Prediction rule ensembles

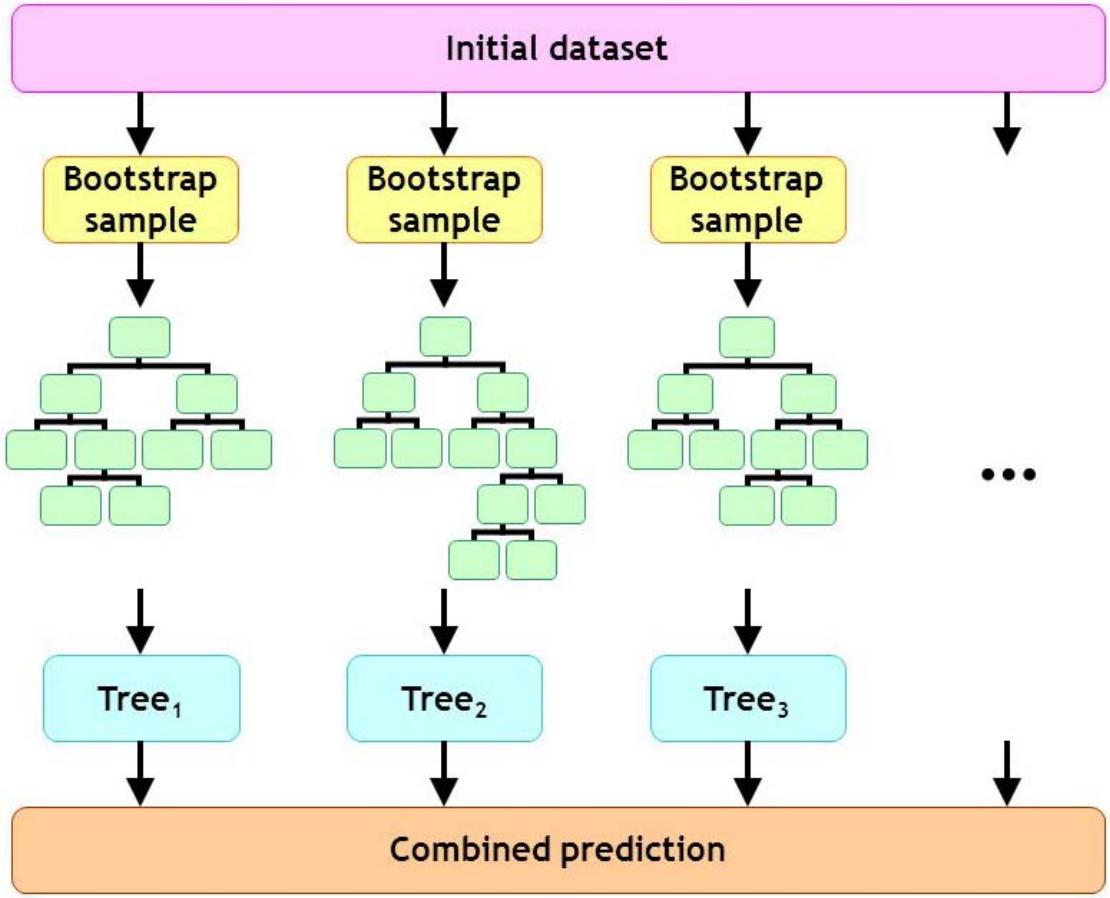
Trees

Good: Easily interpretable and applicable

Bad: Not most accurate method

Ugly: Unstable

Tree ensembles



Tree ensembles

Advantage: High predictive accuracy

Disadvantage: Difficult to interpret and apply

- Many (possibly complex) trees
- Prediction for new observations requires a lot of computation and all predictor variables need to be assessed

Solution: Prediction rule ensembles

- Start with tree ensemble
- Every node in every tree can be written as a rule (i.e., 0-1 coded variable reflecting node membership)
- Select rules that contribute most to predictive accuracy
- Rulefit (Friedman & Popescu, 2008), Node Harvest (Meinshausen, 2010)

Rulefit algorithm for deriving prediction rule ensembles

1) Take subsamples from training data
2) Grow tree on every sample
   - Boosting with CART trees
3) Create initial ensemble
   - Includes every node from every tree as a rule, and/or
   - Predictor variables as linear functions
4) Select final ensemble by sparse regression on training data
   - Lasso, ridge, elastic net, forward stepwise
   - Optimal value of penalty parameter determined by k-fold cv

Original implementation in Fortran: fast, not open source, not flexible, not well-documented.

R package **pre**
(Fokkema & Christoffersen, 2017)

1) Take sub- or bootstrap samples from training data
2) Grow tree on every sample

   Tuning parameters: learning rate, partitioning algorithm (ctree, glmtree or rpart), ntrees, sampling fraction, max. tree depth, learning rate, …
3) Create initial ensemble
   - Every node from every tree as a rule, and/or
   - Predictor variables as linear functions
   - Experimental: hinge functions and user-defined baselearners
4) Select final ensemble by sparse regression on training data

   Currently only lasso, ridge or elastic net

Example: Depression data

Cross-sectional dataset, N = 112 Spanish respondents

Response:

- Depressive symptomatology, as measured by the Beck Depression Inventory (BDI)

Predictors:

- Neuroticism subscale and total scores: n1, n2, n3, n4, n5, n6, ntot
- Extraversion subscale and total scores: e1, e2, e3, e4, e5, e6, etot
- Openness to experience subscale and total scores: open1, open2, open3, open4, open5, open6, opentot
- Altruism total score: altot
- Conscientiousness total score: contot
- Sex
- Age in years

