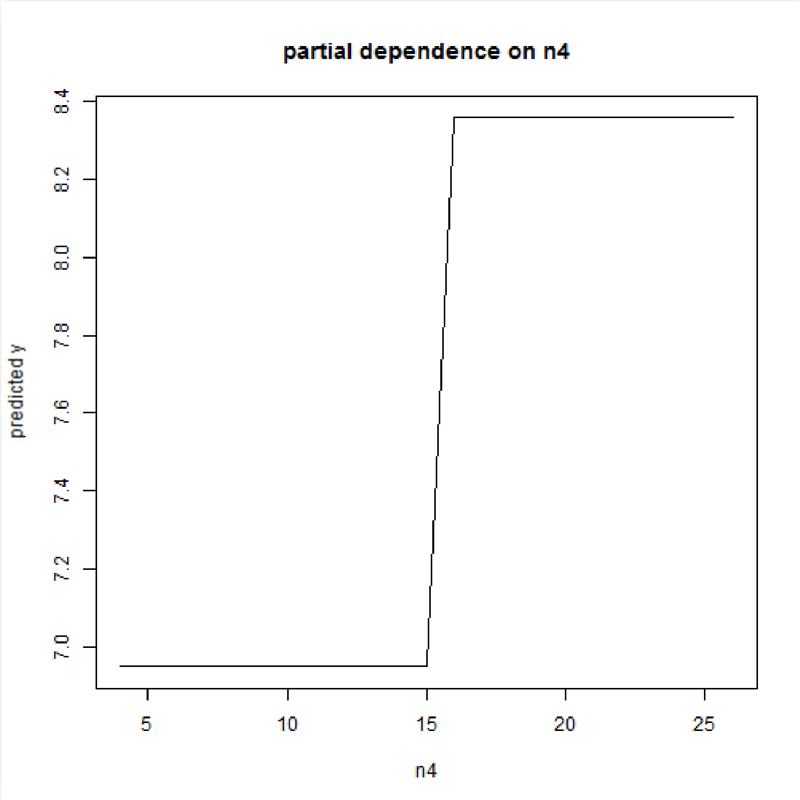
Importances

```
imps <- importance(carpre)
```



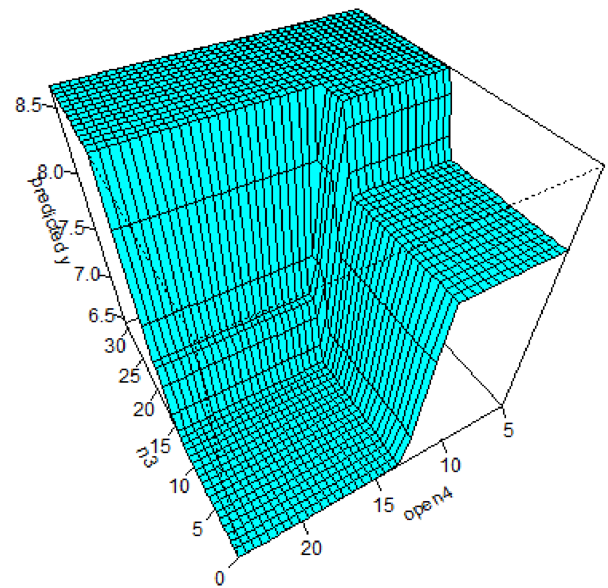
2D partial dependence plot

```
singleplot(carpre, "n4")
```



3D partial dependence plot

```
pairplot(carpre, c("n3", "open4"), nticks = 6, theta = 240)
```

