Random Forests: Myths and Facts

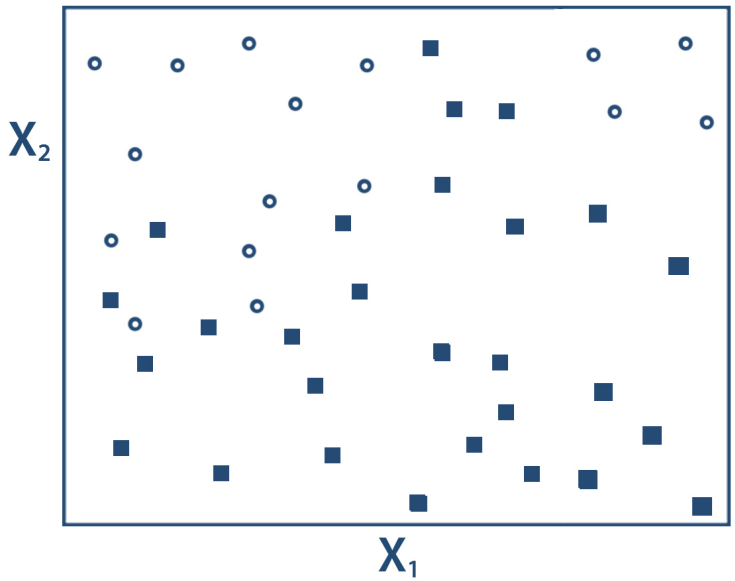
Marvin N. Wright

Leibniz Institute for Prevention Research & Epidemiology – BIPS
University of Bremen
University of Copenhagen