Rule generation

$r_2(\mathrm{x}) = I(n3 \leq 21)$

$r_3(\mathrm{x}) = I(n3 \leq 21) \cdot I(ntot \leq 110)$

$r_4(\mathrm{x}) = I(n3 \leq 21) \cdot I(ntot \leq 110) \cdot I(etot \leq 101)$

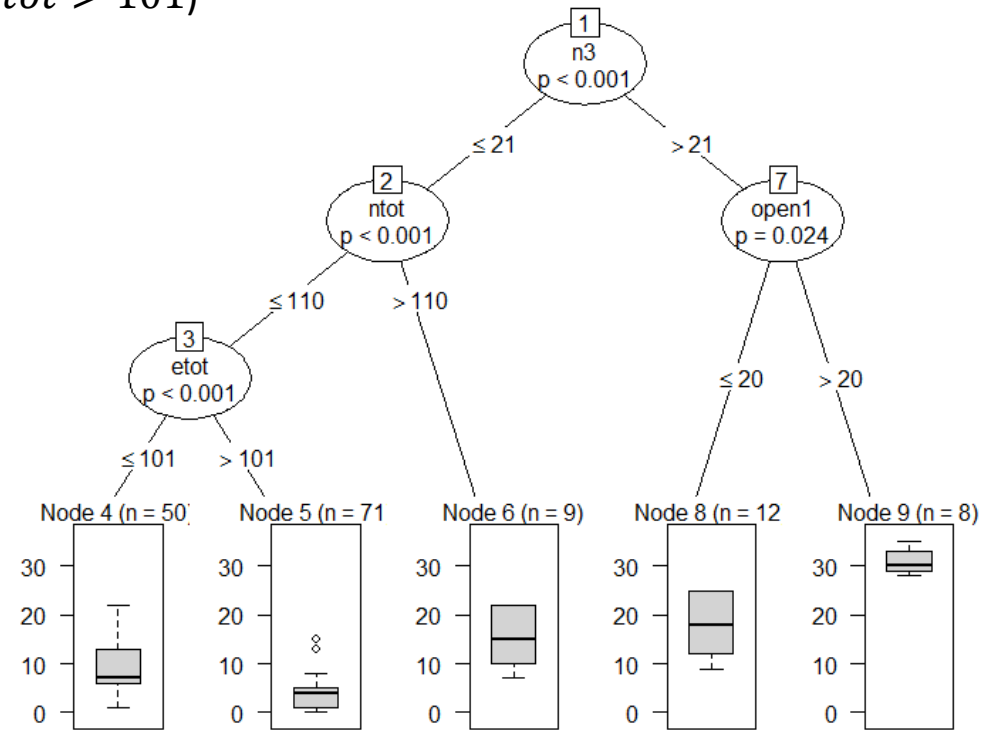
$r_5(\mathrm{x}) = I(n3 \leq 21) \cdot I(ntot \leq 110) \cdot I(etot > 101)$

$r_6(\mathrm{x}) = I(n3 \leq 21) \cdot I(ntot > 110)$

$r_7(\mathrm{x}) = I(n3 > 21)$

$r_8(\mathrm{x}) = I(n3 > 21) \cdot I(open1 \leq 20)$

$r_9(\mathrm{x}) = I(n3 > 21) \cdot I(open1 > 20)$



Rule generation

$r_2(\mathrm{x}) = I(n3 \leq 21)$

$r_3(\mathrm{x}) = I(n3 \leq 21) \cdot I(ntot \leq 110)$

$r_4(\mathrm{x}) = I(n3 \leq 21) \cdot I(ntot \leq 110) \cdot I(etot \leq 101)$

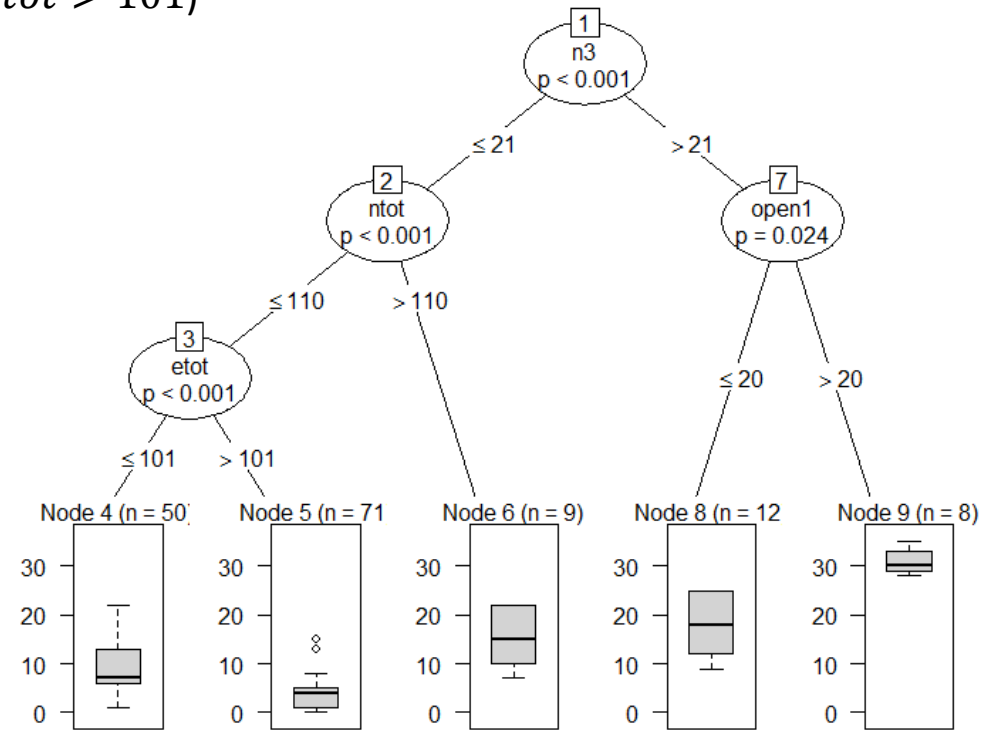
$r_5(\mathrm{x}) = I(n3 \leq 21) \cdot I(ntot \leq 110) \cdot I(etot > 101)$

$r_6(\mathrm{x}) = I(n3 \leq 21) \cdot I(ntot > 110)$

~~$r_7(\mathrm{x}) = I(n3 > 21)$~~

$r_8(\mathrm{x}) = I(n3 > 21) \cdot I(open1 \leq 20)$

$r_9(\mathrm{x}) = I(n3 > 21) \cdot I(open1 > 20)$



Example: Depression data

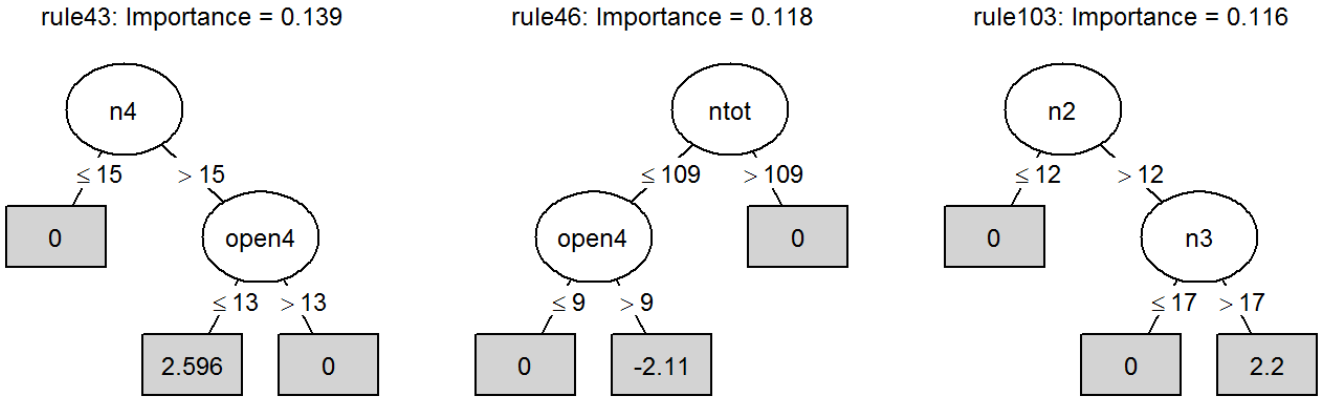
```
library("pre")
data("carrillo")
```

```
set.seed(3432)
bdi.ens <- pre(bdi ~ . , data = carrillo)
bdi.ens
```

```
##
## Final ensemble with cv error within 1se of minimum:
##   lambda =  0.6932192
##   number of terms = 9
##   mean cv error (se) = 33.76187 (6.695341)
##
##   cv error type : Mean-Squared Error
##
##          rule  coefficient              description
##   (Intercept)   9.00928263                    <NA>
##        rule43   2.59610067   n4 > 15 & open4 <= 13
##       rule103   2.19966580       n2 > 12 & n3 > 17
##        rule46  -2.11046811 ntot <= 109 & open4 > 9
##        rule55  -1.34941743   ntot <= 110 & e6 > 14
##        rule27  -1.33961480   n3 <= 22 & etot > 101
##        rule57  -0.58508689   n3 <= 17 & open4 > 10
##        rule68  -0.34947264                n1 <= 20
##            n3   0.11996095       2 <= n3 <= 30.225
##        rule81  -0.06191138                n1 <= 23
```

