Explainability via tree-based methods

Erwan Scornet
Lecturer at Sorbonne University

October 2025

Outline

1. Explainability and random forests

2. Decision rules

3. Variable importance
   - A first variable importance in random forests: MDI
   - A second variable importance in random forests: MDA
   - Shapley values via random forests

Summary

1. Explainability and random forests

2. Decision rules

3. Variable importance
   - A first variable importance in random forests: MDI
   - A second variable importance in random forests: MDA
   - Shapley values via random forests

Why do we need interpretability?

Machine learning is used for **decision support**.

Predicting is not enough

Understanding predictions is vital

- for Machine learning to be accepted (sensible applications in health, justice, defense)
- To improve algorithms (e.g., detect unfairness and try to correct it)

Keywords: trust, transparency, accountability, fairness, ethics.

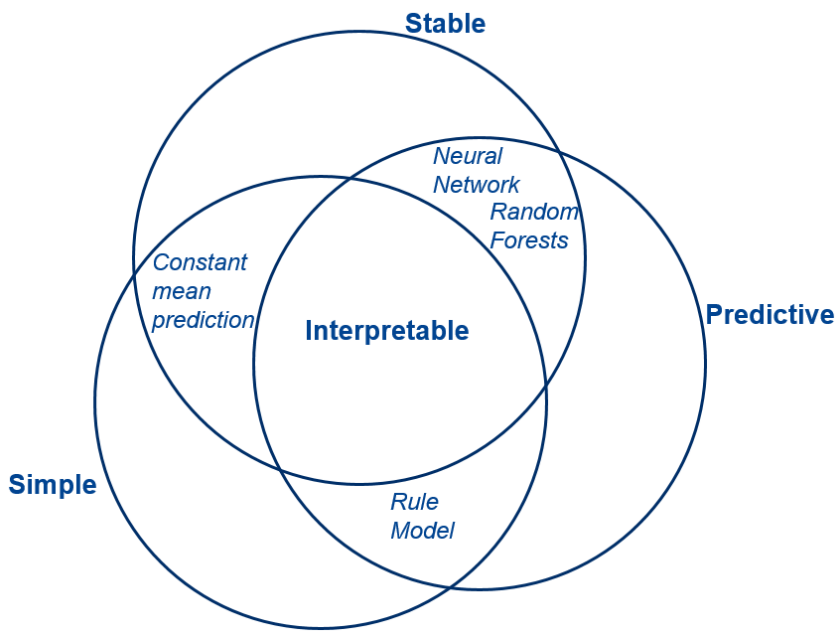
NIPS2017 debate: Interpretability is necessary for Machine learning

`https://www.youtube.com/watch?v=93Xv8vJ2acI`

Interpretable Models

- No agreement about a rigorous definition of interpretability [Lipton, 2016, Doshi-Velez and Kim, 2017, Murdoch et al., 2019]

- Minimum requirements for interpretability
  1. Simplicity [Murdoch et al., 2019]
  2. Stability [Yu, 2013]
  3. Predictivity [Breiman, 2001c]



Random forests are great!

Random forests are a class of algorithms created by Breiman [2001b] to solve regression and classification problems

- Among state-of-the-art methods for tabular data
- No need to precisely tune parameters
- Valuable in high-dimension settings
- Based on trees which are interpretable

Random forests are great!

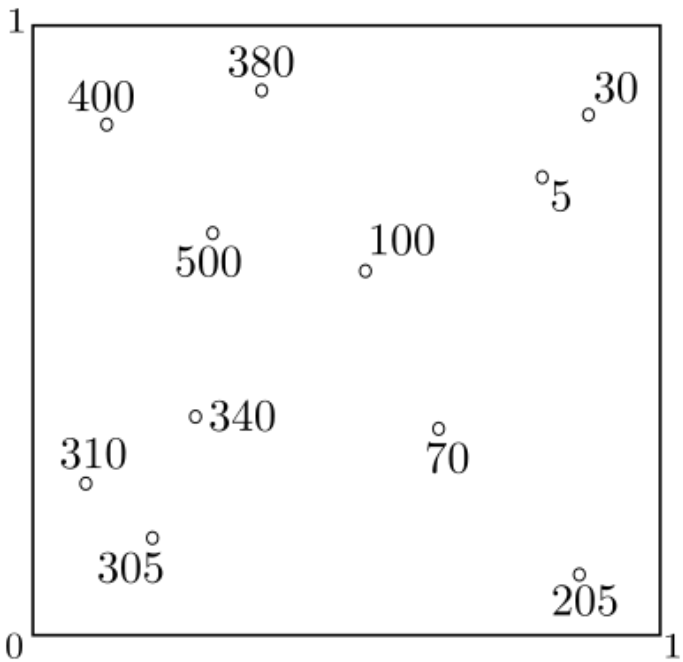
Random forests are a class of algorithms created by Breiman [2001b] to solve regression and classification problems

- Among state-of-the-art methods for tabular data
- No need to precisely tune parameters
- Valuable in high-dimension settings
- Based on trees which are interpretable

- Difficult to analyze theoretically
- Difficult to interpret

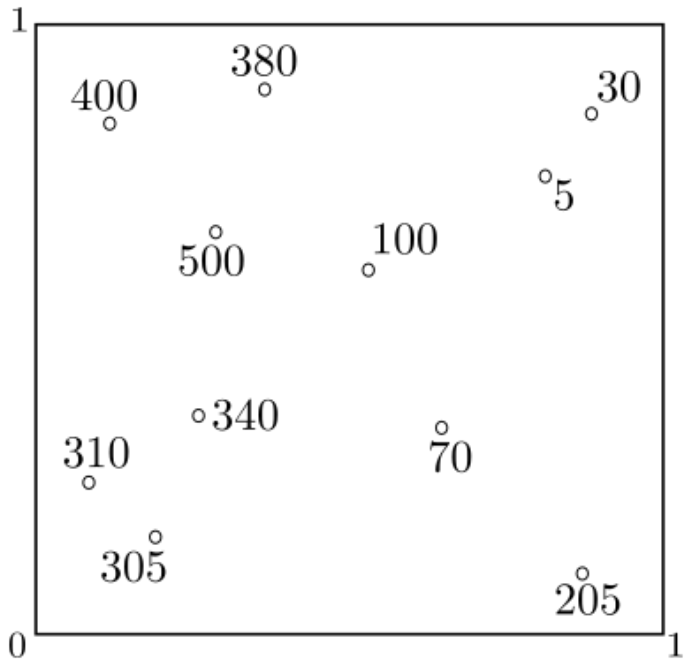
How to build a tree?