November 2021

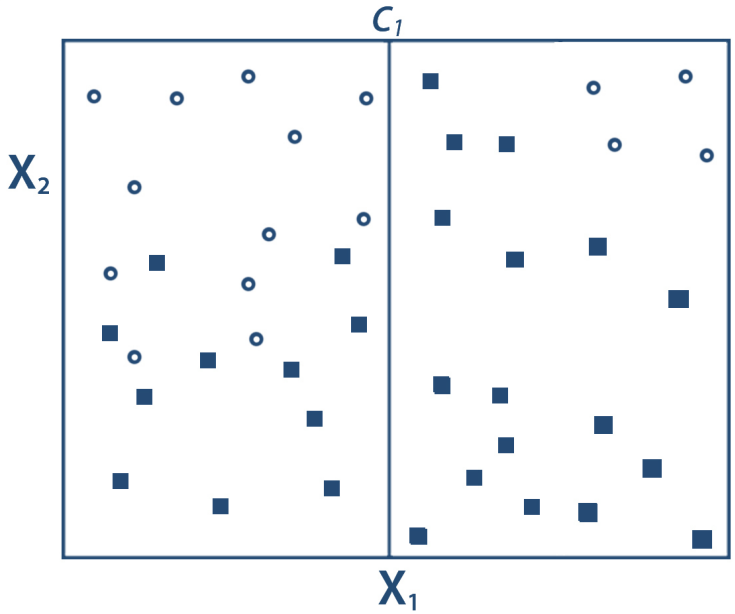
Outline

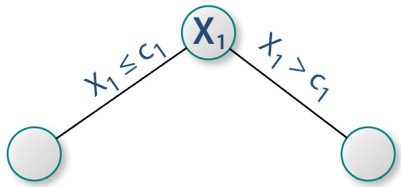
Random forests



Breiman 2001 Mach Learn 45:5

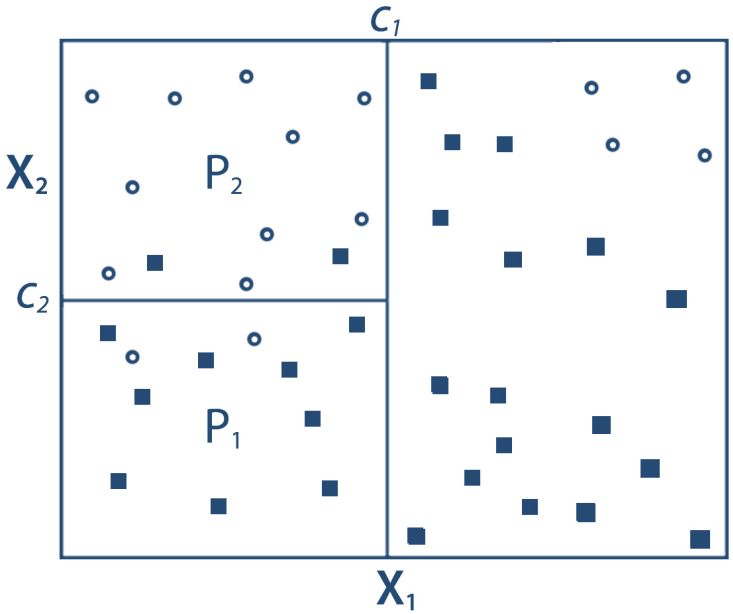
Random forests

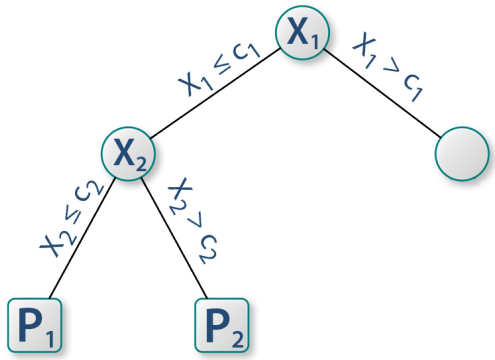




Breiman 2001 Mach Learn 45:5

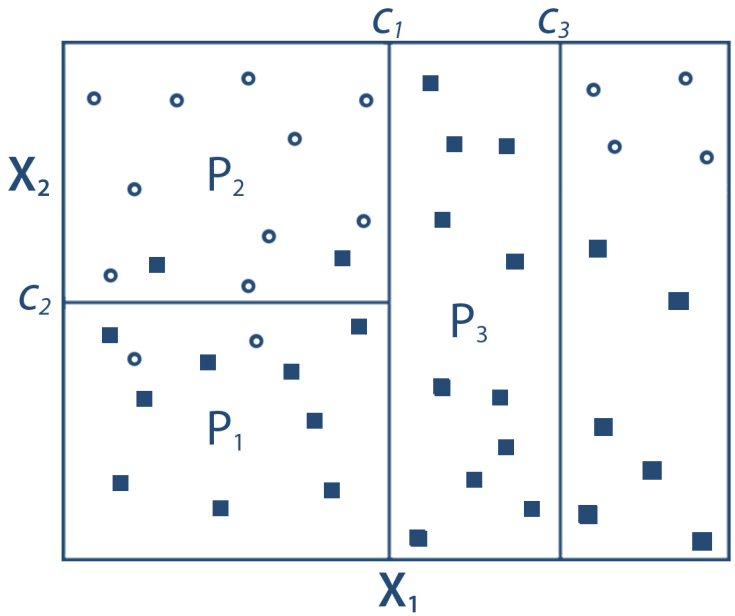
Random forests

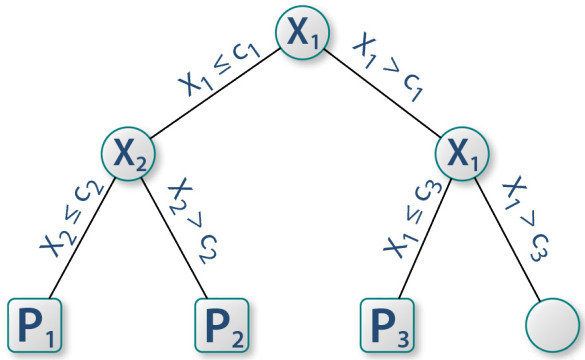




Breiman 2001 Mach Learn 45:5

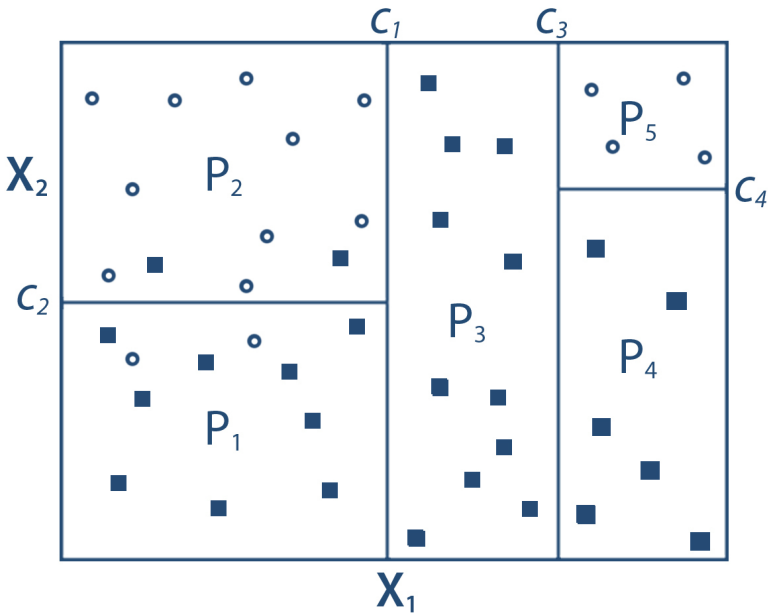
Random forests

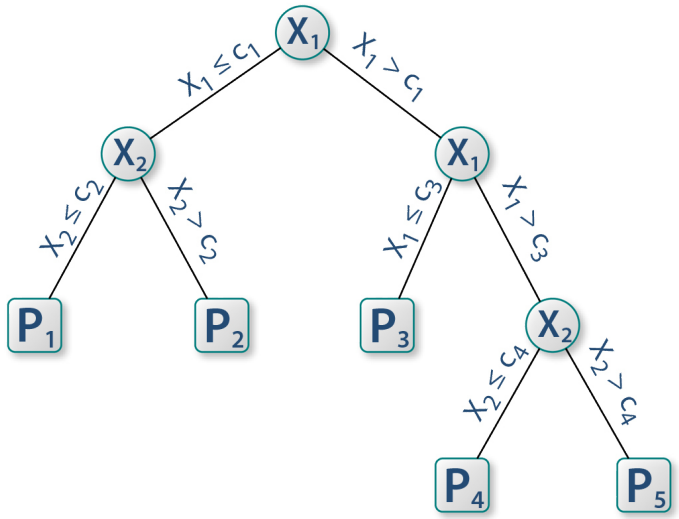




Breiman 2001 Mach Learn 45:5

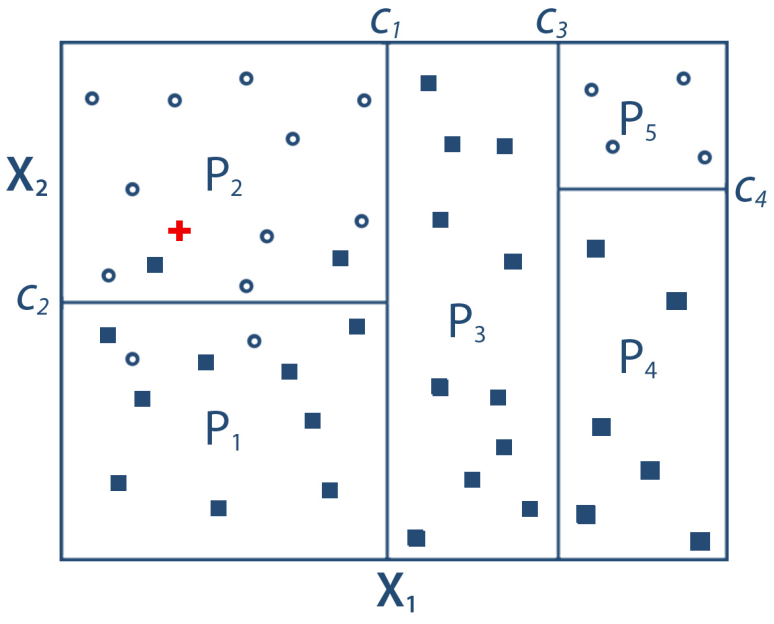
Random forests

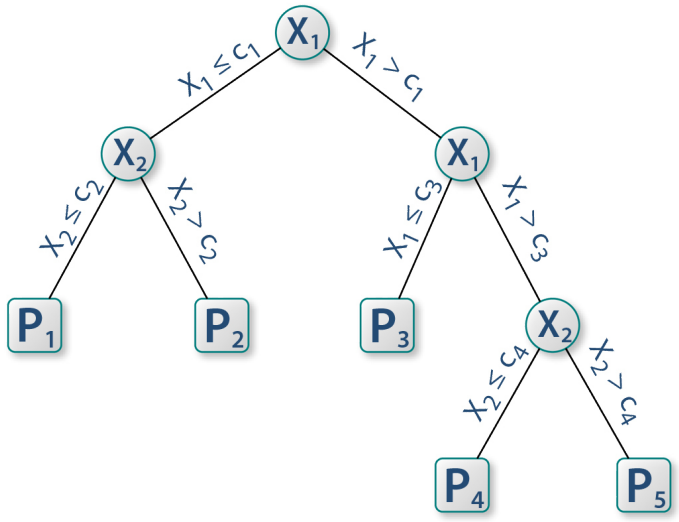




Breiman 2001 Mach Learn 45:5

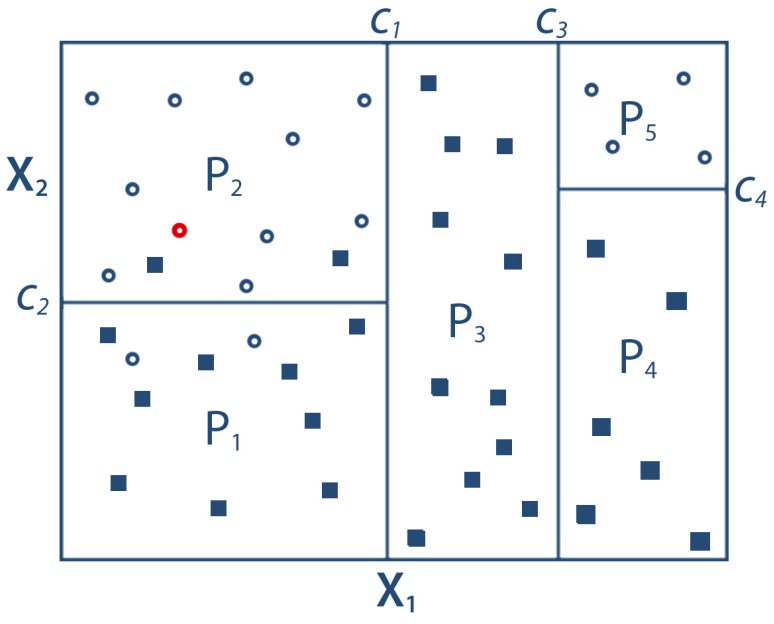
Random forests

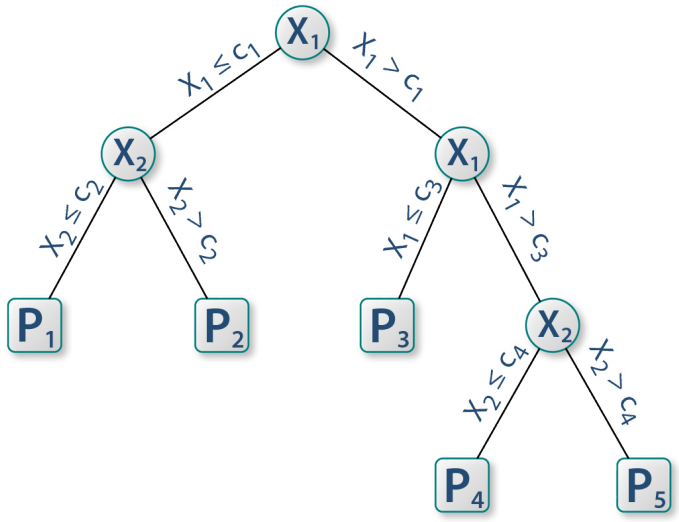




Breiman 2001 Mach Learn 45:5

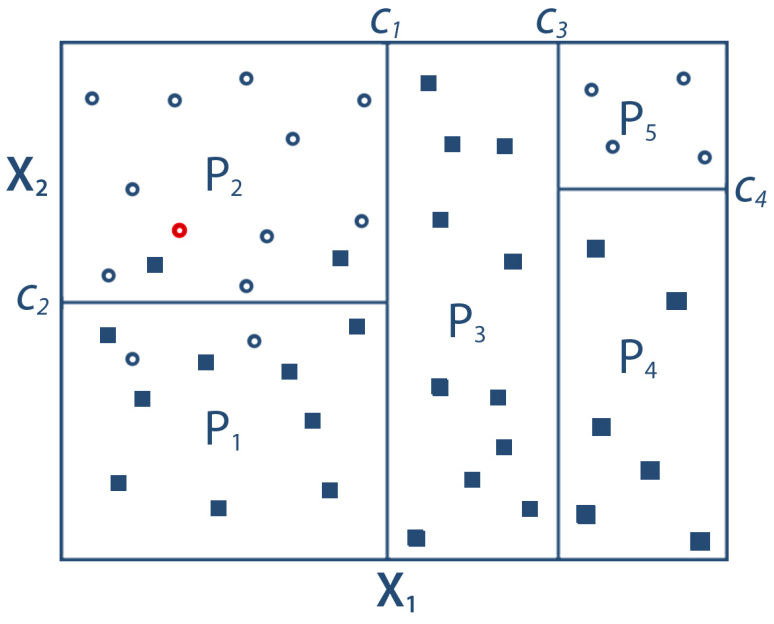
Random forests

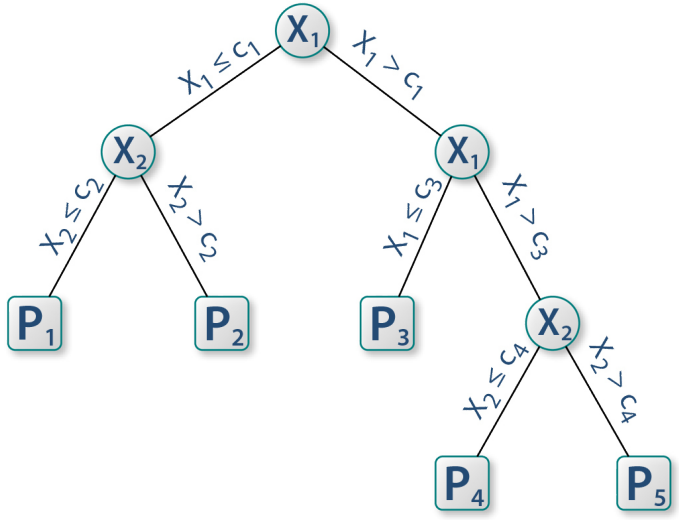


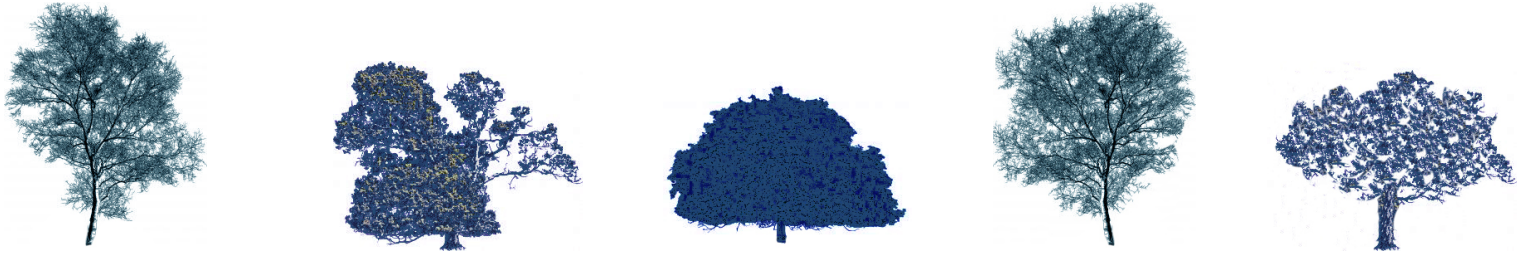


Breiman 2001 Mach Learn 45:5

Random forests

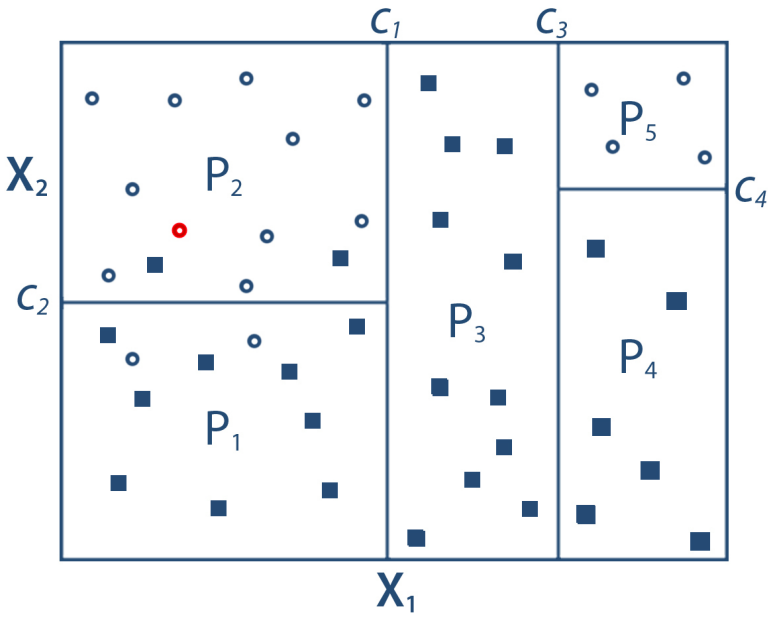


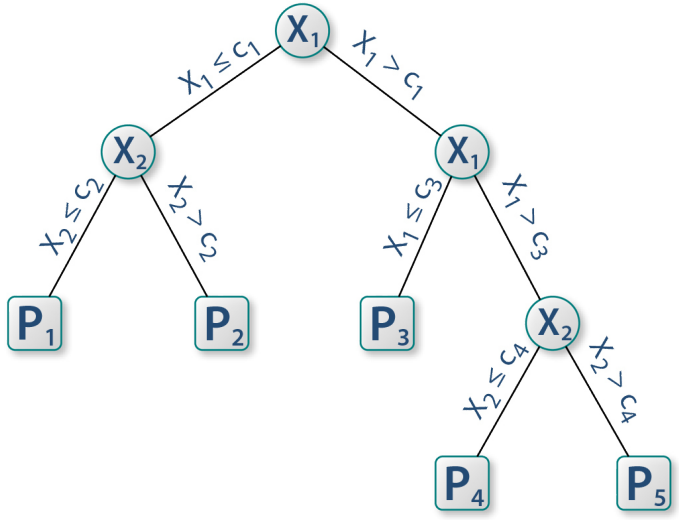


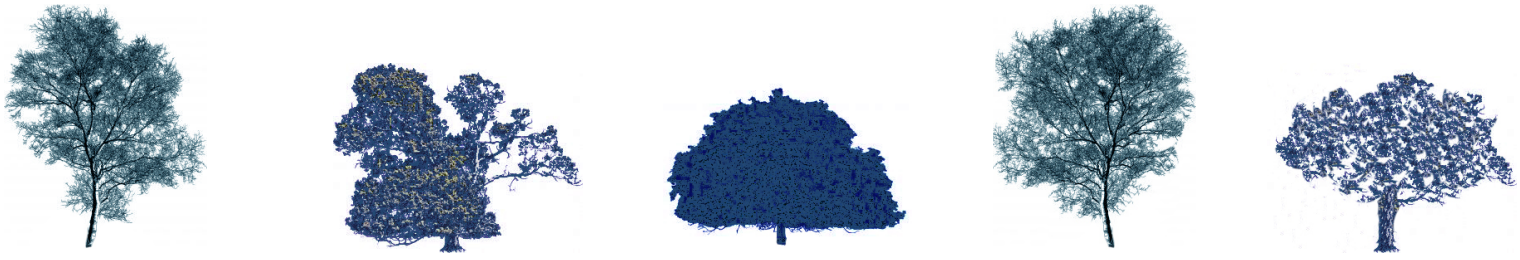


Breiman 2001 Mach Learn 45:5

Random forests

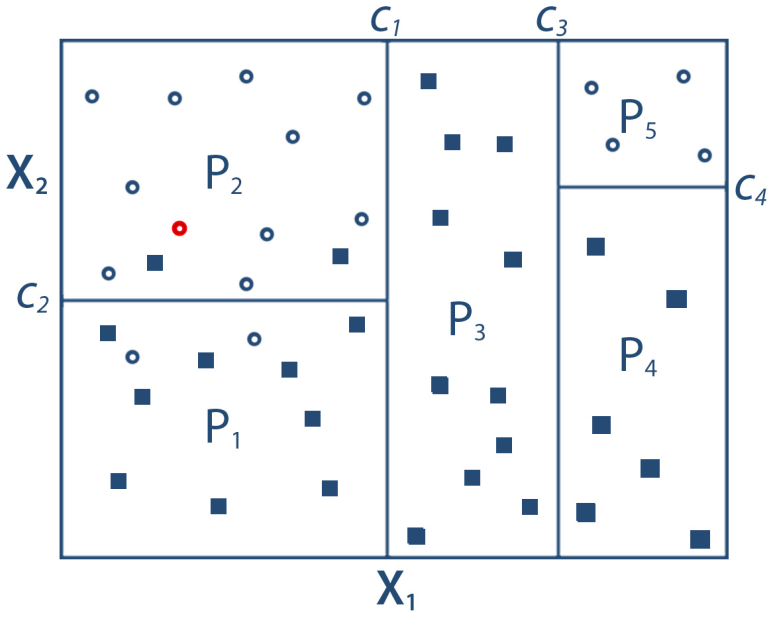


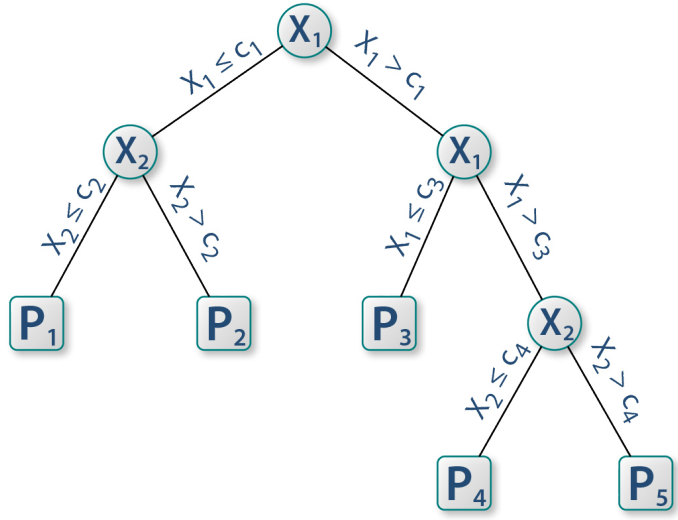


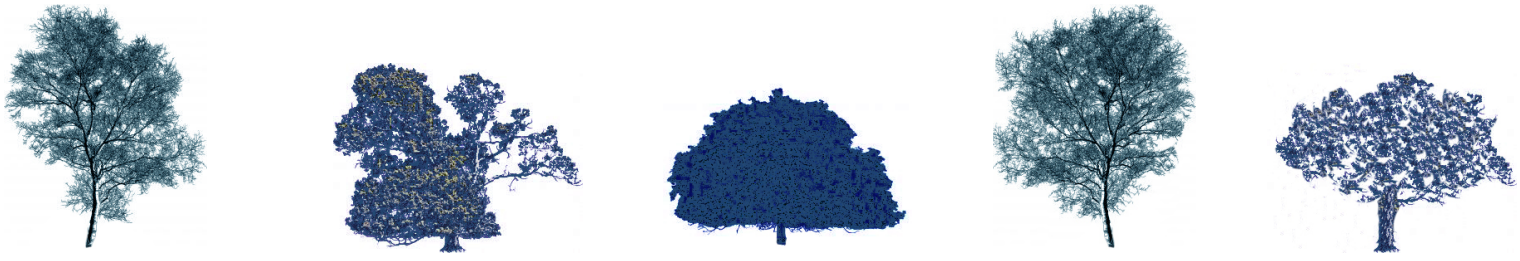


1 1 0 1 0

Random forests







1 1 0 1 0

1 according to majority vote

Random forests