Example: Depression data

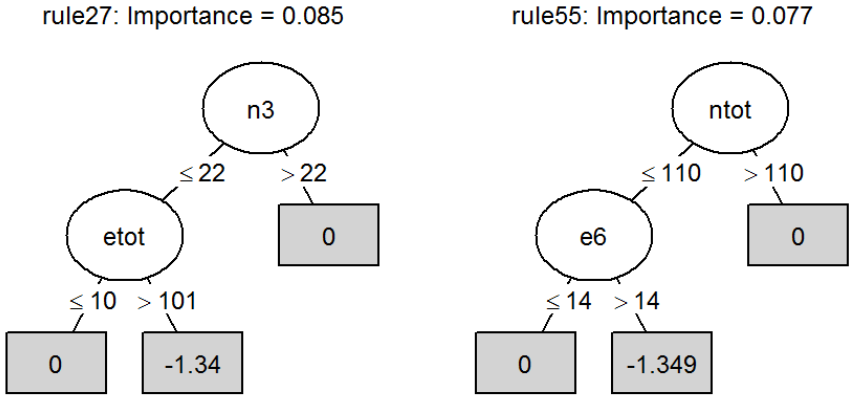
```
plot(bdi.ens)
```



Linear effect of n3

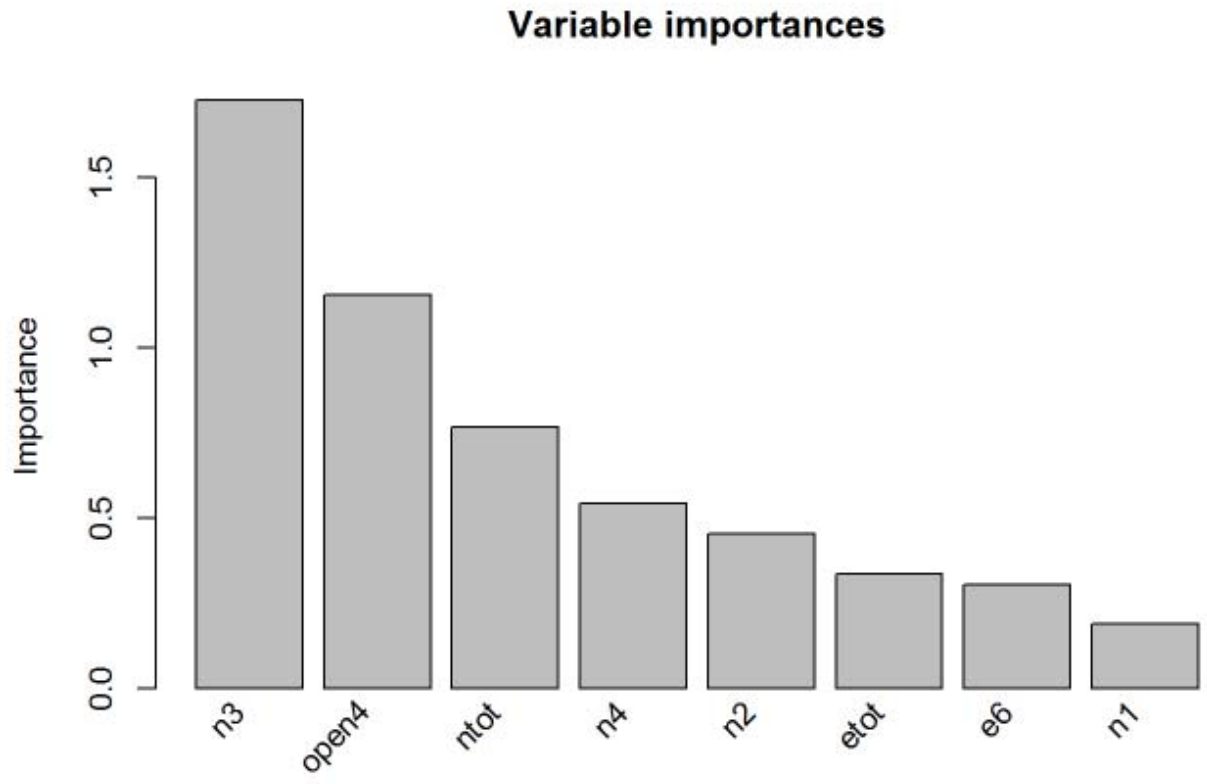
Coefficient = 0.12

Importance = 0.101



Example: Depression data

```
imps <- importance(bdi.ens)
```



Example: Depression data

```
imps
```

```
## $varimps
##   varname       imp
## 1      n3 1.7264522
## 2   open4 1.1538112
## 3    ntot 0.7675468
## 4      n4 0.5429385
## 5      n2 0.4533182
## 6    etot 0.3344724
## 7      e6 0.3032279
## 8      n1 0.1884188
##
## $baseimps
##      rule              description        imp coefficient        sd
## 1  rule43   n4 > 15 & open4 <= 13 1.08587706  2.59610067 0.4182723
## 2  rule46 ntot <= 109 & open4 > 9 0.92863776 -2.11046811 0.4400151
## 3 rule103       n2 > 12 & n3 > 17 0.90663641  2.19966580 0.4121701
## 4      n3       2 <= n3 <= 30.225 0.79210776  0.11996095 6.6030470
## 5  rule27   n3 <= 22 & etot > 101 0.66894480 -1.33961480 0.4993561
## 6  rule55   ntot <= 110 & e6 > 14 0.60645582 -1.34941743 0.4494205
## 7  rule57   n3 <= 17 & open4 > 10 0.29310766 -0.58508689 0.5009643
## 8  rule68                n1 <= 20 0.16723856 -0.34947264 0.4785455
## 9  rule81                n1 <= 23 0.02118027 -0.06191138 0.3421062
```