- Trees are built recursively by splitting the current cell into two children until some stopping criterion is satisfied.



How to build a tree?

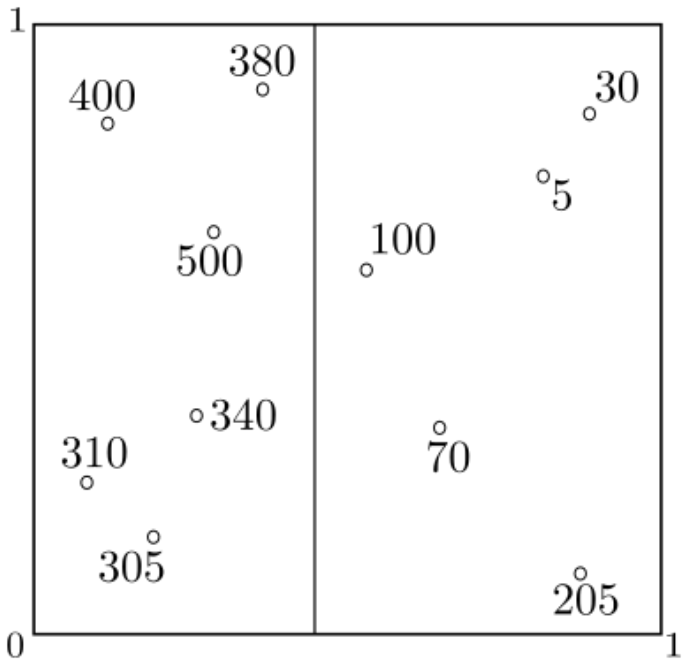
- Trees are built recursively by splitting the current cell into two children until some stopping criterion is satisfied.

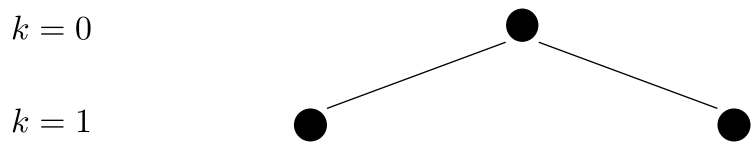


$k = 0$

How to build a tree?

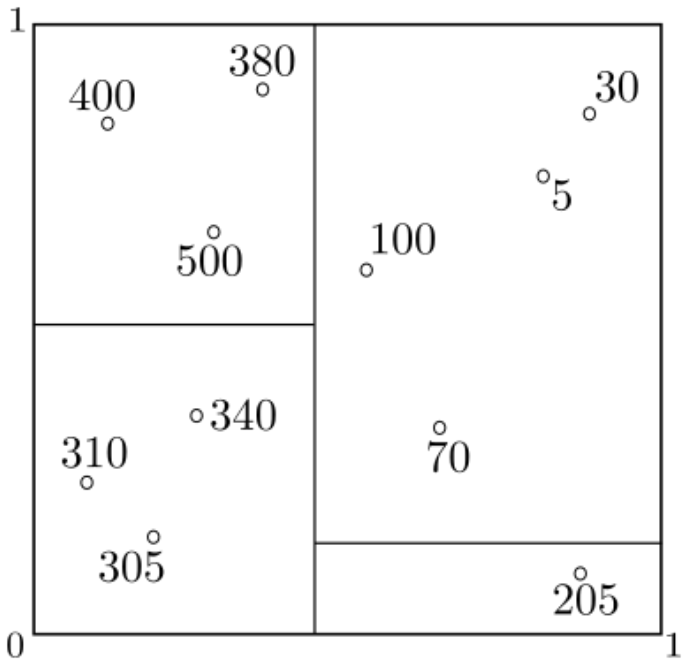
- Trees are built recursively by splitting the current cell into two children until some stopping criterion is satisfied.

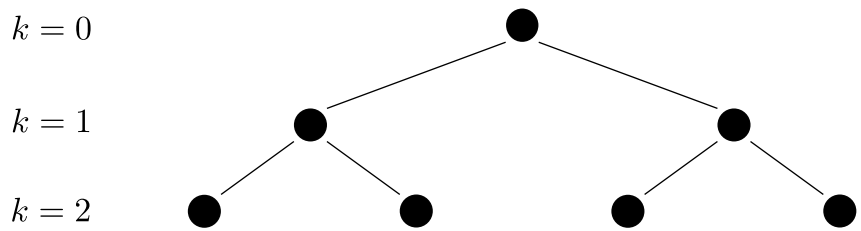




How to build a tree?

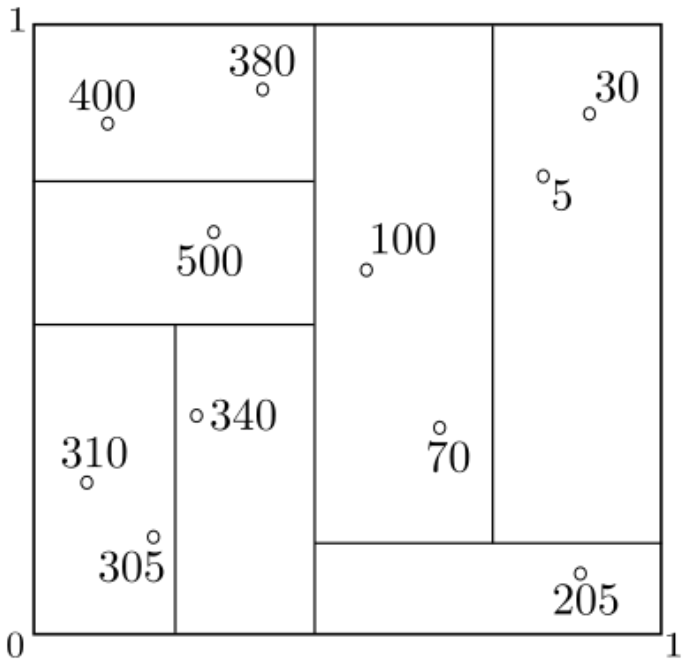
- Trees are built recursively by splitting the current cell into two children until some stopping criterion is satisfied.