Algorithm

Step 1

Draw bootstrap sample or subsample

Step 2

Grow tree

Step 3

At each node, randomly select features (`mtry` value)

Step 4

Repeat steps 1-3, average over all trees

Random forests
Algorithm

Step 1

Draw bootstrap sample or subsample

Step 2

Grow tree

Step 3

At each node, randomly select features (`mtry` value)

Step 4

Repeat steps 1-3, average over all trees

Random forests
Tuning parameters

Number of trees

- Usual default: 500
- Use more trees for high dimensional datasets

`mtry` value

Number of features selected as splitting candidates in nodes

- Usual default: $\sqrt{p}$, where $p = \#\text{features}$
- For large p use at least $p/10$

Terminal node size

Required number of observations in terminal nodes

- Determines tree size
- Typically small for classification, large for regression/survival

Outline

1. Introduction
2. Common Claims
3. Implementations in R
4. Discussion & Conclusion

Common Claims

"no need to scale or recode predictors" "cannot overfit"

"the split variable selection is biased"

"the statistical properties are well understood"

"works well without tuning"

"detects interactions" "works for any kind of data"

"performance is not state of the art"

"works well on high dimensional data" "is an interpretable model"

Common Claims

Claim 1: “works well without tuning”

Common Claims

Claim 1: "works well without tuning"