Resolution

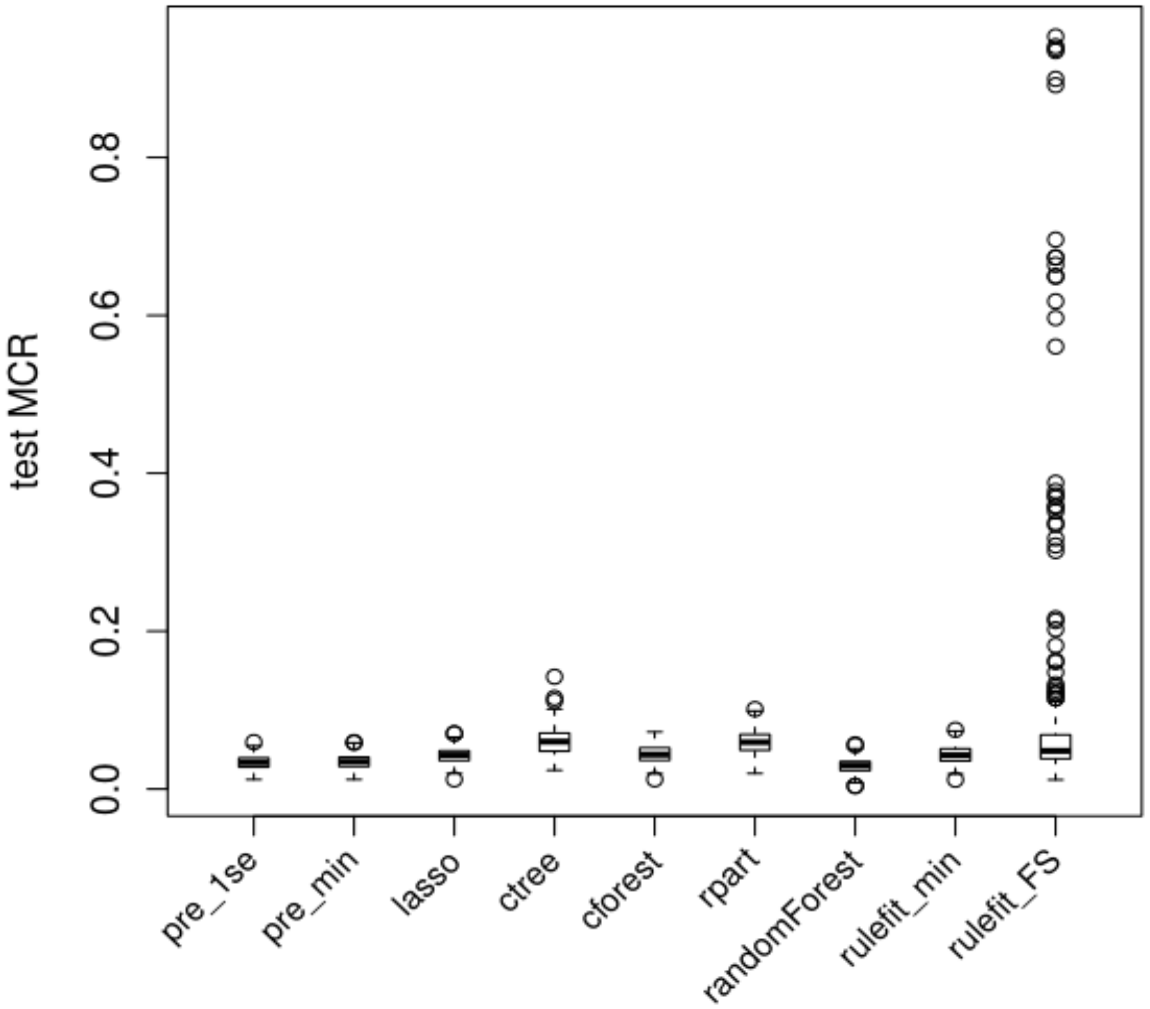
Does pre kill the bad?

Does the good survive?

The good survived (predictive accuracy)

Ionosphere data

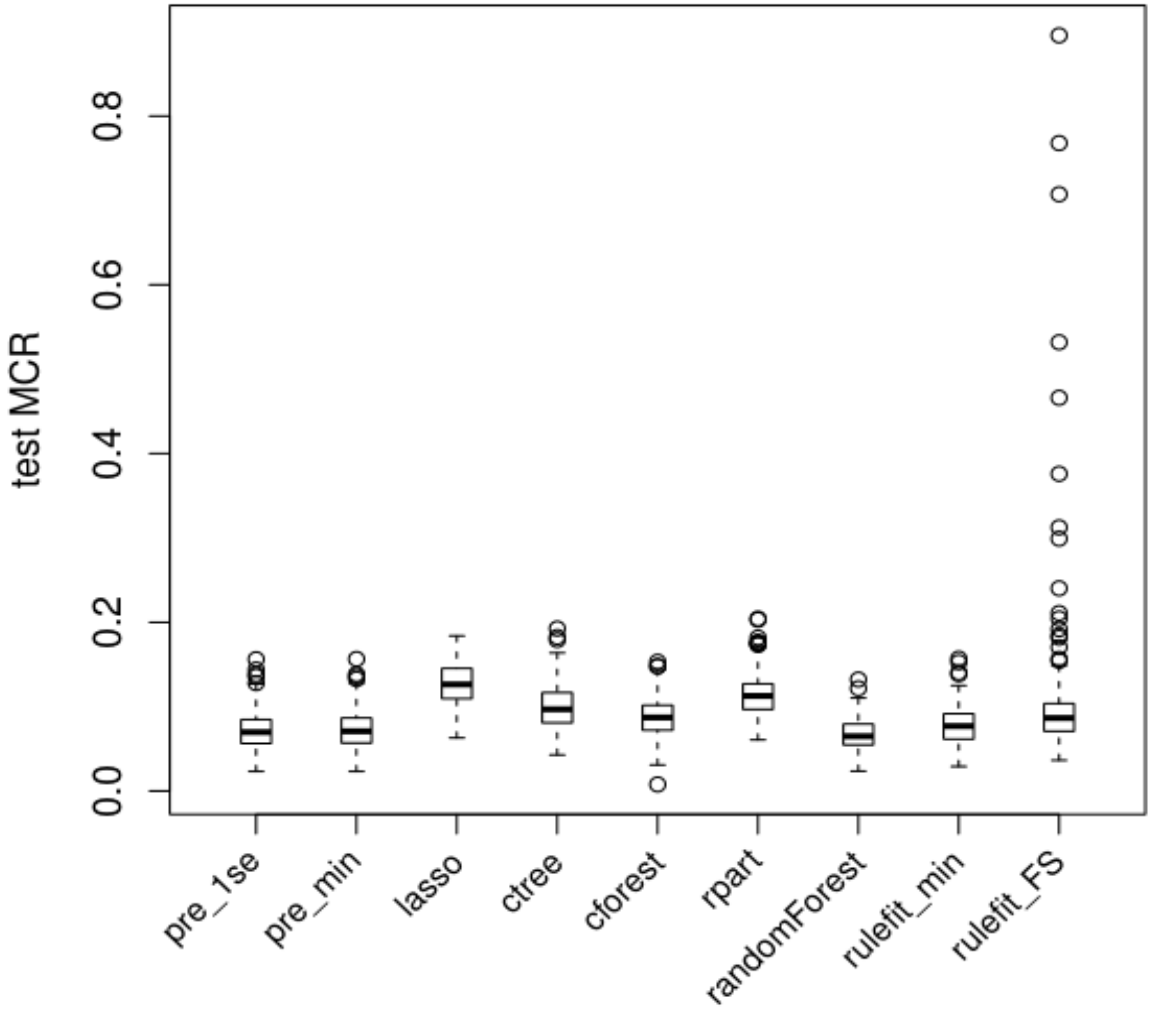
(250 bootstrap samples)



