Prediction rule ensembles

or: a Japanese gardening approach to tree ensembles

Marjolein Fokkema
Institute of Psychology
Leiden University

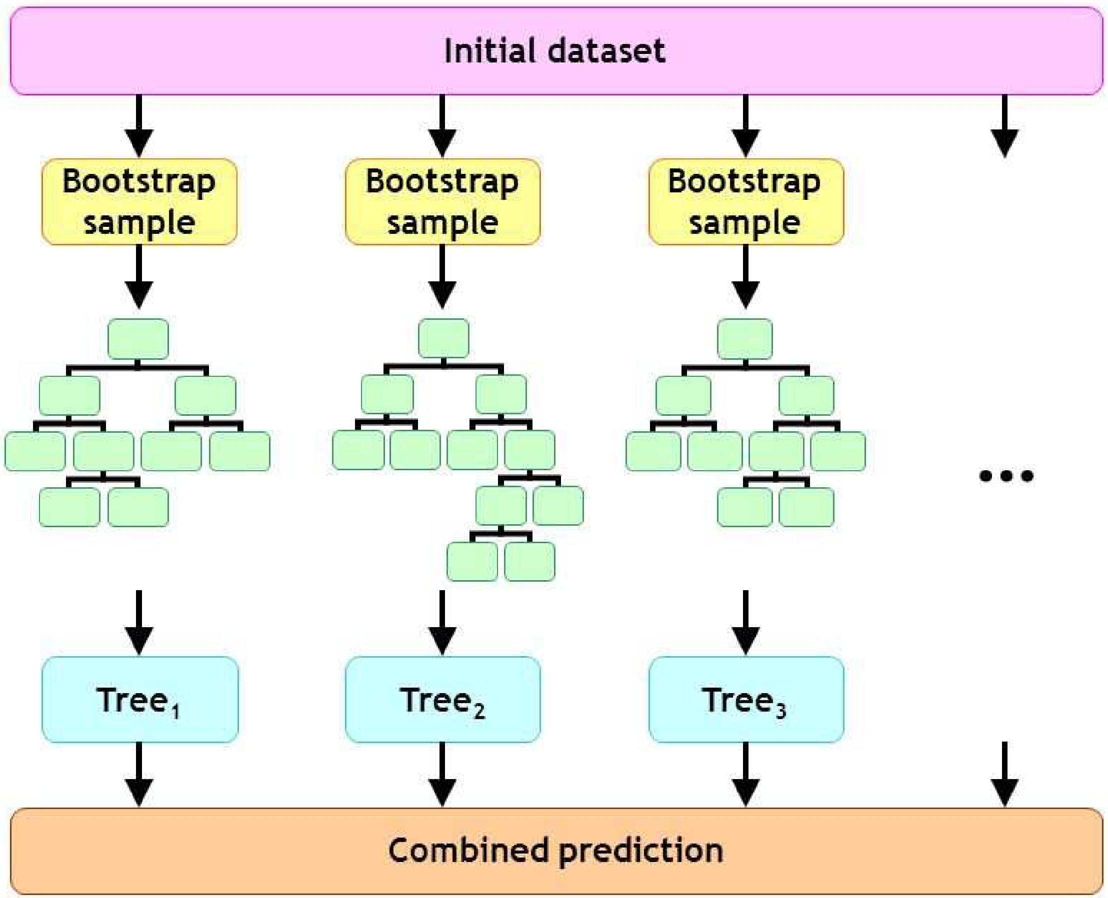
Trees

Good: Easily interpretable and applicable

Bad: Not most accurate method

Ugly: Unstable

Tree ensembles



Tree ensembles

+ High predictive accuracy

- Difficult to interpret

  many (complex) trees

- Difficult to apply

  prediction for new observations requires a lot of computation and information

Solution: Prediction rule ensembles

Take every node from every tree, select only most accurate ones

E.g., 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

24 predictor variables:
- 7 Neuroticism scales
- 7 Extraversion scales
- 6 Openness scales
- Altruism scale, Conscientiousness scale, Sex & Age