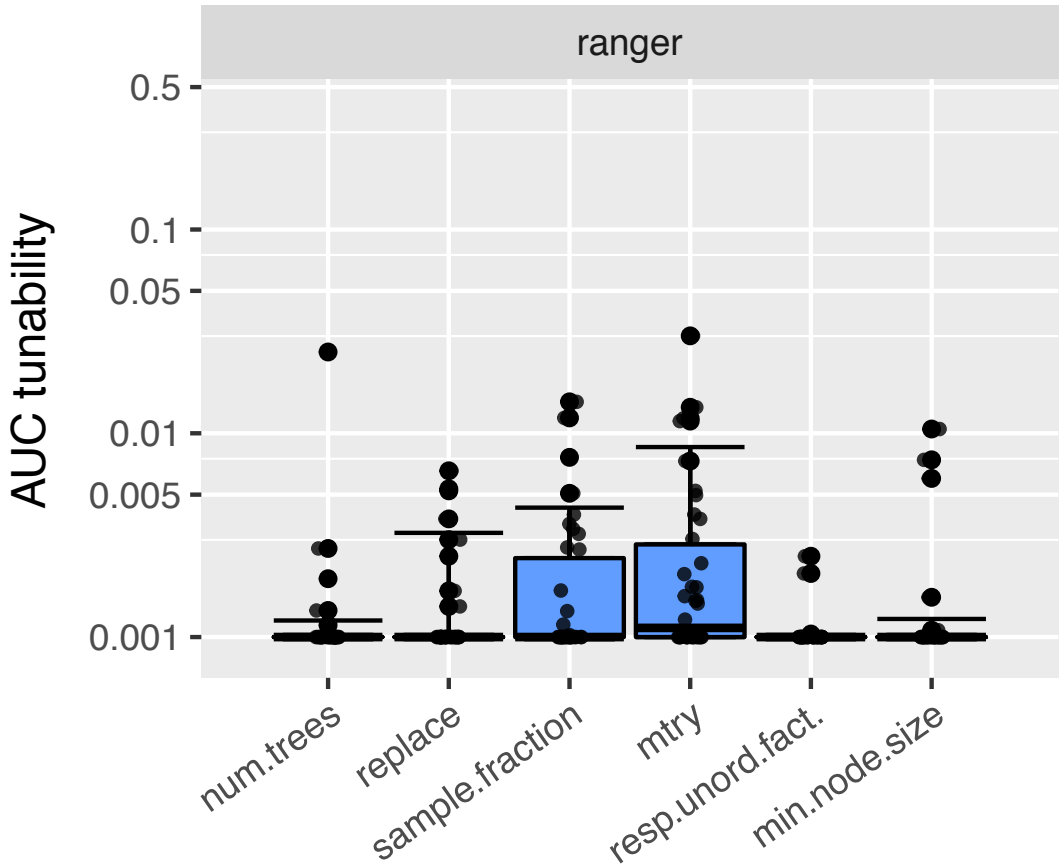
"RF is the algorithm with the smallest tunability."

— Probst et al. 2019a

They compared `ranger`, `glmnet`, `rpart`, `kknn`, `e1071::svm`, `xgboost`.

Common Claims

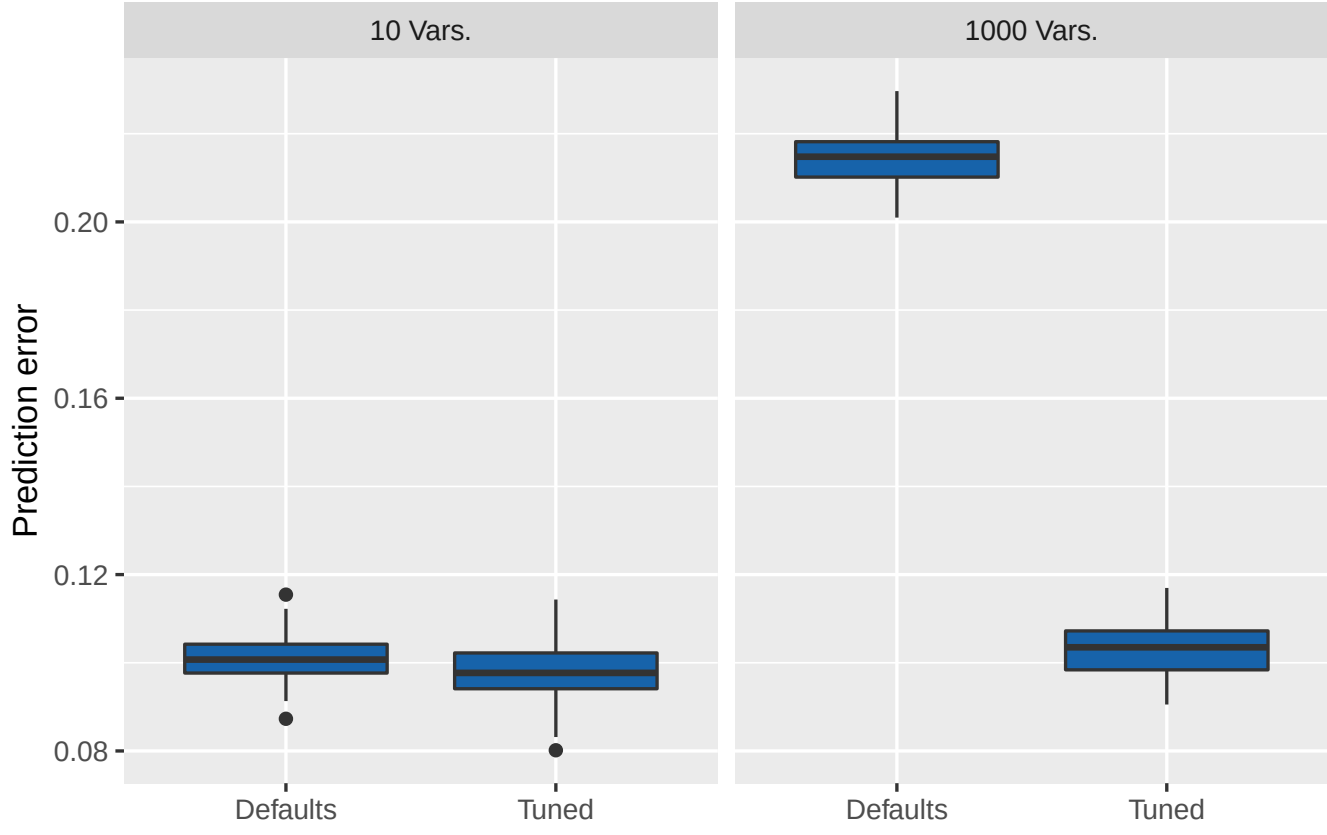
Claim 1: "works well without tuning"



Probst et al. 2019a J Mach Learn Res 20:1

Common Claims

Claim 1: “works well without tuning”



Common Claims

- ✓ Works well without tuning
  - ✗ Exception: High dimensional data, low signal-to-noise ratio

Common Claims

Claim 2: “no need to scale or recode predictors”

Common Claims

Claim 2: "no need to scale or recode predictors"

Scaling

RF invariant to monotonic transformations
$\Rightarrow$ scale-invariant

Example

Original	4.6	4.7	4.9	5.0	5.1	5.4
Scaled	-1.2	-0.9	-0.2	0.2	0.5	1.6
Logarithm	1.53	1.55	1.59	1.61	1.63	1.69

Common Claims

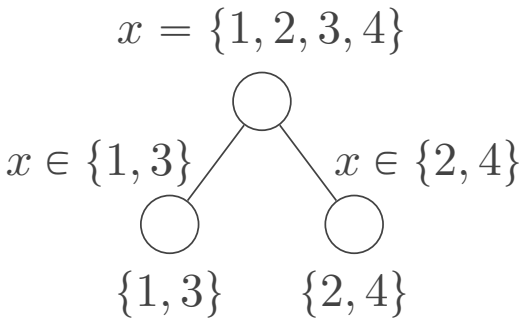
Claim 2: "no need to scale or recode predictors"

Categorical predictors

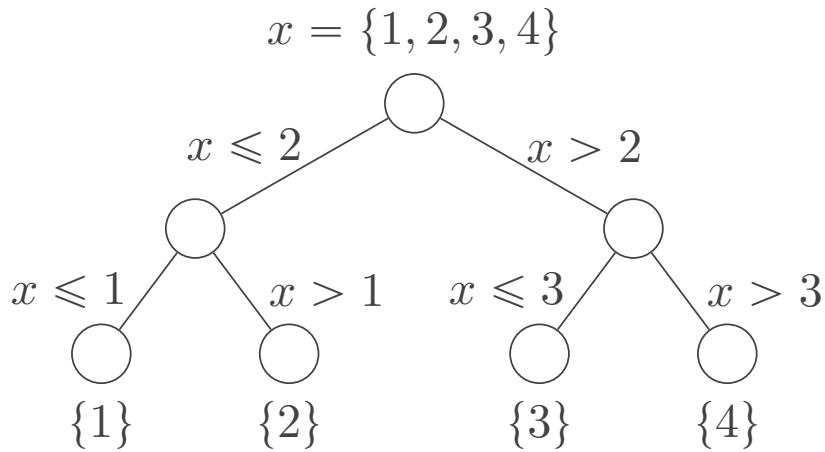
$x = \{1, 2, 3, 4\}$

Aim: Separate odd and even digits

Standard approach



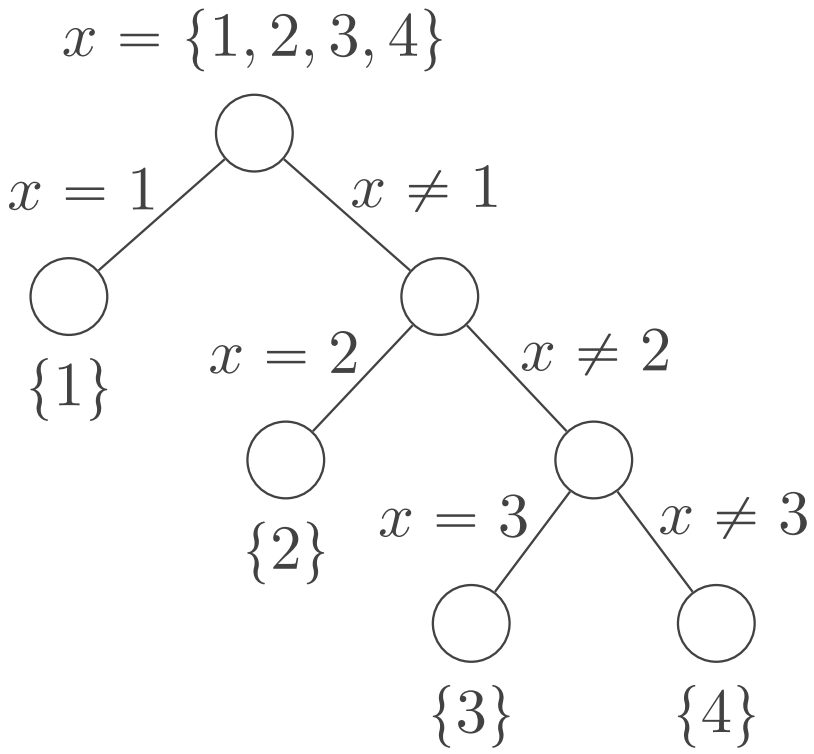
Naïve approach



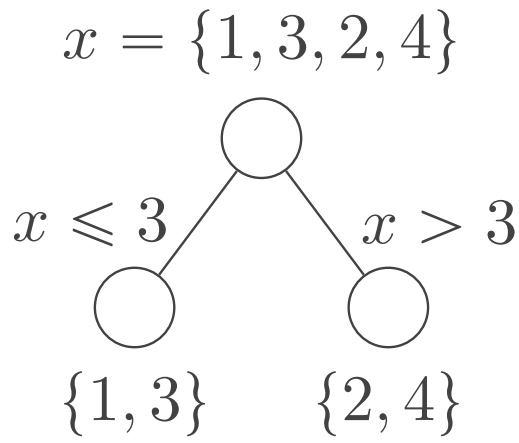
Common Claims

Claim 2: "no need to scale or recode predictors"