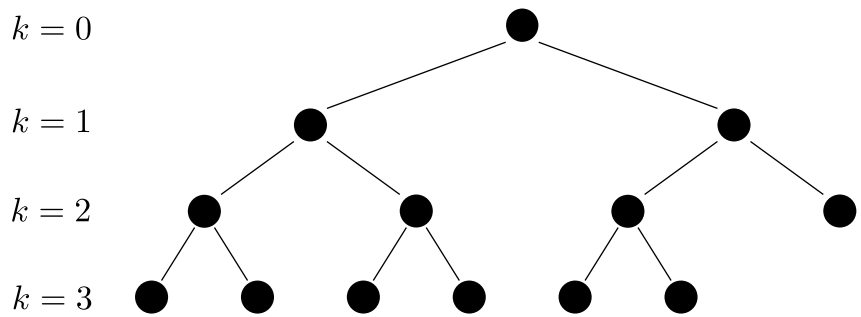




How to build a tree?

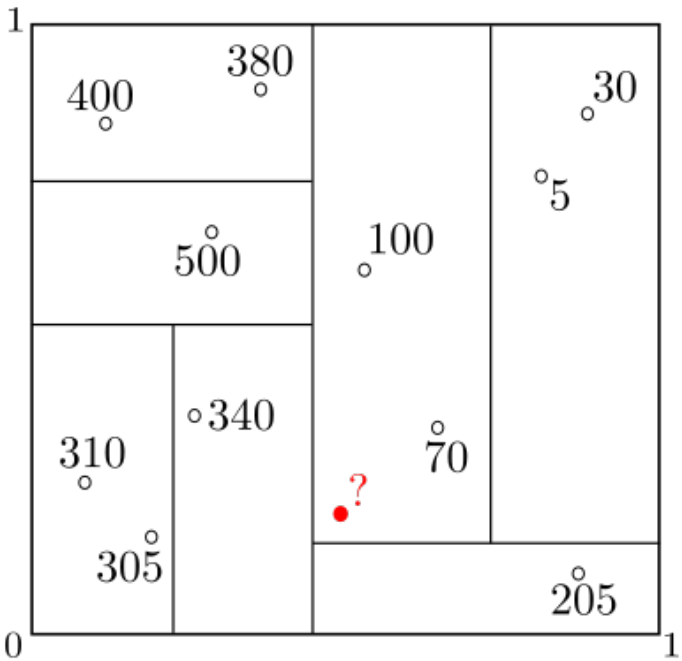
- Trees are built recursively by splitting the current cell into two children until some stopping criterion is satisfied.

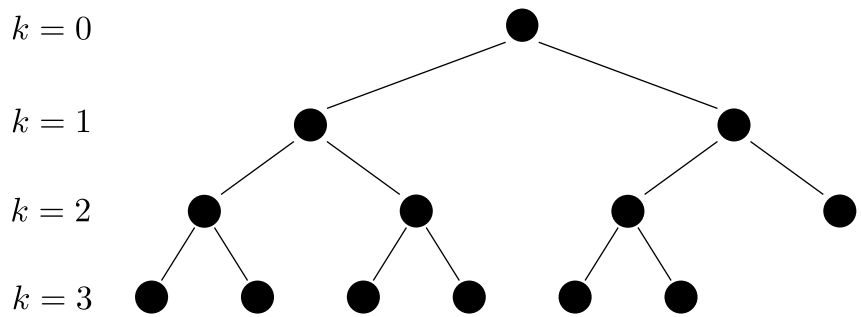




How to build a tree?

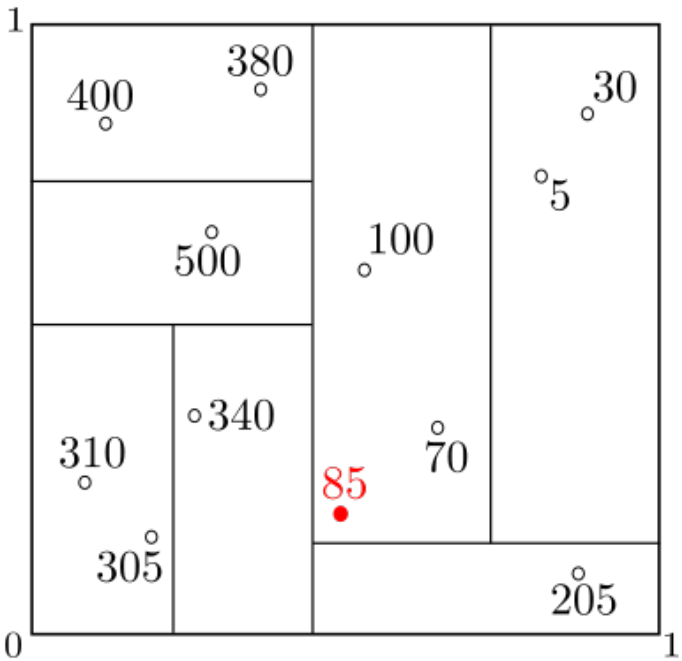
- Trees are built recursively by splitting the current cell into two children until some stopping criterion is satisfied.