The good survived (predictive accuracy)

Breast Cancer data

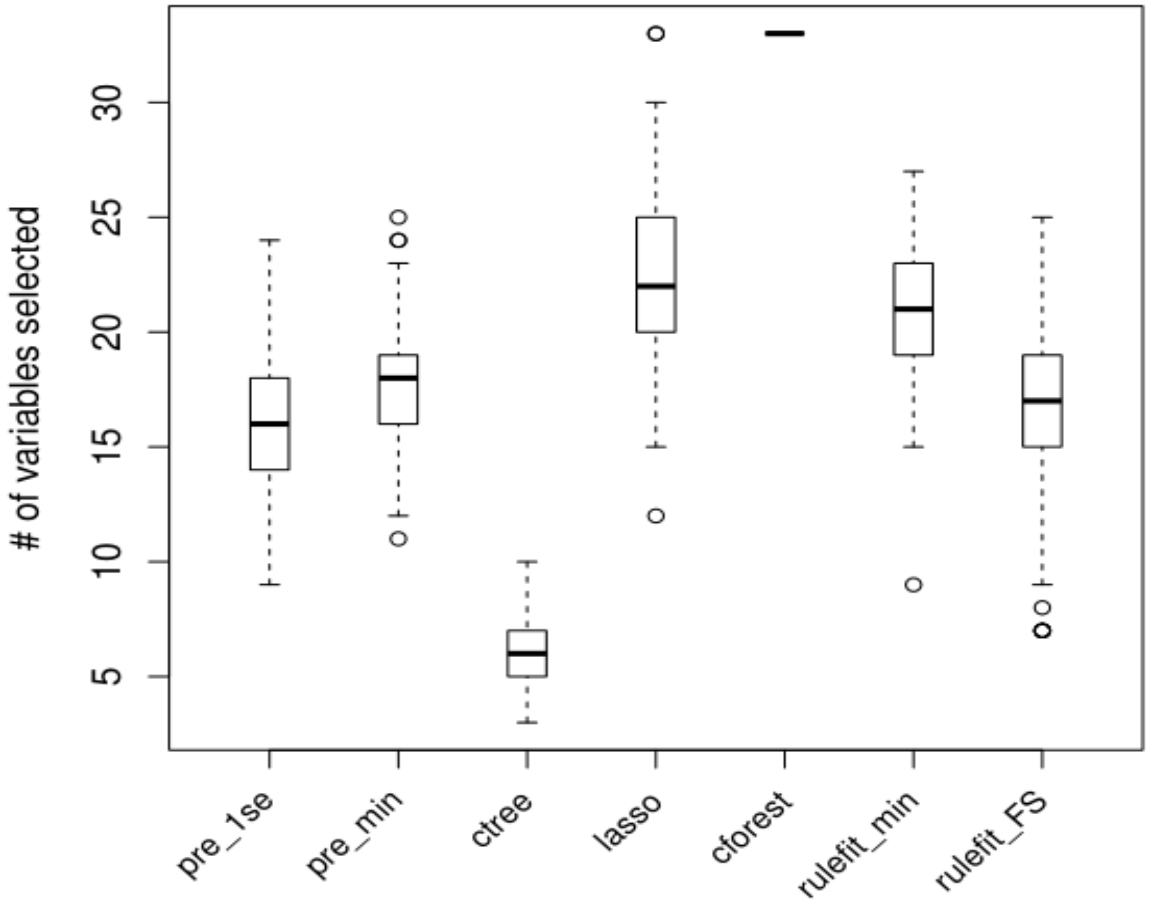
(250 bootstrap samples)



Killed the bad?
(complexity)

Ionosphere data

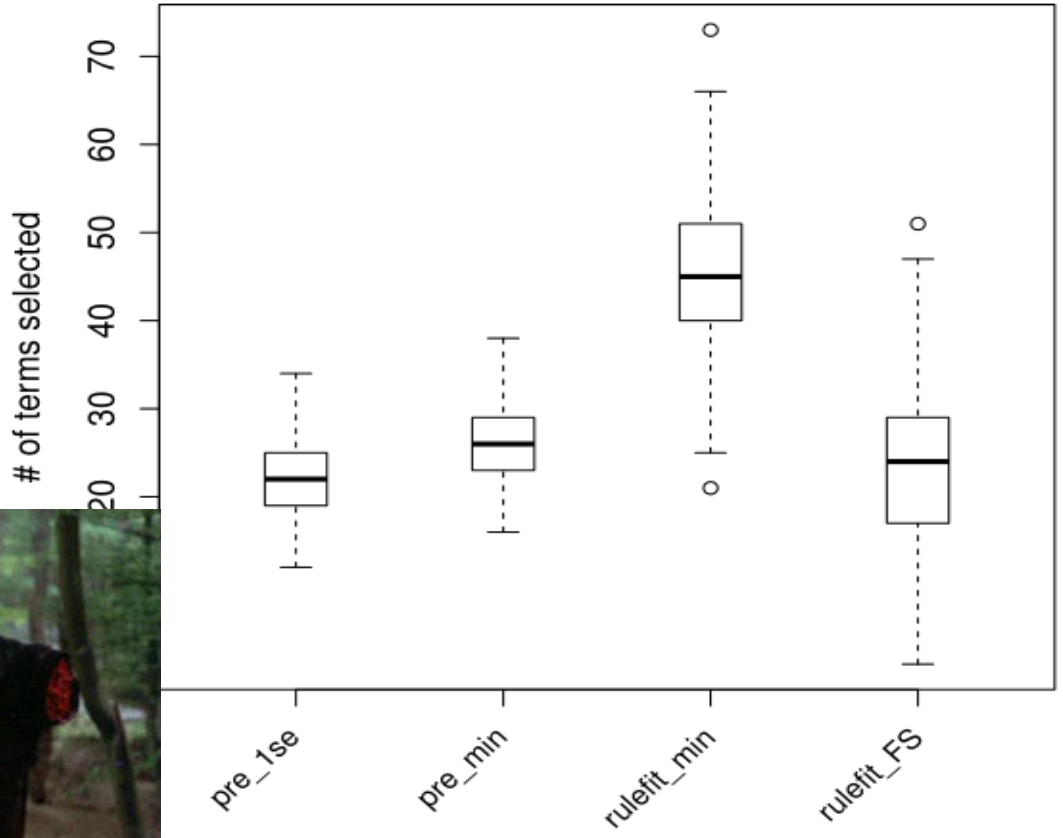
(250 bootstrap samples)