Dummy coding



A priori ordering



Order by average outcome

Common Claims

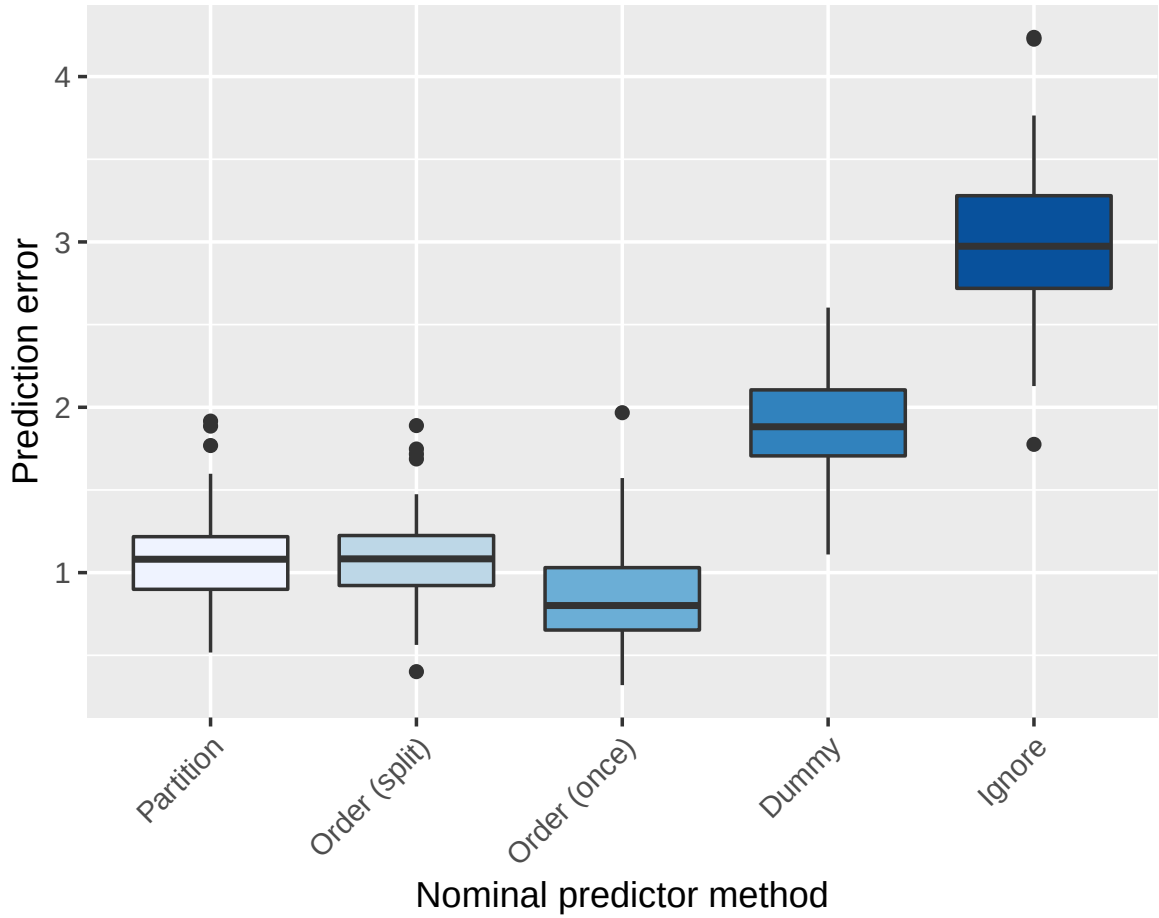
Claim 2: "no need to scale or recode predictors"

Standard approach works well only for few categories

- 4 numbers $\Rightarrow 2^{4-1} - 1 = 7$ partitions
- 28 EU countries $\Rightarrow 2^{28-1} - 1 = 1.34 \times 10^8$ partitions

Common Claims

Claim 2: “no need to scale or recode predictors”



Wright & König 2019 PeerJ 7:e6339

Common Claims

- ✓ Works well without tuning
  - ✗ Exception: High dimensional data, low signal-to-noise ratio
- ✓ No need to scale or recode predictors

Common Claims

Claim 3: “works well on high dimensional data”

Common Claims

Claim 3: "works well on high dimensional data"

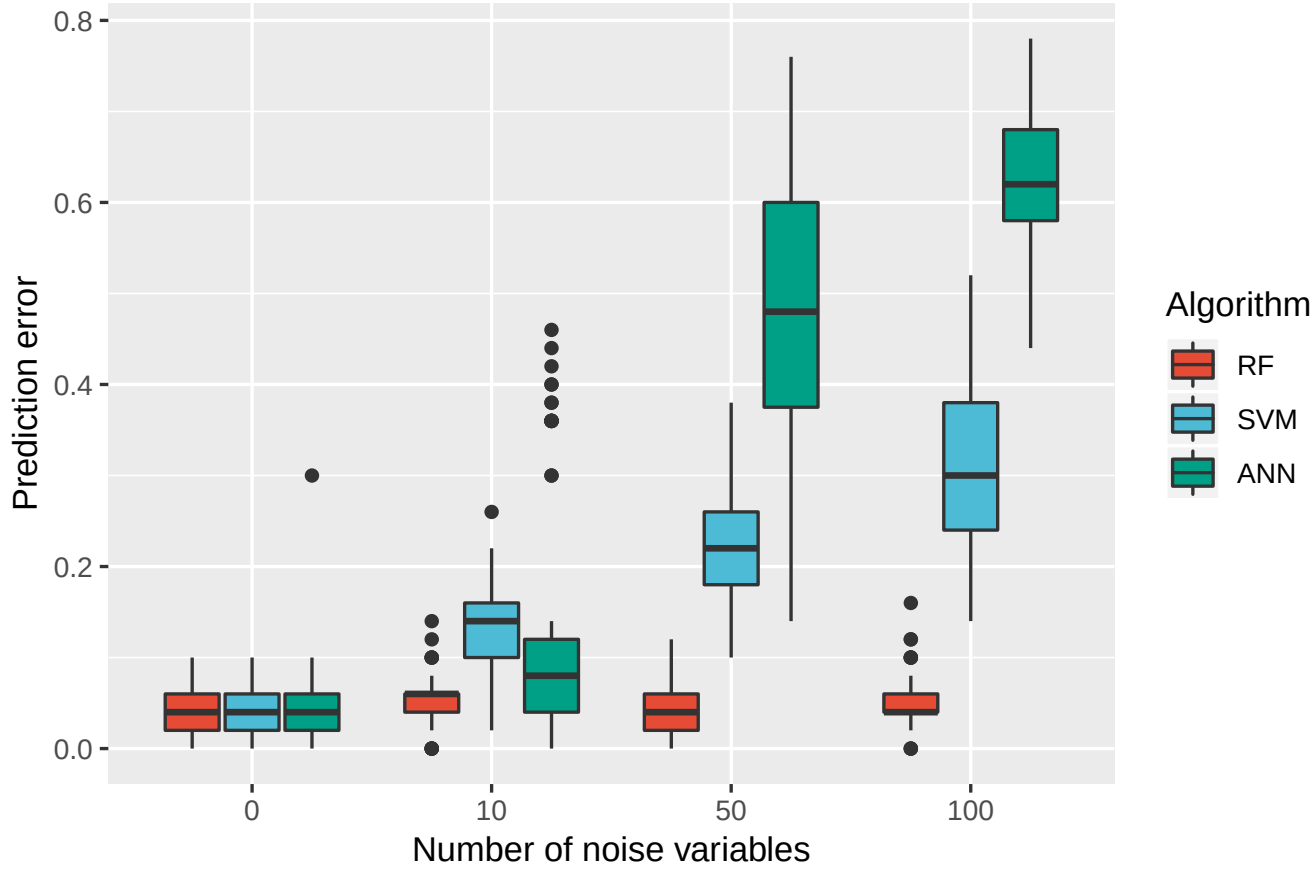
"[...] ability to deal with small sample sizes and high-dimensional feature spaces."

"the rate of **convergence depends only on** the number $|S|$ of **strong variables**, not on the dimension p."

— Biau & Scornet 2016

Common Claims

Claim 3: "works well on high dimensional data"



Common Claims

Claim 3: "works well on high dimensional data"

Intrinsic variable selection

- Greedy splitting algorithm selects best splitting variable
- No fitting of parameters or weights for other variables

Fast computation

See second part

Common Claims

- ✓ Works well without tuning
  - ✗ Exception: High dimensional data, low signal-to-noise ratio
- ✓ No need to scale or recode predictors
- ✓ Works well on high dimensional data

Common Claims

Claim 4: “cannot overfit”

Common Claims

Claim 4: "cannot overfit"