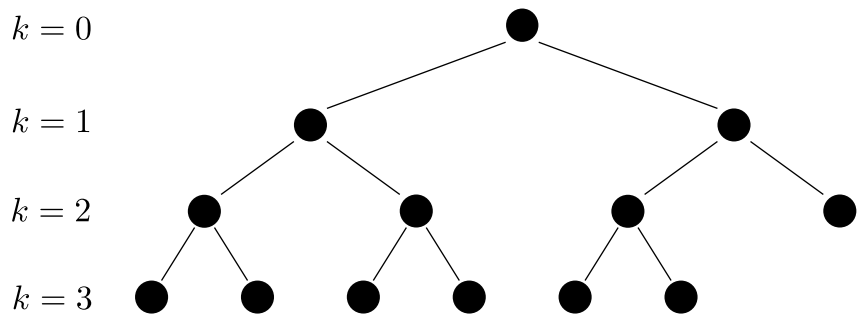




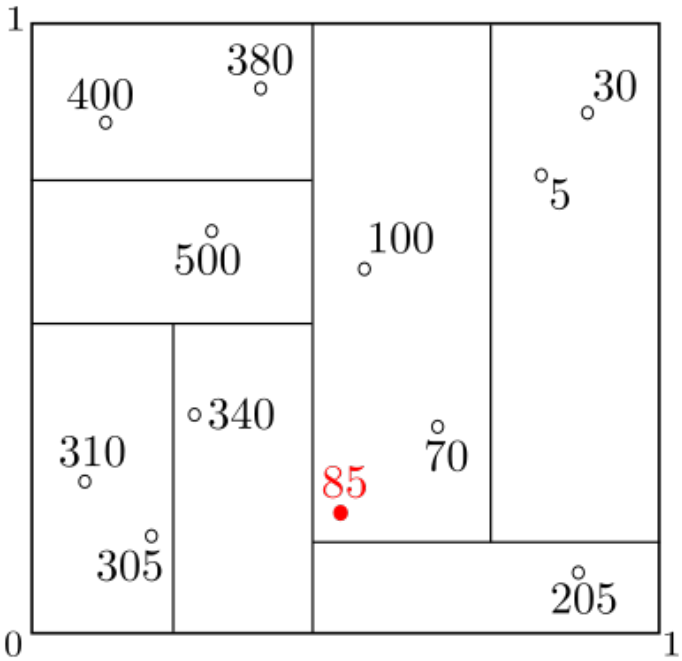
How to build a tree?

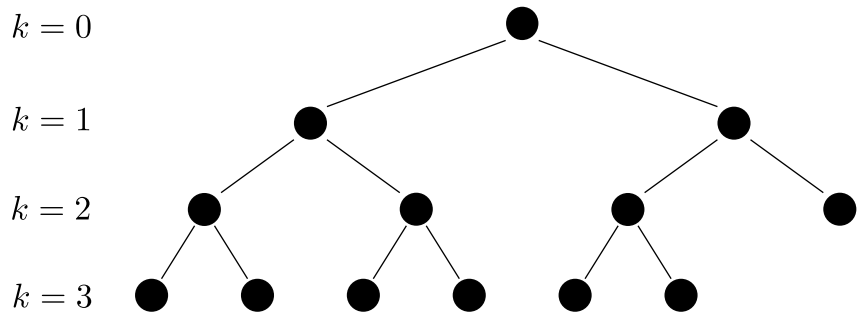
- Trees are built recursively by splitting the current cell into two children until some stopping criterion is satisfied.





How to build a tree?





Breiman Random forests are defined by

1. A splitting rule : minimize the variance within the resulting cells.
2. A stopping rule : stop when each cell contains less than `nodesize` $= 2$ observations.

How to perform splits?

For a split direction $j \in \{1, \ldots, d\}$ and a split position $z \in [0, 1]$, the criterion takes the form

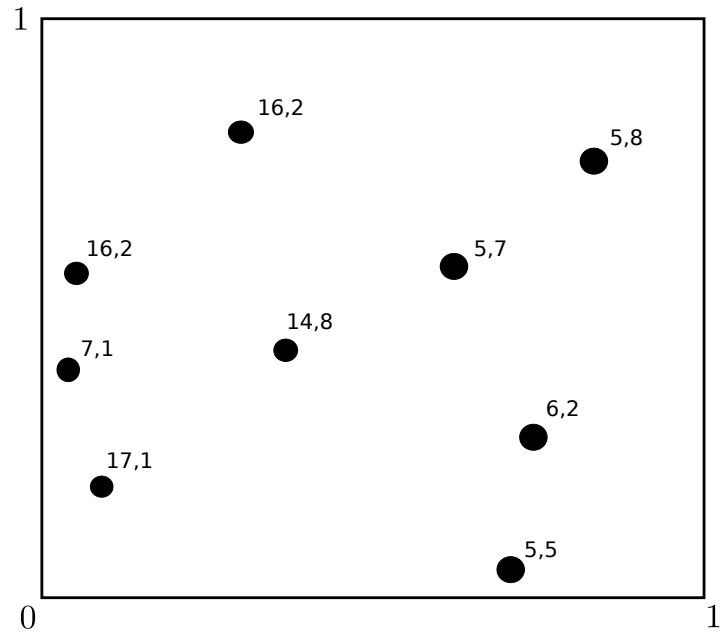
$$L_n(j, z) = \frac{1}{N_n(A)} \sum_{i=1}^{n} \left(Y_i - \bar{Y}_{A_L} \mathbb{1}_{\mathbf{x}_i^{(j)} < z} - \bar{Y}_{A_R} \mathbb{1}_{\mathbf{x}_i^{(j)} \geqslant z} \right)^2,$$

where

- $A_L = \{\mathbf{x} \in A : \mathbf{x}^{(j)} < z\}$ and $A_R = \{\mathbf{x} \in A : \mathbf{x}^{(j)} \geqslant z\}$
- $\bar{Y}_A$ is the average of the Y_i's belonging to A.
- $N_n(A)$ is the number of points in A