Killed the bad?
(complexity)

Breast Cancer data

(250 bootstrap samples)



Clustered data structures

Final ensemble (additive model):

Rule, e.g.:
$f_m(\mathbf{x}) = I(x_1 > 5) \cdot I(x_4 \leq 10)$

$$F(\mathbf{x}) = \hat{a}_0 + \sum_{m=1}^{M} \hat{a}_m f_m(\mathbf{x})$$

Linear term, e.g.:
$f_m(\mathbf{x}) = x_2$

What if data are clustered?

-> Ignore clustered structure

-> Sample level-2 instead of level-1 units

-> Estimate random effects

Mixed-effect PRE estimation

$$F(\mathbf{x}, \mathbf{z}) = a_0 + \sum_{m=1}^{M} a_m f_m(\mathbf{x}) + \mathbf{z}^{\mathrm{T}}\mathbf{b}$$

Estimation steps:

0) Initialize by assuming $\hat{\mathbf{b}} = \mathbf{0}$
1) Estimate $f_m(\mathbf{x})$
2) Estimate $\mathbf{a}$ given current $\hat{\mathbf{b}}$
3) Estimate $\mathbf{b}$ given current $\hat{\mathbf{a}}$
4) Repeat steps 2 & 3 until convergence

Full mixed effects: random effects are estimated in steps 1 and 3

Partial mixed-effects: random effects assumed 0 in step 1, only estimated in step 3

PRE methods for clustered data

pre: ignores multilevel structure

pre_cs: accounts for multilevel structure through cluster-level sampling

pre_m_part: random effects estimated in final ensemble only

pre_m_full: random effects estimated in rule induction and estimation of final ensemble

R package **`premixed`**: fits mixed-effects PREs

https://github.com/marjoleinF/premixed

Global Adult Tobacco Survey

- WHO survey on tobacco consumption
- Data from Indonesia (N = 709)
- Outcome: Smoker (1) vs non-smoker (0)
- Predictors:
  - age, gender, level of education
  - belief that smoking causes illness for oneself, for other people around
- Level-2 unit: multistage sampling units (used to produce nationally representative sample)

GATS data: Results

	AUC M (SE)	# terms M	# vars M
pre	.9243 (.006)	37.7	4.7
pre_cs	.9233 (.006)	49.4	5.0
pre_m_part	.9305 (.006)	40.8	5.0
pre_m_full	.9309 (.006)	28.1	4.7
glmertree	.9089 (.016)	-	-
glmer	.9200 (.007)	-	-
Note: All values estimated based on 10-fold CV.			

Simulation

Aim: Compare accuracy, complexity and computation time

Prediction rule ensembles:

- **pre:** ignores multilevel structure
- **pre_cs**: cluster-level sampling
- **pre_m_part:** random effects estimated in final ensemble only
- **pre_m_full**: random effects estimated in rule induction and estimation of final ensemble

Other methods:

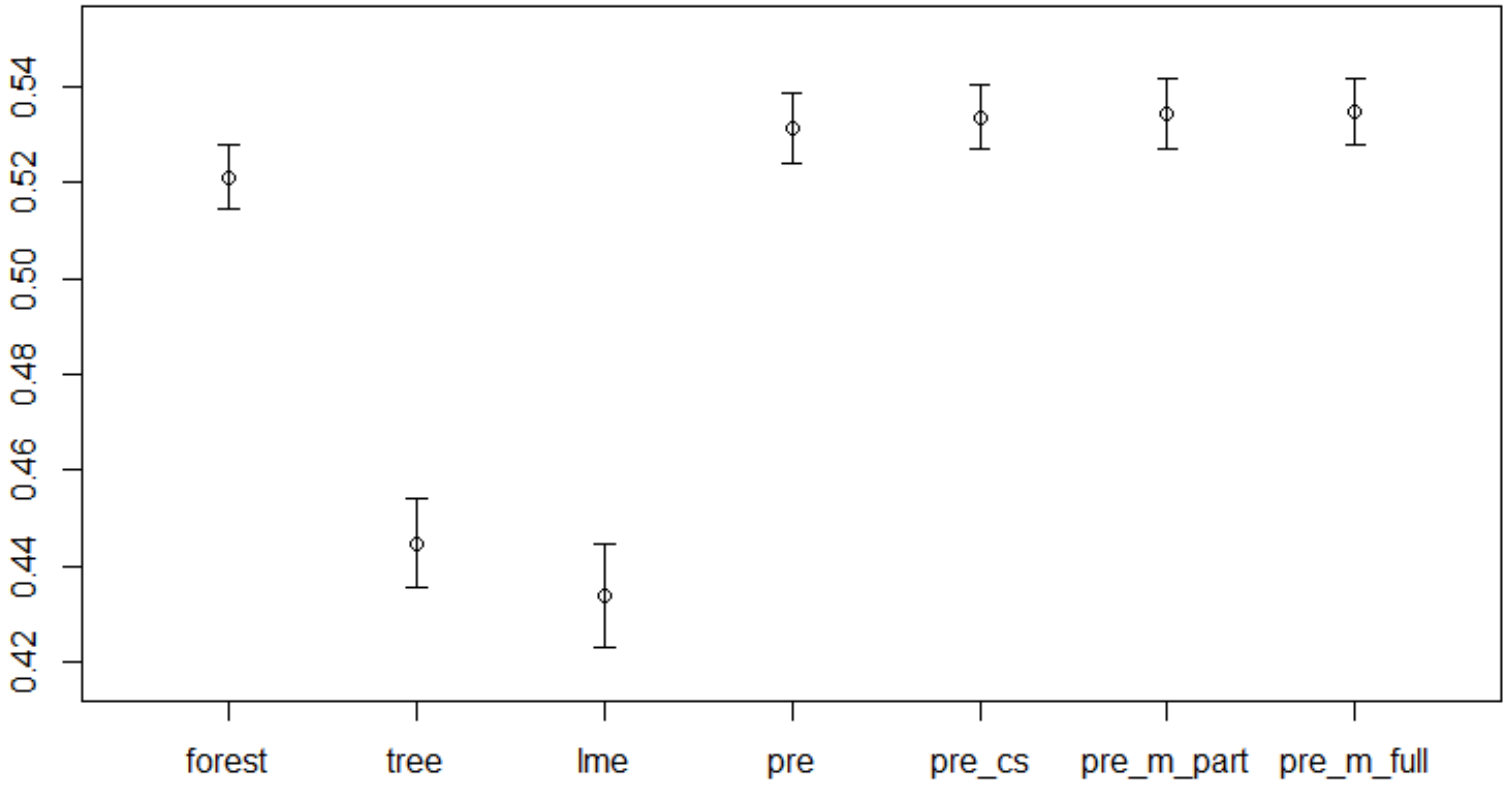
- **lme**: linear mixed-effects model
- **tree:** mixed-effects regression tree
- **forest**: random forest