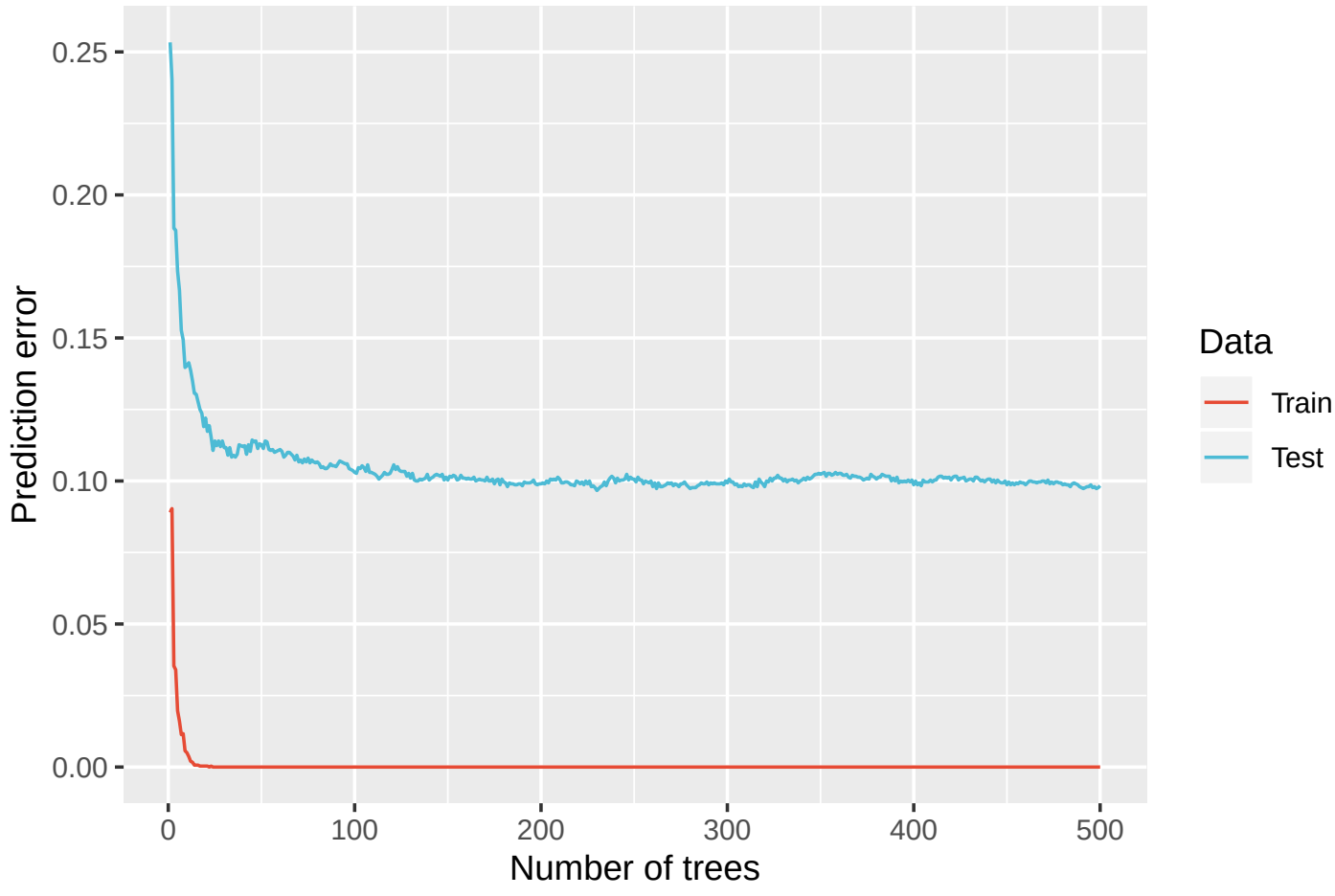
"[...] overfitting is not a problem."

"Random forests do not overfit as more trees are added."

— Breiman 2001

Common Claims

Claim 4: “cannot overfit”



Common Claims

Claim 4: "cannot overfit"

Definition 1

Adding trees does not hurt generalization error ✓

Definition 2

Training error is not smaller than generalization error ✗

Common Claims

- ✓ Works well without tuning
  - ✗ Exception: High dimensional data, low signal-to-noise ratio
- ✓ No need to scale or recode predictors
- ✓ Works well on high dimensional data
- ✓✗ Cannot overfit

Common Claims

Claim 5: “works for any kind of data”

Common Claims

Claim 5: "works for any kind of data"

Extensions

- Random survival forests: Time-to-event outcomes
- Conditional inference forests: Avoid split variable selection bias
- Quantile regression forests: Quantile regression
- Transformation forests: Predict distributions
- Block forests: Multi-omics data
- Random forests for bounded outcomes
- Generalized random forests

And many more ...

Ishwaran et al. 2008 Ann Appl Stat 2:841 • Hothorn et al. 2006 J Comp Graph Stat 15:651 • Meinshausen 2006 J Mach Learn Res 7:983 • Hothorn & Zeileis 2021 J Comp Graph Stat doi:10.1080/10618600.2021.1872581 • Hornung & Wright 2019 BMC Bioinformatics 20:358 • Weinhold et al. 2020 J Comp Graph Stat 29:639 • Athey et al. 2019 Ann Stat 47:1148

Common Claims

Claim 5: "works for any kind of data"

"Methods like random forests regularly outperform neural networks in arbitrary domains, especially when the underlying data sizes are small and **no domain-specific insight has been used to arrange the architecture** of the underlying neural network."

— Wang et al. 2018

Image, speech and natural language processing data

Top benchmark results are all deep learning, mostly CNN's, RNN's and (recently) transformers

Common Claims

- ✓ Works well without tuning
  - ✗ Exception: High dimensional data, low signal-to-noise ratio
- ✓ No need to scale or recode predictors
- ✓ Works well on high dimensional data
- ✓✗ Cannot overfit
- ✓ Works for almost any kind of data
  - ✗ Exception: Image, speech and natural language processing

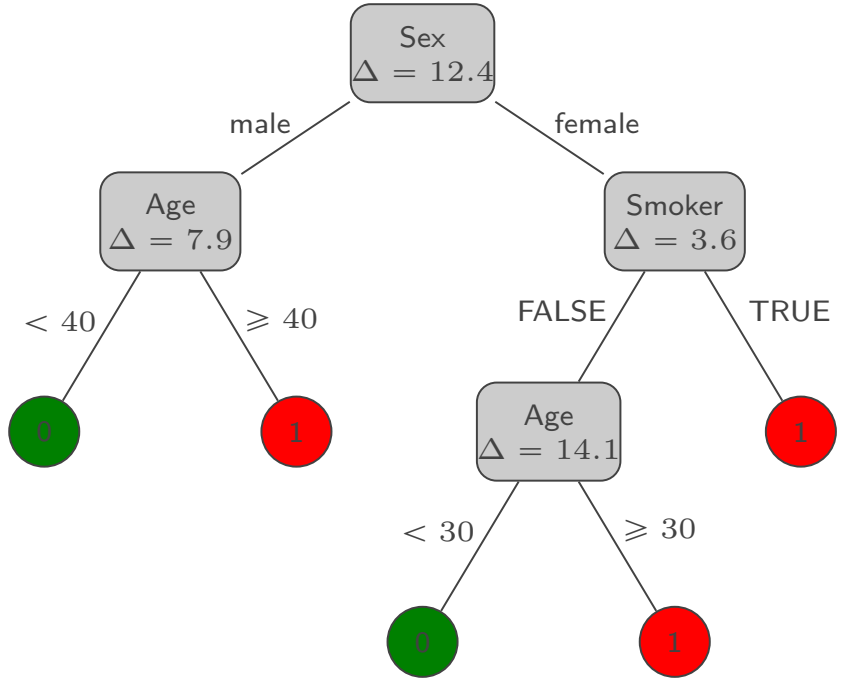
Common Claims

Claim 6: “is an interpretable model”

Common Claims

Claim 6: "is an interpretable model"

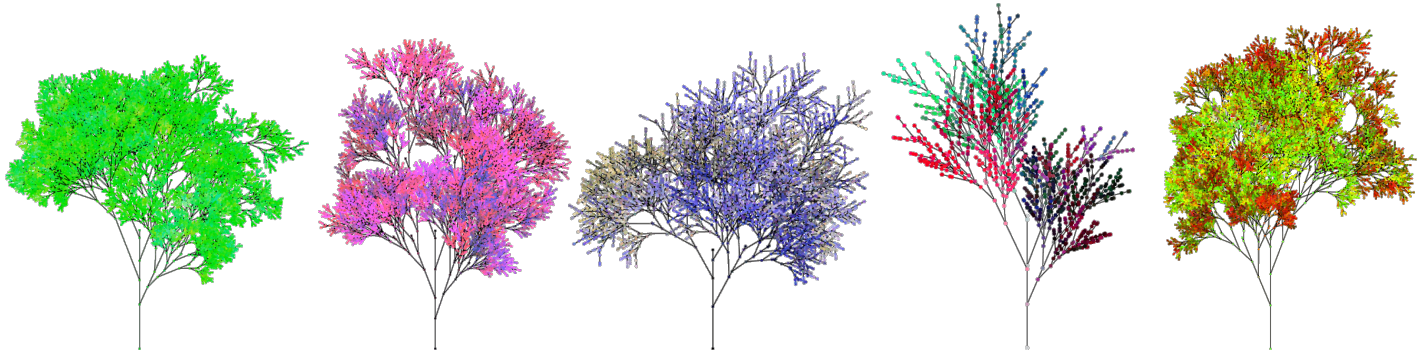
A single tree is interpretable



Common Claims

Claim 6: “is an interpretable model”

A random forest is **not** interpretable



Common Claims

Claim 6: "is an interpretable model"

Several variable importance measures available

- Gini/impurity importance: Sum of impurity measures
- Permutation importance: Permute variable, difference of prediction error
- Bias-corrected impurity importance: Difference of impurity importance to permuted version of variable
- Conditional variable importance: Conditional on other predictor variables
- Maximal subtree: Depth of first split on variable

Breiman et al. 1984 ISBN 9780412048418 • Breiman 2001 Mach Learn 45:5 • Nembrini et al. 2018 Bioinformatics 34:3711 • Strobl et al. 2008 BMC Bioinformatics 9:307 • Ishwaran et al. 2010 J Am Stat Assoc 105:205

Common Claims

- ✓ Works well without tuning
  - ✗ Exception: High dimensional data, low signal-to-noise ratio
- ✓ No need to scale or recode predictors
- ✓ Works well on high dimensional data
- ✓✗ Cannot overfit
- ✓ Works for almost any kind of data
  - ✗ Exception: Image, speech and natural language processing
- ✗ Is an interpretable model
  - ✓ Many variable importance measures available

Common Claims

Claim 7: “the statistical properties are well understood”

Common Claims

Claim 7: "the statistical properties are well understood"

Consistency

- Single tree not consistent
- RF consistent if $a_n/n \to 0$ and $a_n \to \infty$ (a_n/n: Subsampling rate)

Convergence rate

- Single trees slower than minimax rate
- RF achieves minimax rate. If more than 54% of variables have no effect, convergence rate faster than minimax

Asymptotic normality

- Single tree predictions asymptotically normally distributed
- RF predictions asymptotically normally distributed for subsampling

Biau et al. 2008 J Mach Learn Res 9:2015-33 • Scornet 2016 J Multiv Anal 146:72-83 • Wager et al. 2014 J Mach Learn Res 15:1625-51 • Wager & Athey 2018 J Am Stat Assoc 113:1228

Common Claims

Claim 7: "the statistical properties are well understood"