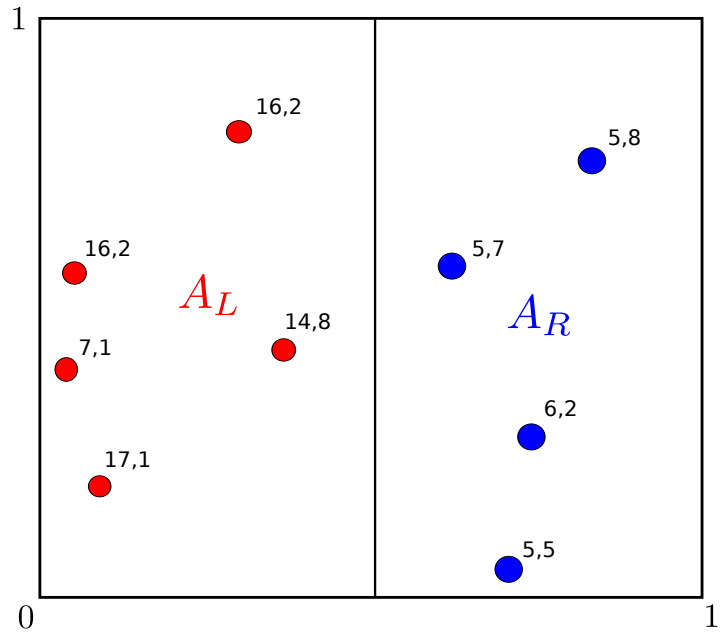
How to perform splits of Breiman's forests?

An example: $j = 1$ and $z = 0.5$.



How to perform splits of Breiman's forests?

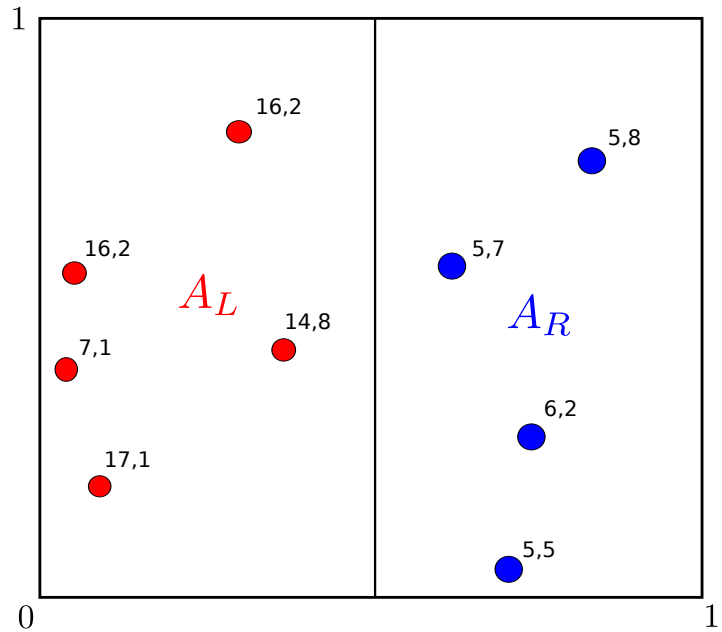
An example: $j = 1$ and $z = 0.5$.



$$L_n(1, 0.5) = \frac{1}{N_n(A)} \sum_{i=1}^{n} \left(Y_i - \underbrace{\bar{Y}_{A_L} \mathbb{1}_{\mathbf{X}_i^{(1)} < 0.5}}_{\text{Average on } A_L} - \bar{Y}_{A_R} \mathbb{1}_{\mathbf{X}_i^{(1)} \geqslant 0.5} \right)^2,$$

How to perform splits of Breiman's forests?

An example: $j = 1$ and $z = 0.5$.