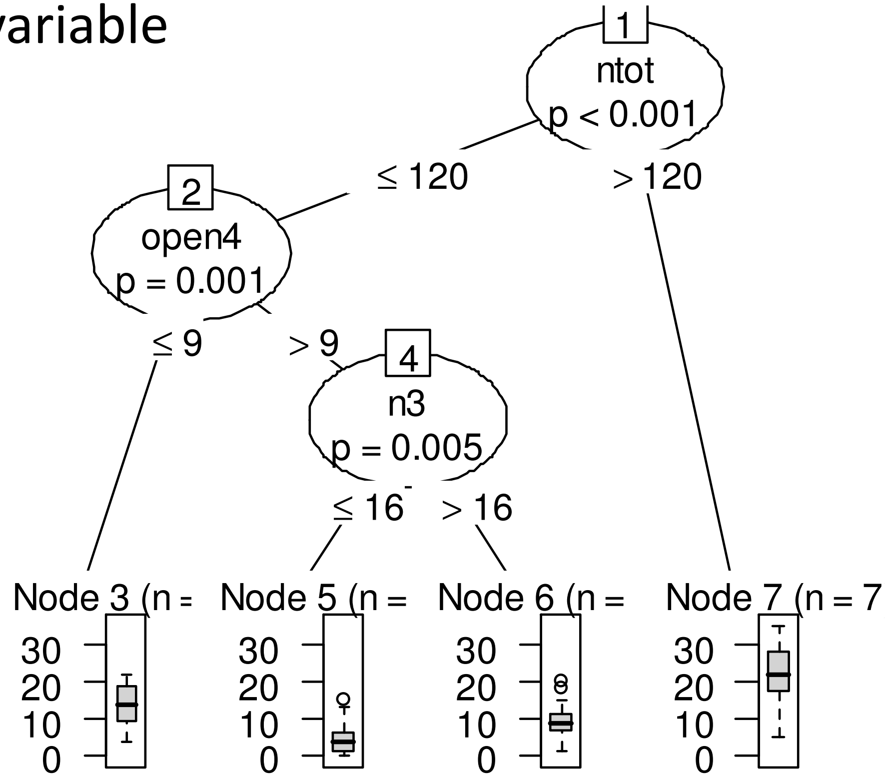
Response variable:
- Beck Depression Inventory total score

Sample size: N = 112

From trees to rules

Path from the root to another node in thee tree is a rule

Each rule can be coded as a 0-1 variable

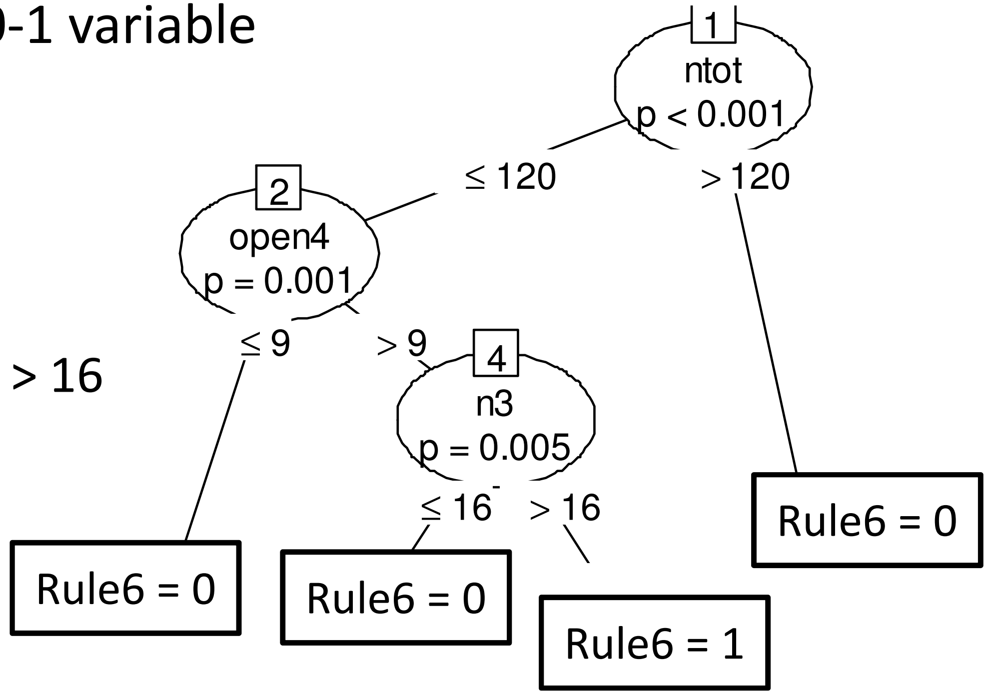


From trees to rules

Path from the root to another node in de tree is a rule

Each rule can be coded as a 0-1 variable

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



Rule ensemble algorithm

1) Take sub- or bootstrap samples from training data
2) Grow tree on every sample
    - Unbiased recursive partitioning
    - Boosting: learning rate > 0
    - Random forest: Use random subset of mtry variables for split selection
3) Create initial ensemble
    - Include every node from every tree as a rule, and/or
    - Include predictor variables as linear functions
4) Select final ensemble by sparse regression on training data
    - Lasso, ridge or elastic net
    - Optimal value of penalty term determined by k-fold cv