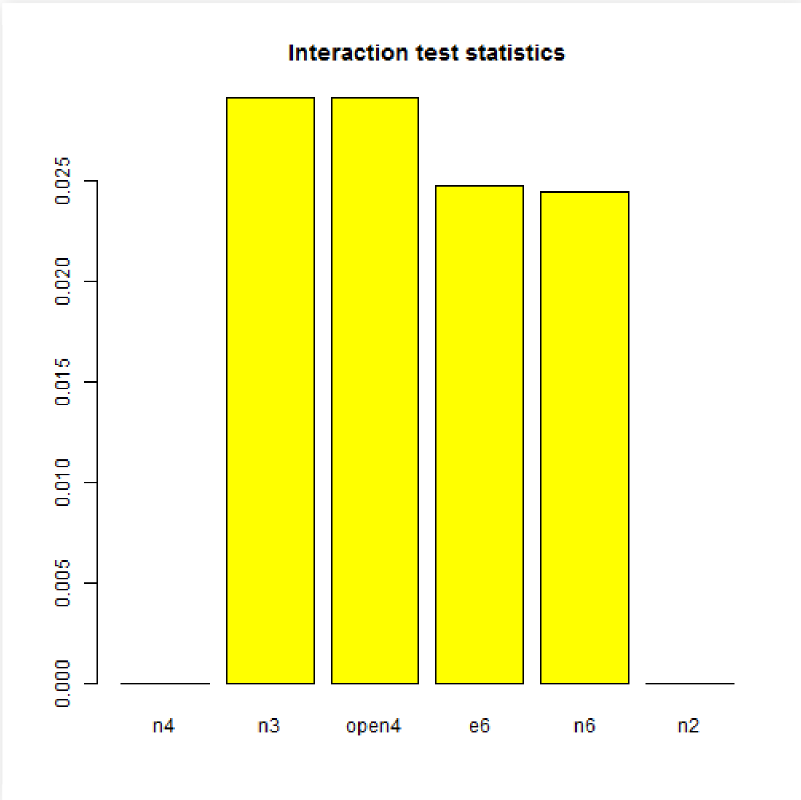



```
NOTE: function pairplot uses package 'akima', which has an ACM license.
```

Assessing interactions

```
int1 <- interact(carpre)
```

```
This will take a while ( 60 dots ).
............................................................
```



Assessing interactions

To assess significance of interactions, we can compute null interaction models from bootstrapped datasets, to derive a reference distribution of the interaction test statistics.

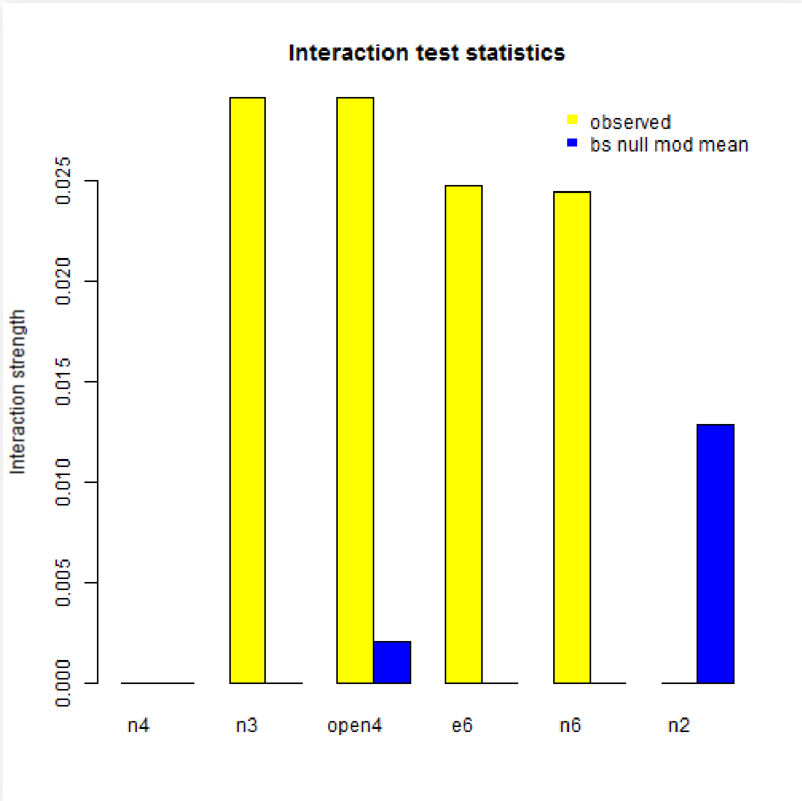
```
bsnullmods <- bsnullinteract(carpre, nsamp = 2)
```

```
This may take a while. Computing null model 1 of 2 ... 2 of 2 ... Done!
```

Assessing interactions

```
int2 <- interact(carpre, nullmods = bsnullmods)
```

```
This will take a while ( 180 dots ).
..........................................................................................
```



What’s next?

- Reducing computation time
- Plotting functions
- Support for other outcomes than continuous and binary ones

Haiku

Forest to branches

prediction rule ensembles

get it from GitHub

https://github.com/marjoleinF/pre

https://cran.r-project.org/web/packages/pre/index.html

Thank you for your attention!

References

Fokkema, M., Smits, N., Kelderman, & Penninx, B.W.J.H. (2015). Connecting clinical and actuarial prediction with rule-based methods. Psychological Assessment, 27(2), 636-644.

Friedman, J. H., & Popescu, B. E. (2008). Predictive learning via rule ensembles. The Annals of Applied Statistics, 2(3), 916-954.

Meinshausen, N. (2010). Node harvest. The Annals of Applied Statistics, 4(4), 2049-2072.