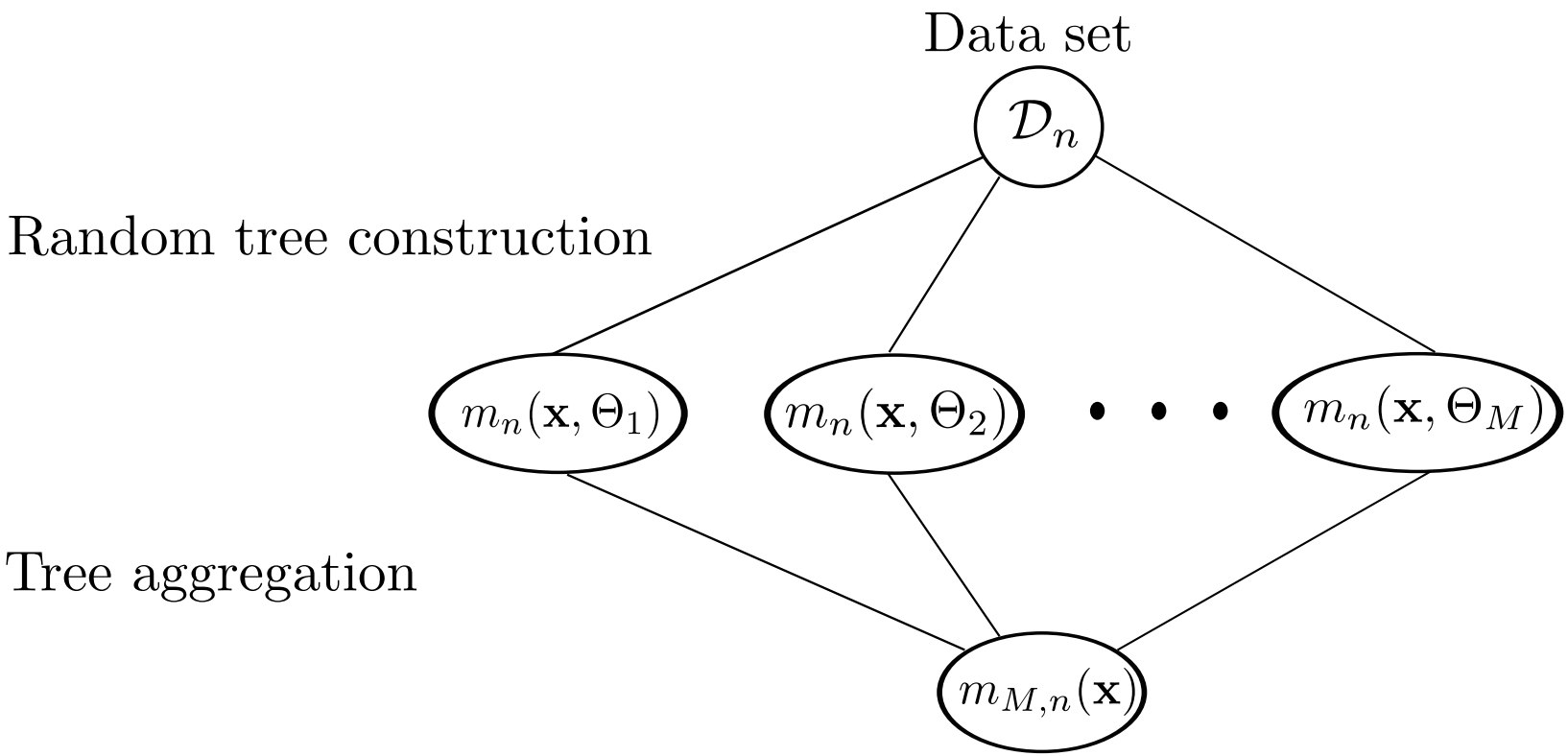


$$L_n(1, 0.5) = \frac{1}{N_n(A)} \sum_{i=1}^{n} \left(Y_i - \bar{Y}_{A_L} \mathbb{1}_{\mathbf{x}_i^{(1)} < 0.5} - \underbrace{\bar{Y}_{A_R} \mathbb{1}_{\mathbf{x}_i^{(1)} \geqslant 0.5}}_{\text{Average on } A_R} \right)^2,$$

Construction of random forests

Randomness in tree construction

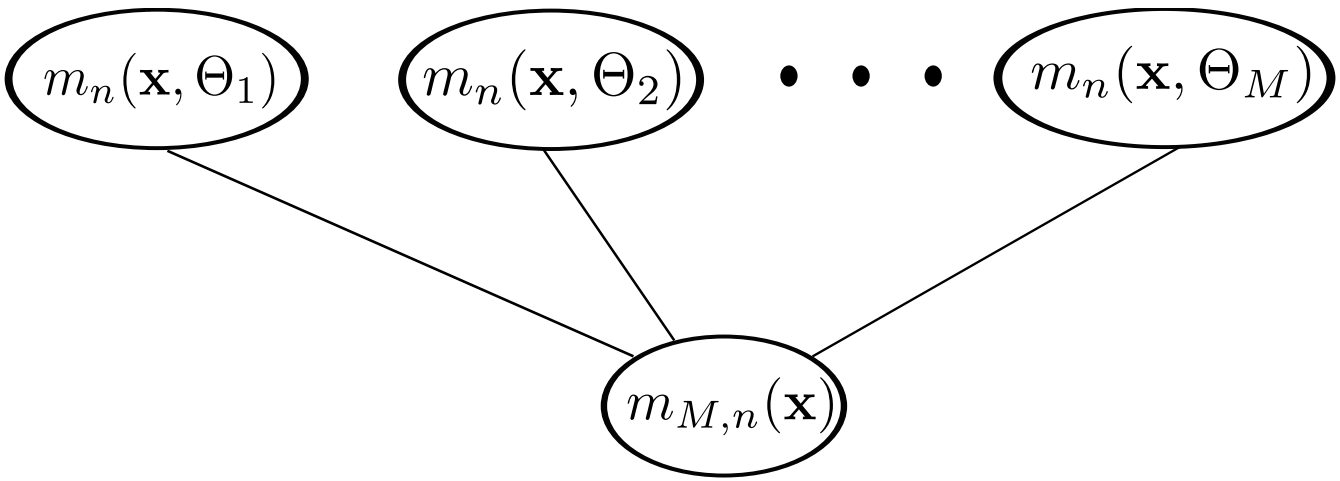
- Resampling the data set via bootstrap;
- For each cell:
  - Preselecting a subset of m_{try} variables, eligible for splitting.



Construction of Breiman forests

Breiman tree

- Select a_n observations with replacement among the original sample $\mathcal{D}_n$. Use only these observations to build the tree.
- For each cell,
  - Select randomly `mtry` coordinates among $\{1, \ldots, d\}$;
  - Choose the best split along previous direction, the one minimizing the CART criterion.
- Stop when each cell contains less than `nodesize` observations.



Theory of RF [Breiman, 2001b]: literature review

- **Simplified RF versions**, whose construction is **independent of the dataset**.

  [Biau et al., 2008, Biau, 2012, Genuer, 2012, Arlot and Genuer, 2014, Scornet, 2016, Mourtada et al., 2020, Klusowski, 2021]

Theory of RF [Breiman, 2001b]: literature review

- **Simplified RF versions**, whose construction is **independent of the dataset**.

  [Biau et al., 2008, Biau, 2012, Genuer, 2012, Arlot and Genuer, 2014, Scornet, 2016, Mourtada et al., 2020, Klusowski, 2021]

- Analysis of more data-dependent forests:
  - **Asymptotic normality** of random forests [Mentch and Hooker, 2016, Wager and Athey, 2018],
  - **Variable importance** [Louppe et al., 2013, Li et al., 2019, Scornet, 2022],
  - **(Rate of) consistency** [Scornet et al., 2015, Wager and Walther, 2015, Klusowski and Tian, 2024].

Theory of RF [Breiman, 2001b]: literature review

- **Simplified RF versions**, whose construction is **independent of the dataset**.