Code example

zukost.Rpres file on https://github.com/marjoleinF/misc/

Prediction rule ensembles

Marjolein Fokkema, Leiden University

Fitting a PRE

```
library(pre)
library(foreign)
car_data <- read.spss("https://github.com/marjoleinF/misc/raw
/master/data Carillo et al.sav", to.data.frame = TRUE)

## Fit ensemble:
set.seed(42)
car_pre <- pre(formula = bdi ~ ., data = car_data, type = "both",
sampfrac = 0.5, maxdepth = 3, mtry = Inf, learnrate = .01, ntrees =
500)
```

Above, defaut settings are specified.

Alternatively, we can generate the initial ensemble like a bagged ensemble or random forest:

```
pre_bag <- pre(formula = bdi ~ ., data = car_data, maxdepth = Inf,
learnrate = 0, mtry = Inf, sampfrac = 1)

pre_rf <- pre(formula = bdi ~ ., data = car_data, maxdepth = Inf,
learnrate = 0, mtry = ncol(car_data)/3, sampfrac = 1)
```

Print method

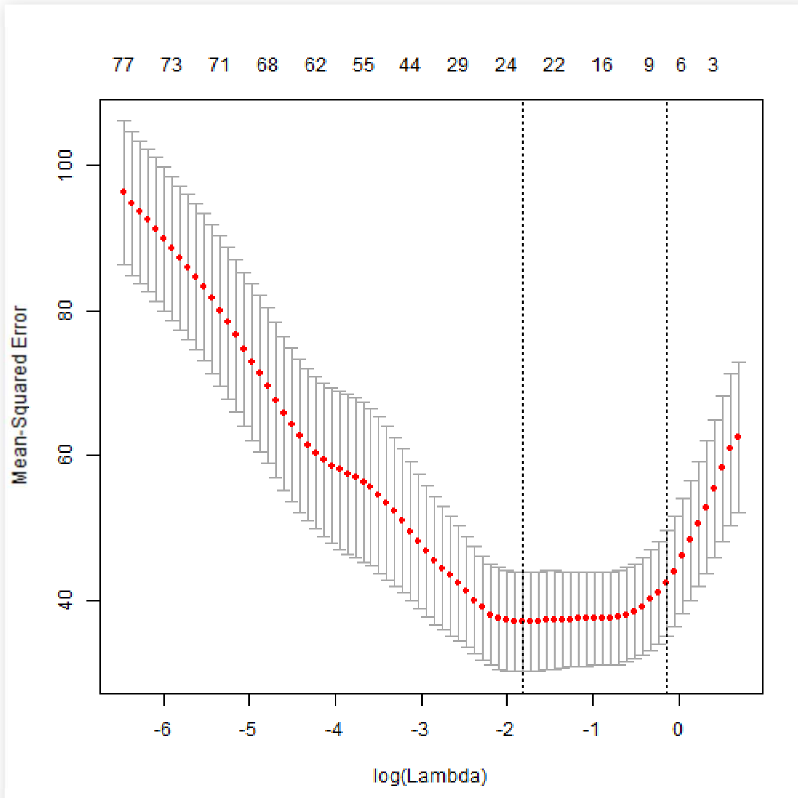
```
print(car_pre)
```

```
Final ensemble with cv error within 1se of minimum:
  lambda =  0.8696862
  number of terms = 8
  mean cv error (se) = 42.38802 (7.282903)
```

```
          rule coefficient       description
1  (Intercept)   7.5888321              <NA>
57      rule27  -1.6628834          n4 <= 15
62      rule48   1.0859951       open4 <= 13
60      rule37  -1.0741043       ntot <= 110
26       rule1  -0.9899318          n3 <= 17
64      rule58  -0.7930130          n1 <= 20
52      rule17   0.7317999           n3 > 22
40     rule139   0.2445816          e2 <= 16
14          n3   0.1793964 2 <= n3 <= 30.225
```