Simulation: Data-generation

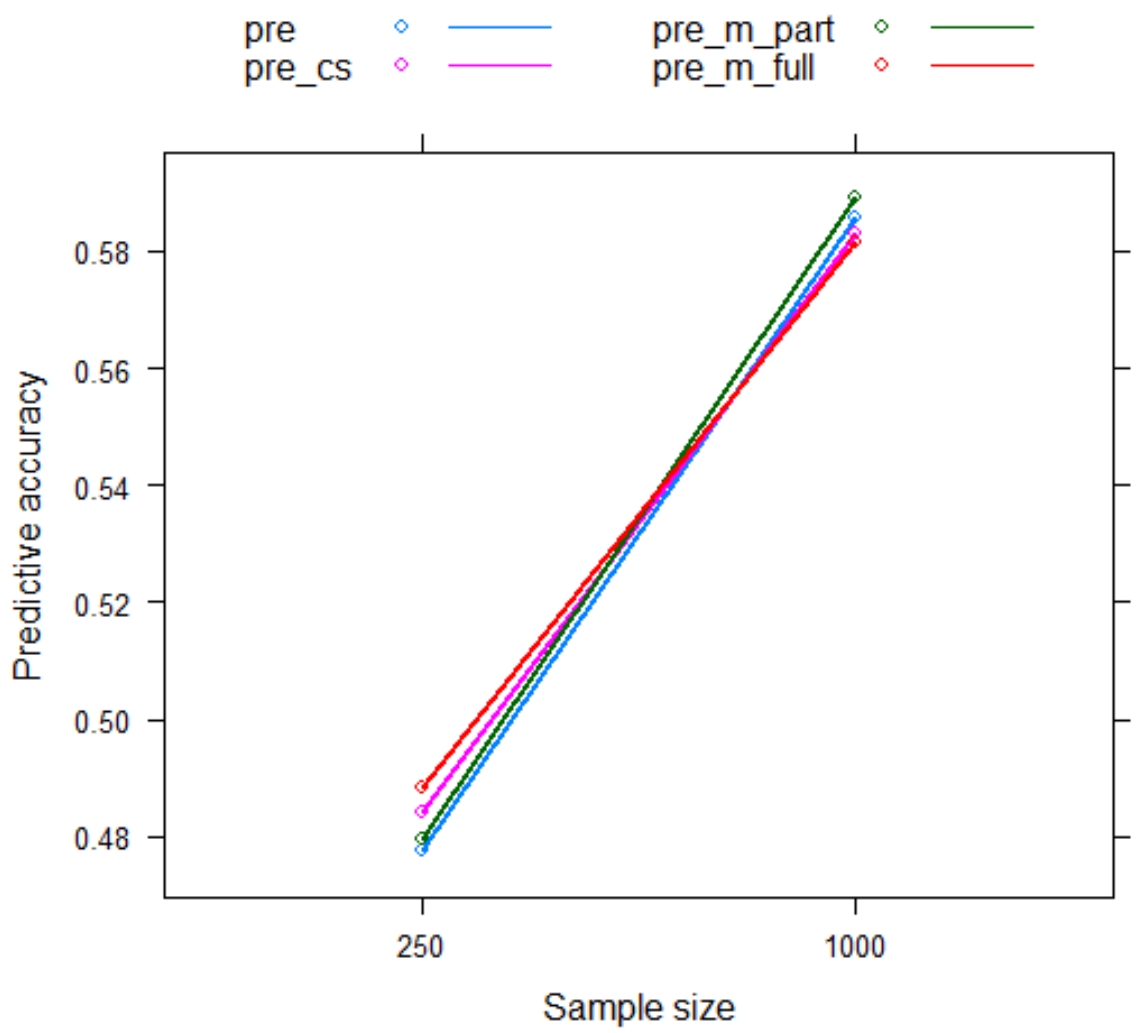
true model:	Linear, rules, rules + linear, tree
sample size	Small (250), large (1000)
# predictor variables	Small (20), large (50)
# of clusters	Small (10), large (50)
variance of random effects	Small (ICC≈.1), large (ICC ≈.25)
dependence between cluster membership and predictor variables	Independent (rho = 0), Dependent (rho ≈ .30)

Predictive accuracy

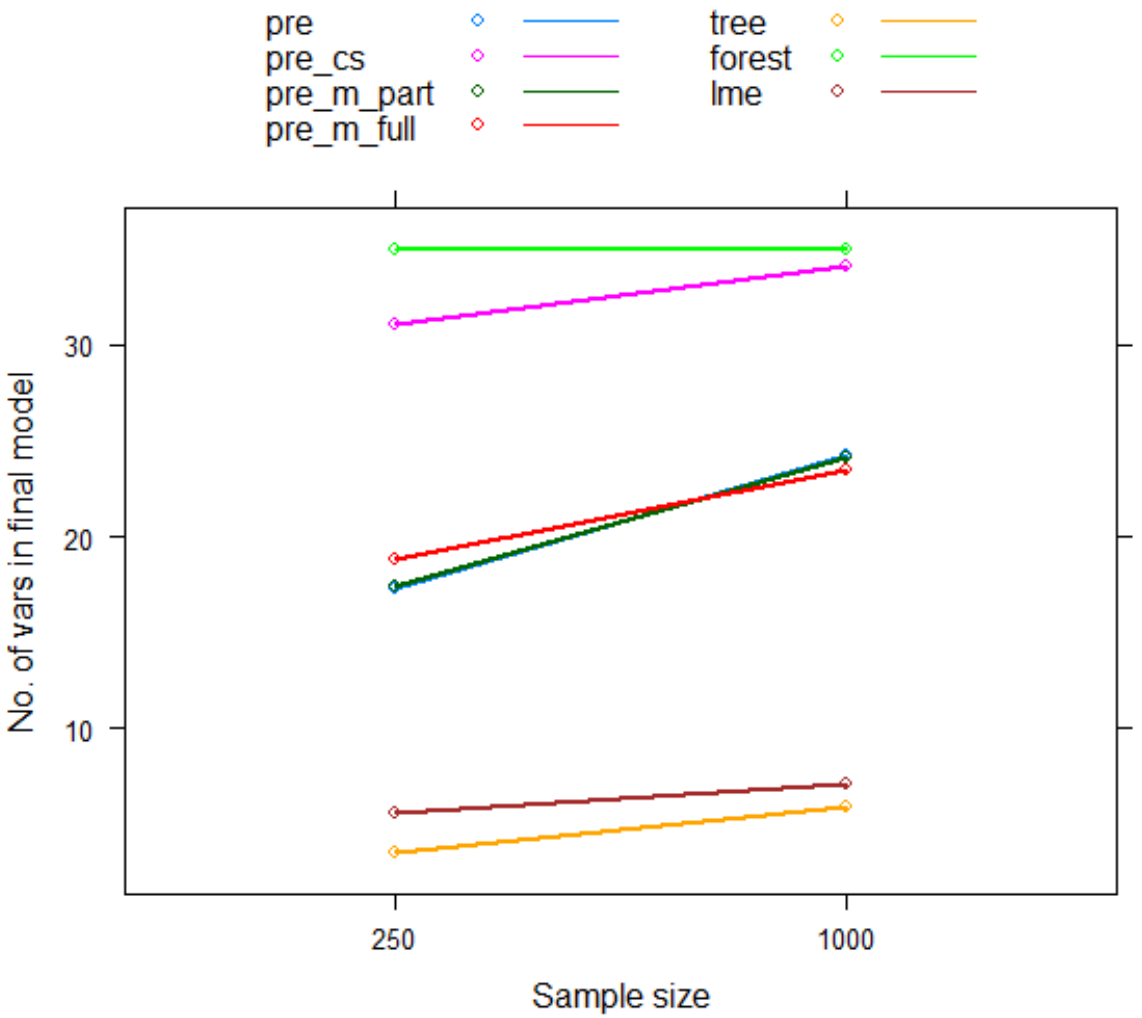
Correlation between true and predicted y in test data (error bars represent M +/- SD):