  [Biau et al., 2008, Biau, 2012, Genuer, 2012, Arlot and Genuer, 2014, Scornet, 2016, Mourtada et al., 2020, Klusowski, 2021]

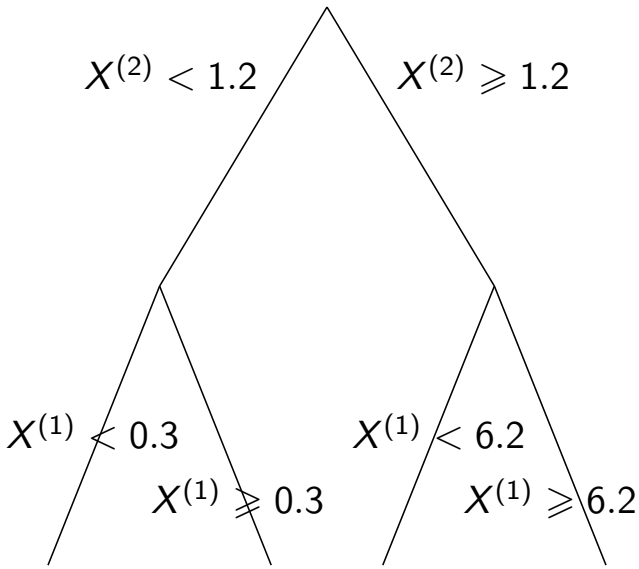
- Analysis of more data-dependent forests:
  - **Asymptotic normality** of random forests [Mentch and Hooker, 2016, Wager and Athey, 2018],
  - **Variable importance** [Louppe et al., 2013, Li et al., 2019, Scornet, 2022],
  - **(Rate of) consistency** [Scornet et al., 2015, Wager and Walther, 2015, Klusowski and Tian, 2024].

- Literature review on random forests:
  - **Methodological review** [Criminisi et al., 2011, Boulesteix et al., 2012],
  - **Theoretical review** [Biau and Scornet, 2016, Scornet and Hooker, 2025]

Existing Approaches to Interpretability/Explainability

- Interpretable models

  E.g. decision trees, decision rules

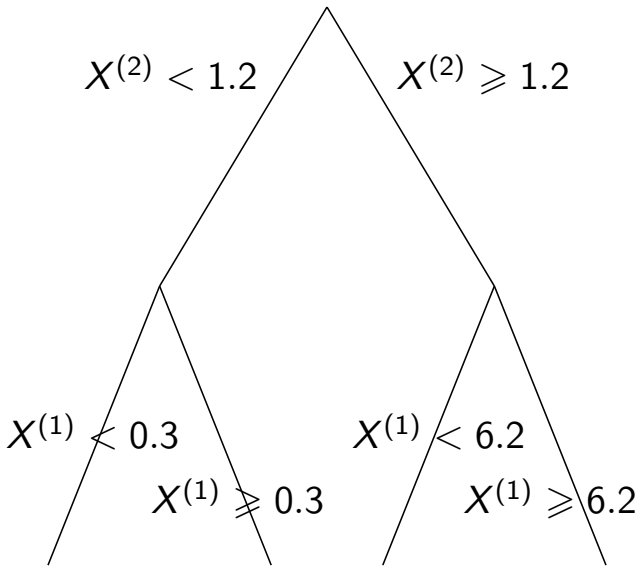


Unstable

Existing Approaches to Interpretability/Explainability

- Interpretable models

  E.g. decision trees, decision rules



Unstable

- Black-box models

E.g. Neural networks, Random forests

Combined with post-processing
E.g. variable importance
sensitivity analysis
local linearization

Hard to operationalize

Going beyong black-box nature of random forests

- Designing simple, interpretable and stable rules extracted from random forests: SIRUS
  - Interpretable Random Forests via Rule Extraction [Bénard et al., 2021a,b],
    by C. Bénard, G. Biau, S. Da Veiga, E. Scornet

Going beyong black-box nature of random forests

- Designing simple, interpretable and stable rules extracted from random forests: SIRUS
  - Interpretable Random Forests via Rule Extraction [Bénard et al., 2021a,b],
    by C. Bénard, G. Biau, S. Da Veiga, E. Scornet
- Variable importance in random forests:
  - Mean Decrease Impurity (MDI) [Breiman, 2002]

    Trees, forests, and impurity-based variable importance [Scornet, 2022]
  - Mean Decrease Accuracy (MDA) [Breiman, 2001b]

    MDA for random forests: inconsistency, and a practical solution via the Sobol-MDA [Bénard et al., 2022a]

    by C. Bénard, S. Da Veiga, E. Scornet
  - How to use random forests to estimate Shapley effects?

    SHAFF: Fast and consistent SHApley eFfect estimates via random Forests [Bénard et al., 2022b]

    by C. Bénard, G. Biau, S. Da Veiga, E. Scornet

Summary

1. Explainability and random forests

2. Decision rules

3. Variable importance
   - A first variable importance in random forests: MDI
   - A second variable importance in random forests: MDA
   - Shapley values via random forests

Decision rules

- An example: Titanic dataset (predict the survival of each passenger)

Average survival rate $p_s = 39\%$.

if	`sex` is male	**then**	$p_s = 19\%$	**else**	$p_s = 74\%$
if	1^{st} or 2^{nd} `class`	**then**	$p_s = 56\%$	**else**	$p_s = 24\%$
if	1^{st} or 2^{nd} `class` & `sex` is female	**then**	$p_s = 95\%$	**else**	$p_s = 25\%$
if	`fare` $< 10.5£$	**then**	$p_s = 20\%$	**else**	$p_s = 50\%$
if	no `parents or children aboard`	**then**	$p_s = 35\%$	**else**	$p_s = 51\%$
if	2^{st} or 3^{nd} `class` & `sex` is male	**then**	$p_s = 14\%$	**else**	$p_s = 64\%$
if	`sex` is male & `age` ≥ 15	**then**	$p_s = 16\%$	**else**	$p_s = 72\%$

Decision rule algorithms

Average survival rate $p_s = 39\%$.