Select smaller ensemble (1)

```
plot(car_pre$glmnet.fit)
```



Select smaller ensemble (2)

```
head(cbind(car_pre$glmnet.fit$nzero, car_pre$glmnet.fit$lambda))
```

```
   [,1]     [,2]
s0    0 2.009088
s1    1 1.830606
s2    3 1.667980
s3    3 1.519801
s4    4 1.384786
s5    4 1.261766
```

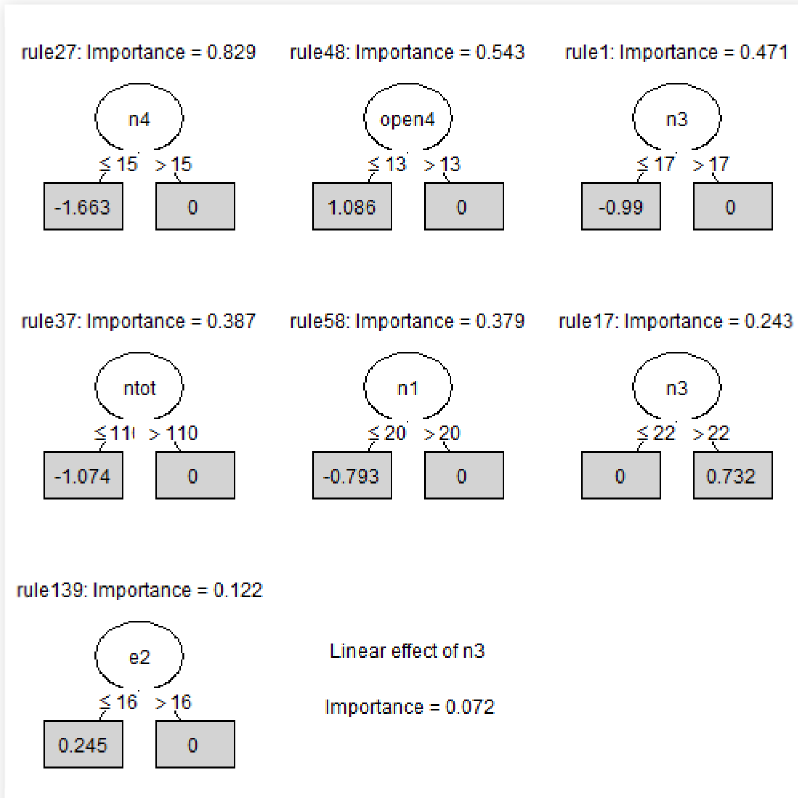
```
print(car_pre, penalty.par.val = 1.38)
```

```
Final ensemble with lambda =  1.384786
  number of terms = 4
  mean cv error (se) = 52.88966 (9.088698)
```

```
          rule coefficient       description
1  (Intercept)   7.8015710              <NA>
26       rule1  -1.1767087         n3 <= 17
57      rule27  -0.8595558         n4 <= 15
64      rule58  -0.1436922         n1 <= 20
14          n3   0.1023191 2 <= n3 <= 30.225
```