Assumptions

- Subsampling, not standard bootstrap
- Limit on subsampling rate, e.g. $a_n/n \to 0$ and $a_n \to \infty$
- Random splitting, e.g. purely random forest, selecting variable and split completely randomly

Biau et al. 2008 J Mach Learn Res 9:2015-33 • Scornet 2016 J Multiv Anal 146:72-83
Wager et al. 2014 J Mach Learn Res 15:1625-51 • Wager & Athey 2018 J Am Stat Assoc 113:1228

Common Claims

- ✓ Works well without tuning
  - ✗ Exception: High dimensional data, low signal-to-noise ratio
- ✓ No need to scale or recode predictors
- ✓ Works well on high dimensional data
- ✓✗ Cannot overfit
- ✓ Works for almost any kind of data
  - ✗ Exception: Image, speech and natural language processing
- ✗ Is an interpretable model
  - ✓ Many variable importance measures available
- ✓ The statistical properties are well understood
  - ✗ Assumptions do not hold with default/realistic settings

Common Claims

Claim 8: “the split variable selection is biased”

Common Claims

Claim 8: "the split variable selection is biased"

More possible split points for variables with more categories

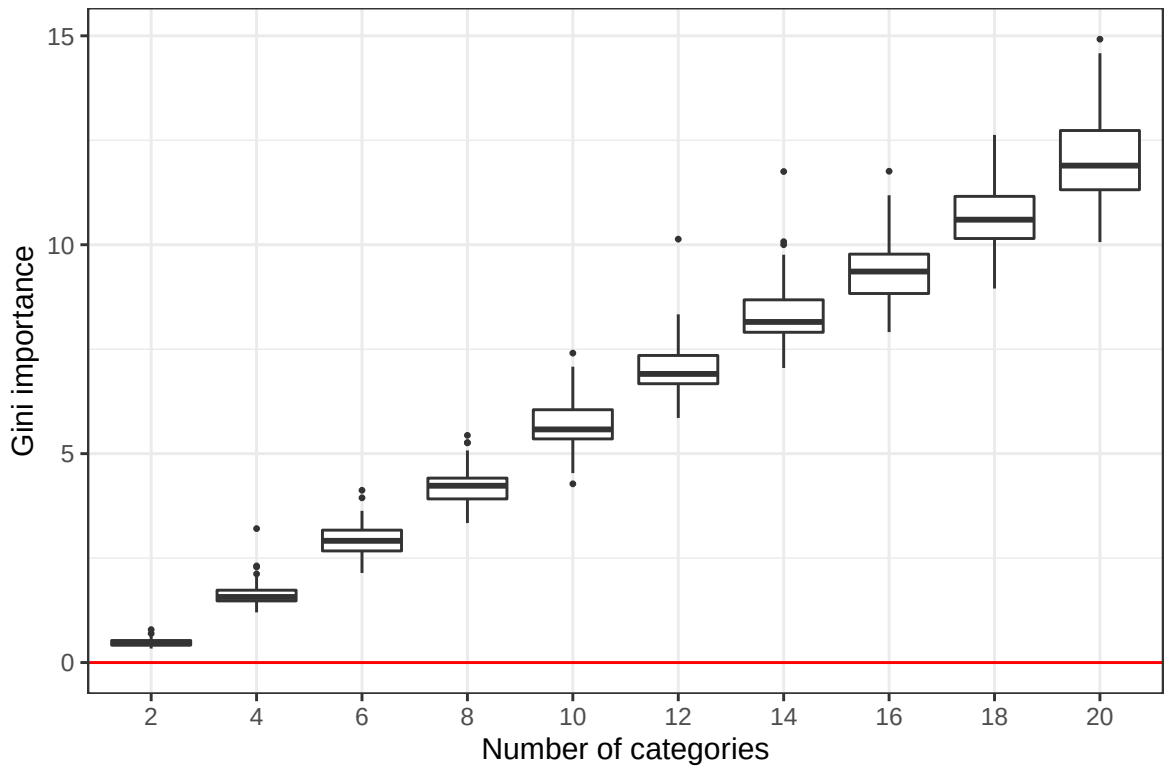
Example

- Sex: 2 unique values
- Medication type: 5 unique values
- Age (in years): m unique values
- Biomarker: n unique values

Strobl et al. 2007 BMC Bioinformatics 8:25 • Wright et al. 2017 Stat Med 36:1272 • Nembrini et al. 2018 Bioinformatics 34:3711

Common Claims

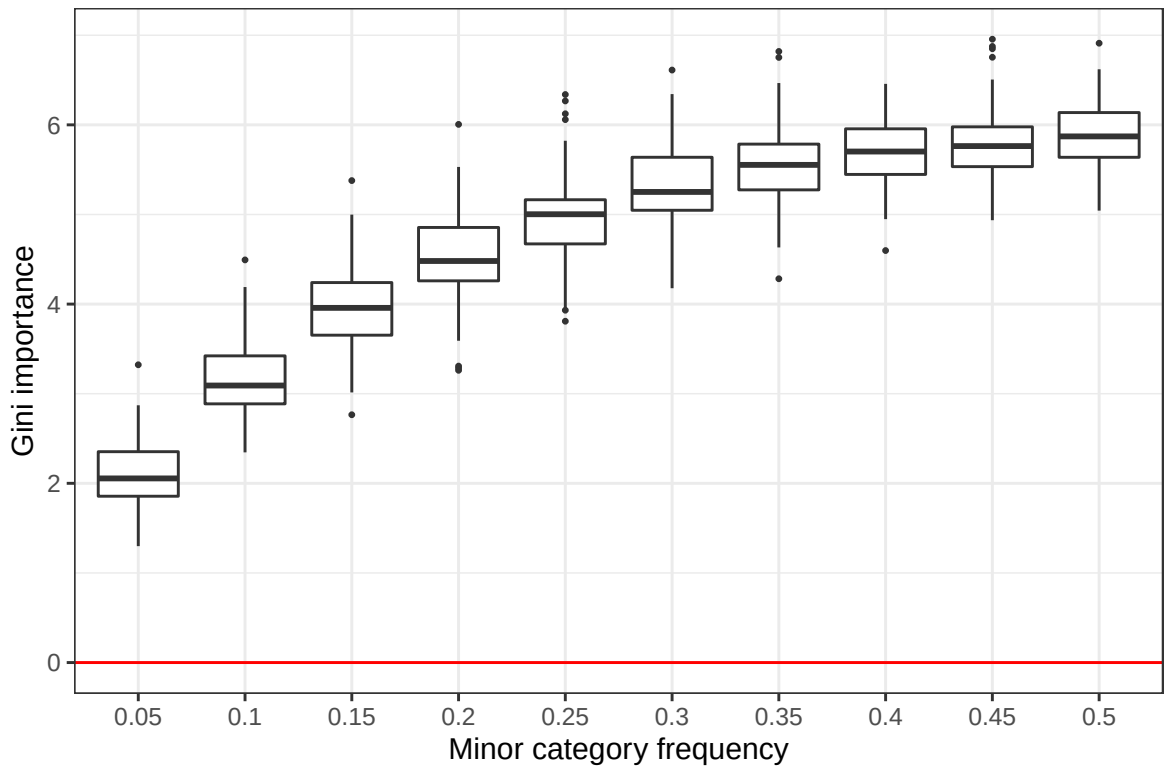
Claim 8: "the split variable selection is biased"



Strobl et al. 2007 BMC Bioinformatics 8:25 • Wright et al. 2017 Stat Med 36:1272 • Nembrini et al. 2018 Bioinformatics 34:3711

Common Claims

Claim 8: "the split variable selection is biased"



Boulesteix et al. 2012 Brief Bioinform 13:292 • Wright et al. 2017 Stat Med 36:1272 • Nembrini et al. 2018 Bioinformatics 34:3711

Common Claims

Claim 8: "the split variable selection is biased"

Solution 1

Randomized splitting rule

Solution 2

Conditional inference forests or maximally selected rank statistics

Solution 3

Bias-corrected variable importance

Geurts et al. 2006 Mach Learn 63:3 • Hothorn et al. 2006 J Comp Graph Stat 15:651 • Wright et al. 2017 Stat Med 36:1272 • Nembrini et al. 2018 Bioinformatics 34:3711

Common Claims

- ✓ Works well without tuning
  - ✗ Exception: High dimensional data, low signal-to-noise ratio
- ✓ No need to scale or recode predictors
- ✓ Works well on high dimensional data
- ✓✗ Cannot overfit
- ✓ Works for almost any kind of data
  - ✗ Exception: Image, speech and natural language processing
- ✗ Is an interpretable model
  - ✓ Many variable importance measures available
- ✓ The statistical properties are well understood
  - ✗ Assumptions do not hold with default/realistic settings
- ✓ The split variable selection is biased $\rightarrow$ solved

Common Claims

Claim 9: “performance is not state of the art”

Common Claims

Claim 9: "performance is not state of the art"

Gijsbers et al. 2019: Comparison of automated machine learning algorithms on 39 datasets with 4h time budget