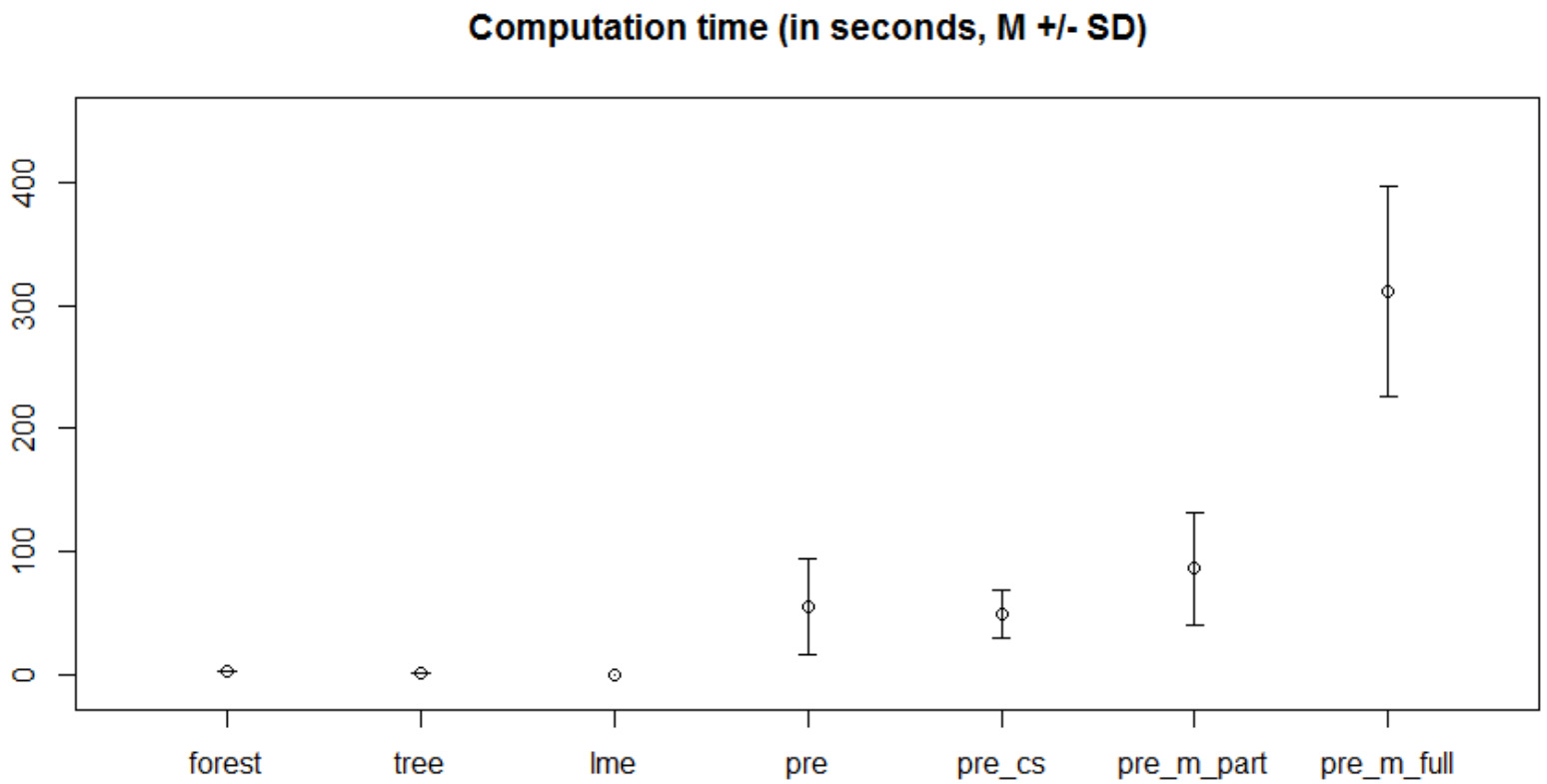
Predictive accuracy



Complexity


pre
pre_cs
pre_m_part
pre_m_full
tree
forest
lme
No. of vars in final model
30
20
10
250
1000
Sample size

Computation time



Discussion

Good news: PREs provide comparable accuracy and lower complexity than random forests and original RuleFit implementation

Adding random-effects estimation not clearly beneficial

- Full mixed-effects PREs may be advantageous for smaller sample sizes; partial mixed-effects PREs more accurate for larger sample sizes
- But: mixed-effects estimation computationally very expensive
- But: complexity not reduced
  - Better selection algorithms than lasso?

Cluster-level sampling hardly affects accuracy but increases complexity substantially

Thank you for your attention!

References

Fokkema, M. (2017). pre: An R Package for Fitting Prediction Rule Ensembles. Working paper, arXiv:1707.07149.

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

Packages:

pre: Prediction Rule Ensembles. https://CRAN.R-project.org/package=pre

premixed: Mixed-Effects Prediction Rule Ensembles. https://github.com/marjoleinF/premixed

m.fokkema@fsw.leidenuniv.nl