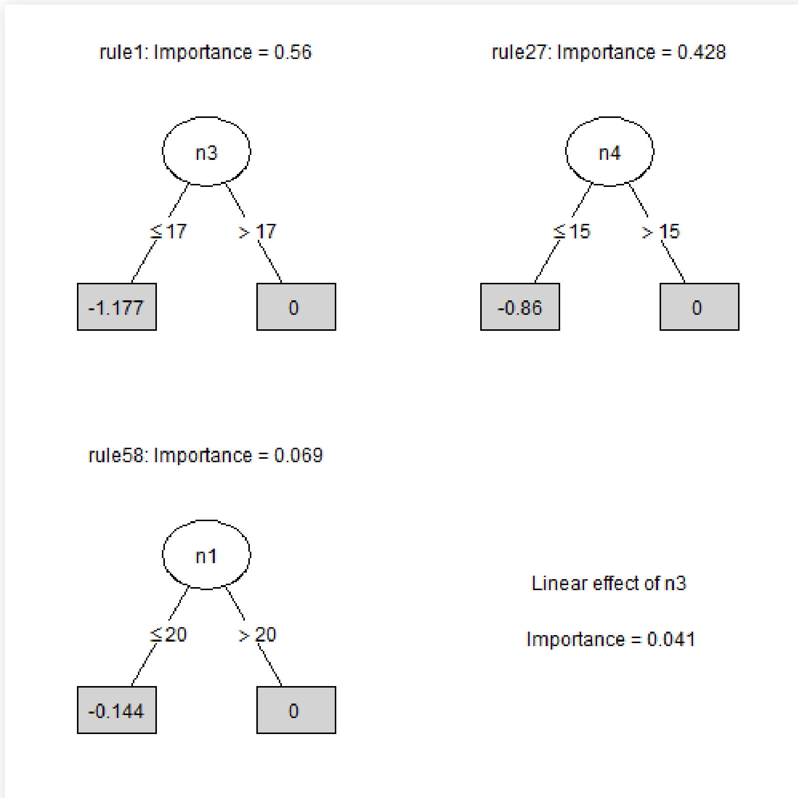
Plot method (1)

```
plot(car_pre)
```



Plot method (2)

```
plot(car_pre, penalty.par.val = 1.38)
```



Other methods

```
head(coef(car_pre))
```

```
          rule coefficient description
1  (Intercept)   7.5888321        <NA>
57      rule27  -1.6628834    n4 <= 15
62      rule48   1.0859951 open4 <= 13
60      rule37  -1.0741043 ntot <= 110
26       rule1  -0.9899318    n3 <= 17
64      rule58  -0.7930130    n1 <= 20
```

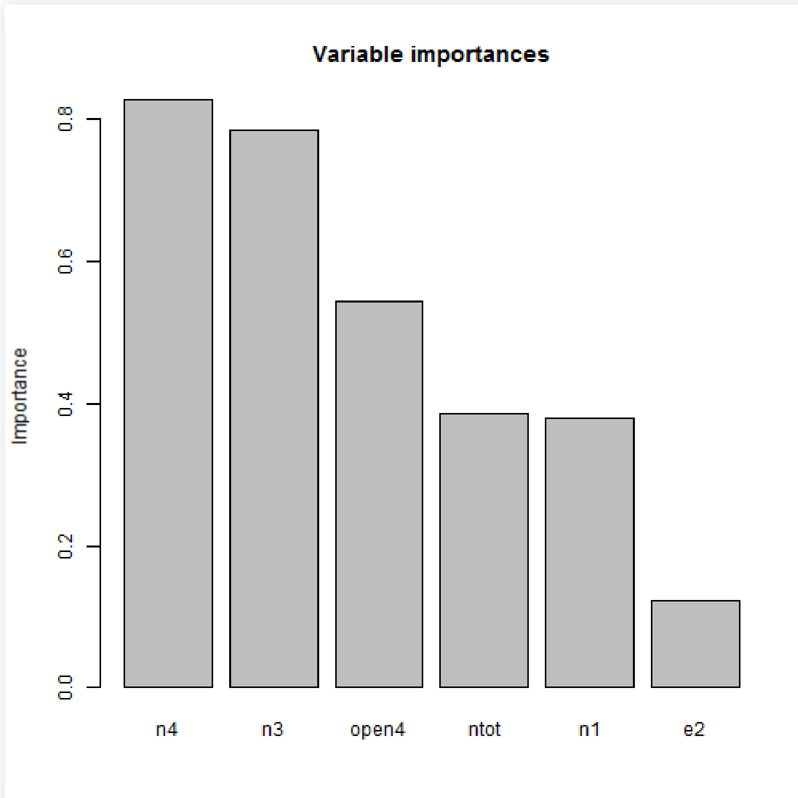
```
predict(car_pre, newdata = car_data[1:10,])
```

```
        1         2         3         4         5         6         7 
 6.183824 12.870727  6.487048  6.525747  6.845841  8.439773 15.212497 
        8         9        10 
 7.532548  3.068900  7.563342 
```

```
set.seed(42)
cvpre(car_pre)
```