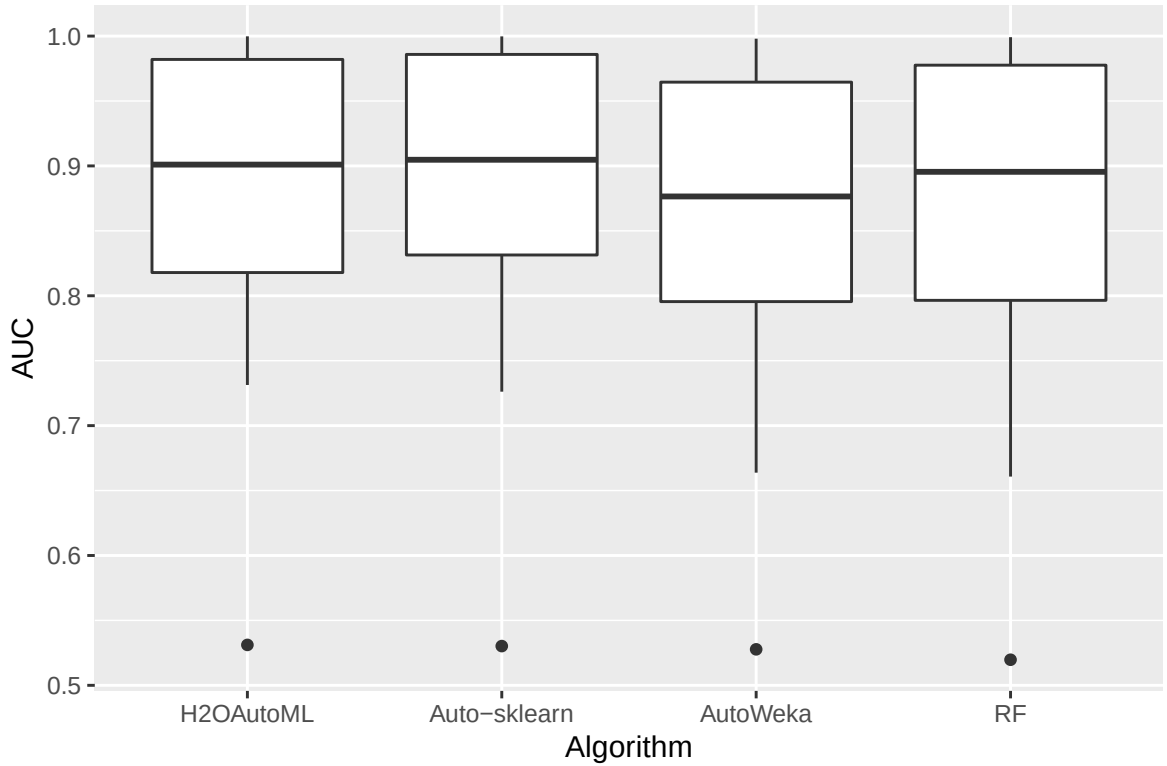
Results

Algorithm	Average rank
H2OAutoML	1.5
Auto-sklearn	2.1
AutoWeka	3.4
RF	3.0

Gijsbers et al. 2019 ICML arXiv:1907.00909

Common Claims

Claim 9: "performance is not state of the art"



Average runtime of random forest: 16.4 seconds

Gijsbers et al. 2019 ICML arXiv:1907.00909

Common Claims

- ✓ Works well without tuning
  - ✗ Exception: High dimensional data, low signal-to-noise ratio
- ✓ No need to scale or recode predictors
- ✓ Works well on high dimensional data
- ✓✗ Cannot overfit
- ✓ Works for almost any kind of data
  - ✗ Exception: Image, speech and natural language processing
- ✗ Is an interpretable model
  - ✓ Many variable importance measures available
- ✓ The statistical properties are well understood
  - ✗ Assumptions do not hold with default/realistic settings
- ✓ The split variable selection is biased → solved
- ✓✗ Performance is not state of the art

Common Claims

Claim 10: “detects interactions”

Common Claims

Claim 10: "detects interactions"

"Random forests are generally **capable of capturing** gene-gene interactions, but current variable importance measures are **unable to detect** them as interactions."

"interactions are **masked by marginal effects** and interactions cannot be differentiated from marginal effects."

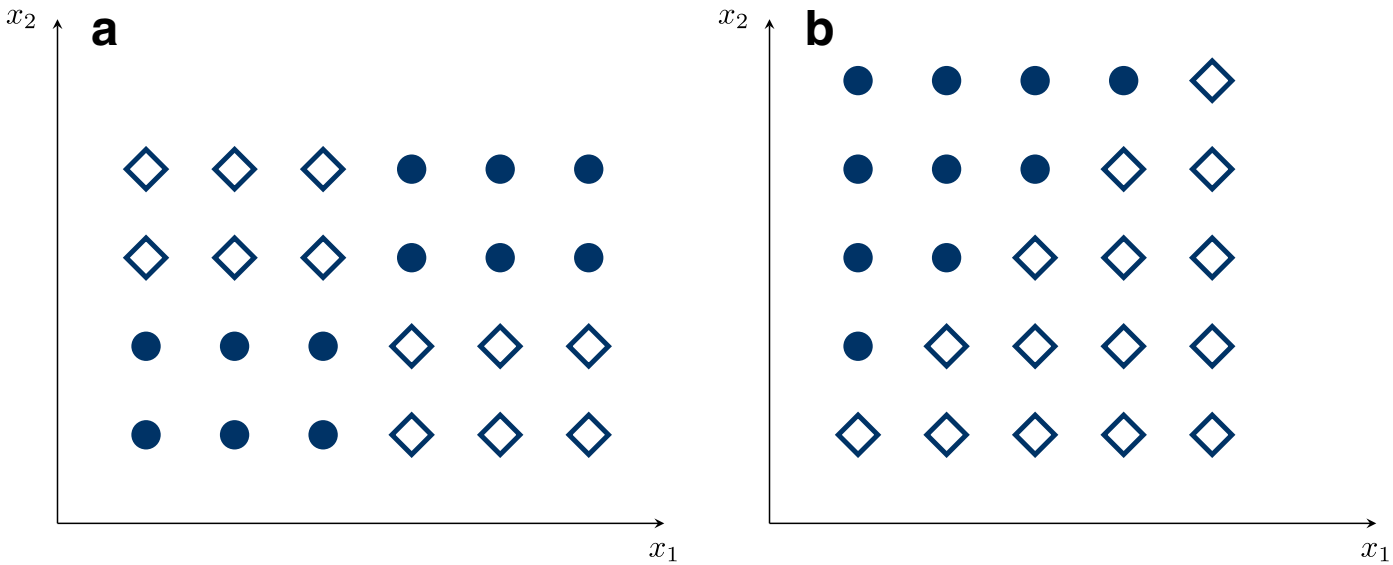
— Wright et al. 2016

"although it is able to **take interactions into account**, it **does not specifically detect them**."

— Schmalohr et al. 2018

Common Claims

Claim 10: "detects interactions"



Wright et al. 2016 BMC Bioinformatics 17:145

Common Claims

Claim 10: "detects interactions"

Curse of dimensionality worse for interactions

Low probability to subsequently select all interacting variables in high dimensional data.

Example with $p = 100\,000$:

$\texttt{mtry} = \sqrt{p} = 316$: $p_{\text{2way}} = 0.00001$, $p_{\text{3way}} = 3 \times 10^{-8}$

$\texttt{mtry} = p/2 = 50\,000$: $p_{\text{2way}} = 0.25$, $p_{\text{3way}} = 0.125$

Need marginal effect for first split

RF splitting only detects marginal effects

Common Claims

- ✓ Works well without tuning
  - ✗ Exception: High dimensional data, low signal-to-noise ratio
- ✓ No need to scale or recode predictors
- ✓ Works well on high dimensional data
- ✓✗ Cannot overfit
- ✓ Works for almost any kind of data
  - ✗ Exception: Image, speech and natural language processing
- ✗ Is an interpretable model
  - ✓ Many variable importance measures available
- ✓ The statistical properties are well understood
  - ✗ Assumptions do not hold with default/realistic settings
- ✓ The split variable selection is biased → solved
- ✓✗ Performance is not state of the art
- ✓✗ Detects interactions

Outline

1. Introduction
2. Common Claims
3. Implementations in R
4. Discussion & Conclusion

RF Implementations in R

Original RF

- `randomForest`
- `randomForestSRC`
- `Rborist`
- `ranger`

Not available anymore

- `bigrf`
- `Random Jungle`

Liaw & Wiener 2002 R News 2:18 • Ishwaran et al. 2008 Ann Appl Stat 2:841 • Seligman 2019 https://CRAN.R-project.org/package=Rborist • Wright & Ziegler 2017 J Stat Softw 77:1

RF Implementations in R

Extensions

- `party`: Conditional inference forests
- `partykit`: Conditional inference forests, model-based recursive partitioning
- `quantregForest`: Quantile regression forests
- `trtf`: Transformation forests
- `blockForest`: Block forests
- `grf`: Generalized random forests

Hothorn et al. 2006 J Comp Graph Stat 15:651 • Hothorn & Zeileis 2015 J Mach Learn Res 16:3905 • Meinshausen 2006 J Mach Learn Res 7:983 • Hothorn & Zeileis 2021 J Comp Graph Stat doi:10.1080/10618600.2021.1872581 • Hornung & Wright 2019 BMC Bioinformatics 20:358 • Athey et al. 2019 Ann Stat 47:1148

Runtime and memory usage

Low dimensional data

- 100,000 samples, 100 variables
- 1000 trees, `mtry`=10
- 12 CPU cores (except `randomForest`)

Package	Runtime (minutes) binary vars.	cont. vars.	Memory usage (GB)
`randomForest`	31.53	42.65	9.37
`randomForest` (MC)	5.34	7.20	13.20
`randomForestSRC`	1.72	5.96	7.26
`Rborist`	5.42	4.93	2.74
`ranger`	0.74	4.85	1.27

Slower machine than in original paper

Runtime and memory usage

Low dimensional data

- 100,000 samples, 100 variables
- 1000 trees, `mtry`=10
- 12 CPU cores (except `randomForest`)

Package	Runtime (minutes) binary vars.	cont. vars.	Memory usage (GB)
`randomForest`	31.53	42.65	9.37
`randomForest` (MC)	5.34	7.20	13.20
`randomForestSRC`	1.72	5.96	7.26
`Rborist`	5.42	4.93	2.74
`ranger`	0.74	4.85	1.27

Slower machine than in original paper

Runtime and memory usage

High dimensional data (genetic data)

- 10,000 samples, 150,000 variables (SNPs)
- 1000 trees, `mtry`=5000
- 12 CPU cores (except `randomForest`)

Package	Runtime (hours)	Memory usage (GB)
`randomForest`	93.04	52.73
`randomForest` (MC)	NA	>96
`randomForestSRC`	1.33	36.05
`Rborist`	NA	>96
`ranger`	0.68	17.71
`ranger` (GWAS mode)	0.30	0.13

Slower machine than in original paper

NA: Memory error

Outline

1. Introduction
2. Common Claims
3. Implementations in R
4. Discussion & Conclusion

Discussion

Pros

- Little or no tuning and data recoding required
- Good performance on almost any kind of data
- Overfitting not a major problem
- Variable importance measures available

Cons

- Bad performance on images, speech and natural language processing
- Not per se interpretable
- Will not win prediction challenges

Discussion

Fast implementations available

- `Rborist` fastest for continuous features and large sample sizes
- `ranger` fastest in all other cases
- Efficient analysis of genome-wide data with `ranger`

Caution

- Some packages differ in results
- Performance depends on type and size of data

Discussion

Fast implementations available

- `Rborist` fastest for continuous features and ~~large~~ huge sample sizes
- `ranger` fastest in all other cases
- Efficient analysis of genome-wide data with `ranger`

Caution

- Some packages differ in results
- Performance depends on type and size of data