Importances (1)

```
imps <- importance(car_pre, round = 4)
```



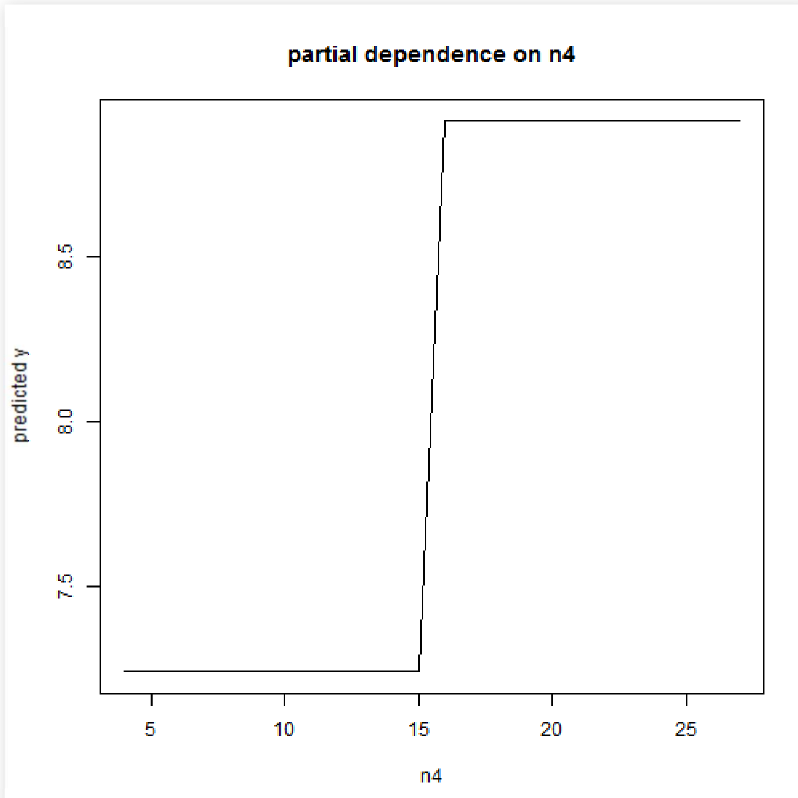
Importances (2)

```
imps$baseimps
```

```
      rule        description    imp coefficient      sd
58  rule27          n4 <= 15 0.8286     -1.6629 0.4983
63  rule48       open4 <= 13 0.5433      1.0860 0.5002
27   rule1          n3 <= 17 0.4708     -0.9899 0.4756
61  rule37       ntot <= 110 0.3871     -1.0741 0.3604
65  rule58          n1 <= 20 0.3795     -0.7930 0.4785
53  rule17           n3 > 22 0.2431      0.7318 0.3322
41 rule139          e2 <= 16 0.1216      0.2446 0.4971
13      n3 2 <= n3 <= 30.225 0.0718      0.1794 0.4000
```

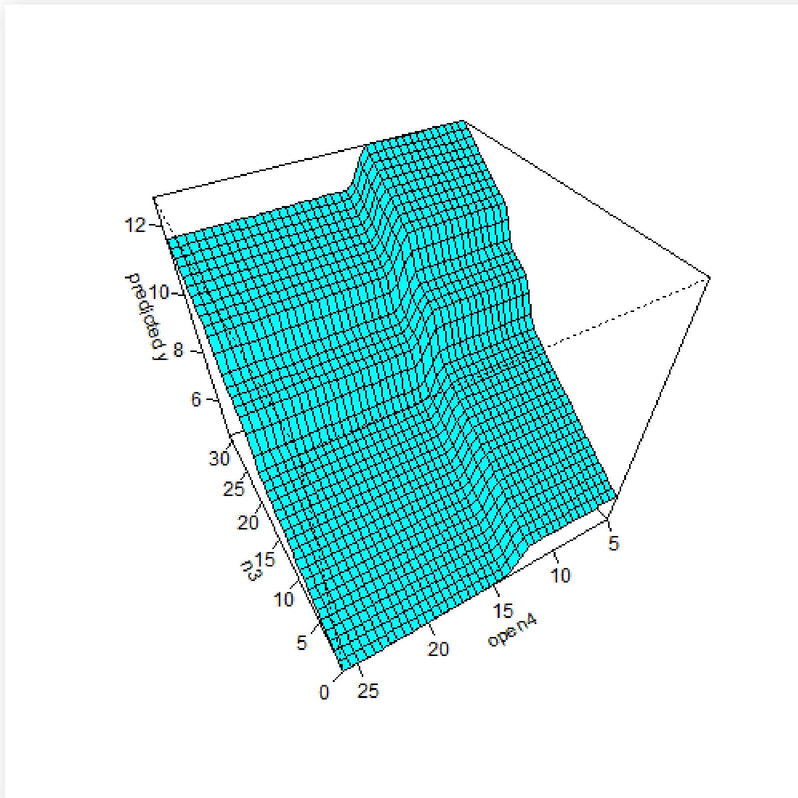
Partial dependence plots (1)

```
singleplot(car_pre, "n4")
```



Partial dependence plots (2)

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




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

Interactions

```
nullmods = bsnullinteract(car_pre)
interact(car_pre, nullmods = nullmods)
```

Resolution

Did we kill the bad?

Did the good survive?

Preliminary simulation results

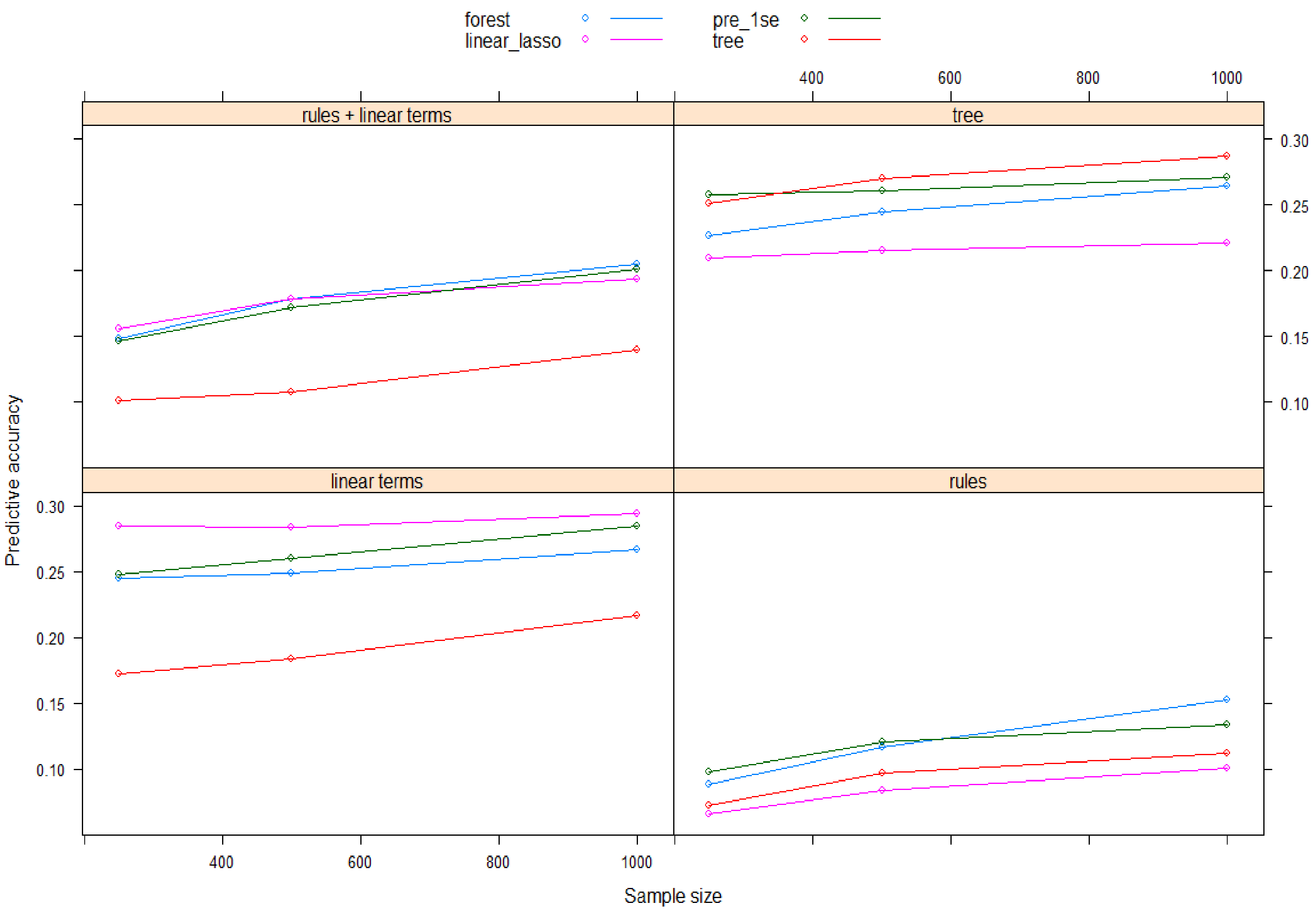
PRE vs. linear lasso, random forest and single tree

Data-generating parameters:

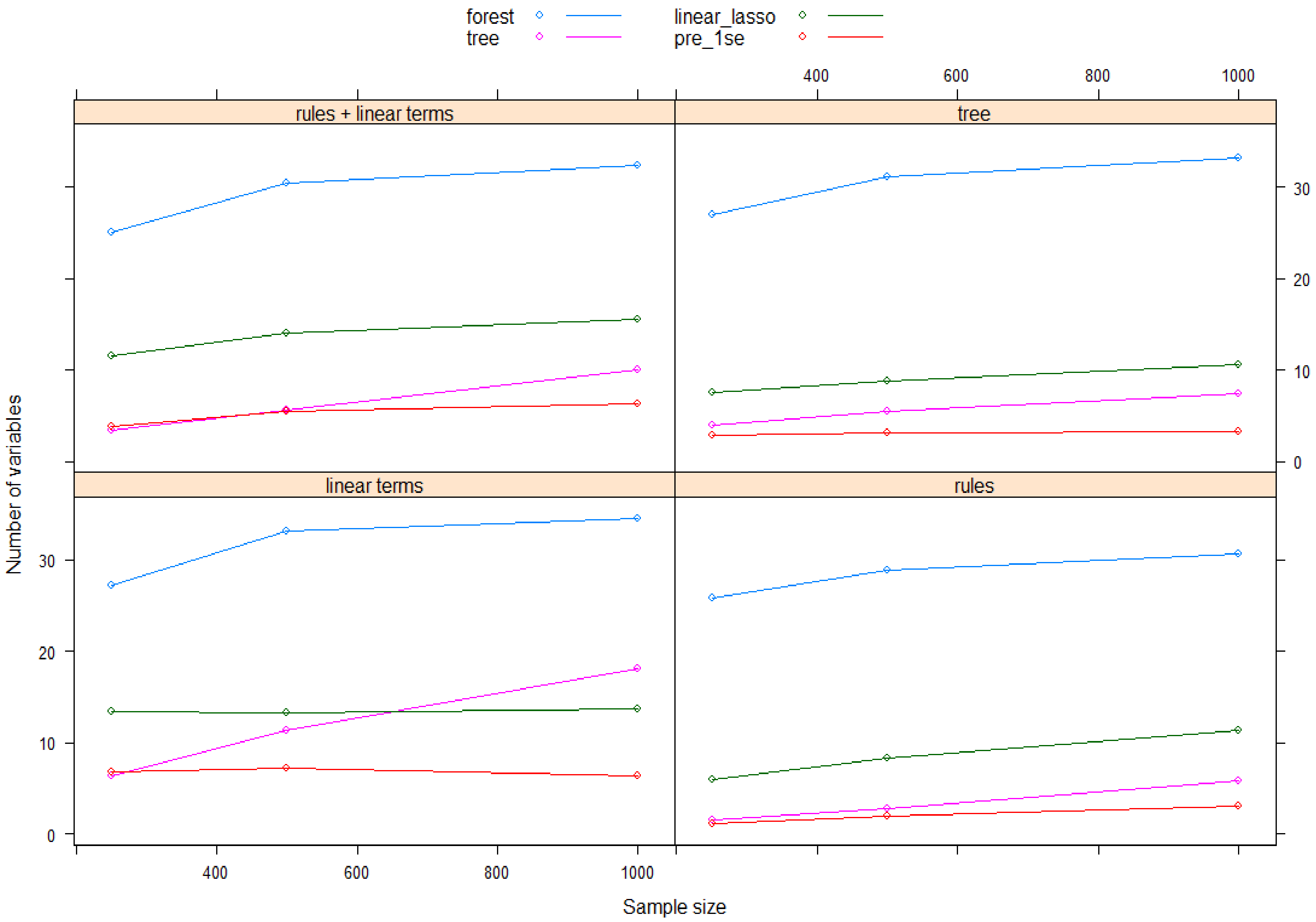
- Sample size
  250 – 1,000
- Signal / noise ratio
  .25 - .50
- True model
  linear, rules, rules + linear, tree
- Number of noise variables
  15 – 85

≈ 1,500 datasets

Predictive accuracy



Interpretability



Discussion

PREs seem to deal well with instability, retaining interpretability as well as accuracy

Future work:

- Multivariate outcomes
- Dependent observations (longitudinal or multilevel data)
- Computational speed
- . . .

Thank you for your attention

https://github.com/marjoleinF/pre

https://cran.r-project.org/package=pre

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.