if	sex is male	**then**	$p_s = 19\%$	**else**	$p_s = 74\%$
if	1^{st} or 2^{nd} class	**then**	$p_s = 56\%$	**else**	$p_s = 24\%$
if	1^{st} or 2^{nd} class & sex is female	**then**	$p_s = 95\%$	**else**	$p_s = 25\%$
if	fare $< 10.5£$	**then**	$p_s = 20\%$	**else**	$p_s = 50\%$
if	no parents or children aboard	**then**	$p_s = 35\%$	**else**	$p_s = 51\%$
if	2^{st} or 3^{nd} class & sex is male	**then**	$p_s = 14\%$	**else**	$p_s = 64\%$
if	sex is male & age ≥ 15	**then**	$p_s = 16\%$	**else**	$p_s = 72\%$

Many decision rule algorithms among which:

- NodeHarvest [Meinshausen, 2010]
  - Extracts all the rules of a random forests
  - Combines them via solving a constraint quadratic linear program
- RuleFit [Friedman and Popescu, 2008]
  - Extracts all the rules of a boosted tree ensemble
  - Combines them via a logistic regression with lasso penalty

Drawbacks:

- Both methods are unstable: running them several times on the same data set may result in different sets of rules

Unstability

Two runs of RuleFit on the SECOM data set.

rule	coefficient	description
(Intercept)	-1.304499863	1
rule616	-0.400252692	V60 <= 4.97 & V105 > -0.0019 & V424 <= 108.6217
rule26	-0.399674943	V349 <= 0.0385 & V60 <= 8.3918 & V64 <= 17.6454
rule496	-0.265685341	V60 <= 0.8045 & V101 <= 5e-04 & V568 <= 0.0896
rule441	-0.260900593	V60 <= 7.8264 & V583 <= 0.5011 & V350 <= 0.049
rule314	-0.258822916	V22 <= -5512.5 & V472 <= 30.7812
rule508	-0.190299769	V511 <= 95.5975 & V101 <= 5e-04 & V153 <= 0.7523
rule43	-0.177421075	V60 <= 8.3918 & V349 <= 0.0342 & V139 <= 90.8
rule97	-0.134937737	V511 <= 95.3413 & V153 <= 0.7523 & V196 <= 0.361
rule444	-0.117968967	V104 <= -0.0087 & V34 <= 9.1637
rule368	-0.087452989	V104 <= -0.0079 & V153 <= 0.8257
rule395	-0.084409096	V65 <= 25.1618 & V60 <= 9.5927 & V438 <= 7.9865
rule628	-0.084144279	V130 <= 0.0946 & V350 <= 0.0611 & V361 <= 0.0036
rule86	-0.023078885	V125 <= 16.05 & V60 <= 4.9555 & V303 <= 0.45
rule362	-0.003972723	V104 <= -0.0087 & V436 <= 10.2733 & V350 <= 0.0595

rule	coefficient	description
(Intercept)	0.178336422	1
rule97	-0.523012600	V349 <= 0.0421 & V511 <= 200.823 & V60 <= 8.1445
rule282	-0.463529803	V511 <= 65.1163 & V153 <= 0.8257 & V197 <= 14.43
rule606	-0.338103339	V432 <= 99.2163 & V438 <= 7.1906 & V65 <= 30.5136
rule496	-0.297717157	V250 <= 0.0034 & V65 <= 25.1618 & V125 <= 16.05
rule289	-0.278210742	V456 <= 3.7084 & V288 <= 0.3448 & V555 <= 0.852
rule674	-0.272413104	V153 <= 0.7377 & V125 <= 16.04
rule404	-0.266285107	V60 <= 4.9382 & V303 <= 0.4304 & V105 > -0.0017
rule556	-0.261565996	V250 <= 8e-04 & V130 <= 0.0946 & V361 <= 0.0029
rule600	-0.258720261	V512 <= 708.5714 & V558 <= 2.9289 & V65 <= 30.68
rule500	-0.245999282	V22 <= -5394.25 & V438 <= 7.3595
rule461	-0.197524877	V22 <= -5581
rule197	-0.166101239	V104 <= -0.0087 & V301 <= 0.121 & V34 <= 9.7836
rule635	-0.157494908	V334 <= 6.6293 & V366 <= 0.013
rule92	-0.156029423	V349 <= 0.0362 & V511 <= 95.5975 & V438 <= 5.1928
rule130	-0.145965819	V104 <= -0.0087 & V299 <= 0.1024 & V41 > 14
rule140	-0.121309793	V349 <= 0.0369 & V472 <= 21.8646 & V60 <= 4.9991
rule84	-0.120009890	V60 <= 5.4718 & V104 <= -0.0067 & V526 <= 7.5026
rule171	-0.085220151	V334 <= 5.4943
rule595	-0.079847068	V34 <= 8.5891
rule571	-0.078349545	V60 <= 1.6018 & V526 <= 8.8106
rule36	-0.067557526	V60 <= 8.3918 & V511 <= 80.4829 & V349 <= 0.0441
rule361	-0.053981777	V349 <= 0.0369 & V511 <= 167.2026 & V334 <= 6.1301
rule368	-0.041471470	V65 <= 31.4709 & V60 <= 9.8518 & V168 <= 1.1
rule636	-0.037163161	V334 <= 6.6293 & V366 <= 0.013 & V34 <= 9.088
rule150	-0.032344454	V60 <= 4.92 & V349 <= 0.0437 & V288 <= 0.3456
rule448	-0.014851459	V130 <= 0.1892 & V350 <= 0.0595

Unstability

Two runs of NodeHarvest on the SECOM data set.

```
"if V122 > 16 & V52 > 189 then 0.6 (n=10, weight=0.38)"
"if V122 > 16 & V481 < 56 then 0.545 (n=11, weight=0.12)"
"if V511 > 105 & V206 > 14.1 then 0.692 (n=13, weight=0.074)"
"if V65 > 30.7 & V116 < 722 then 0.643 (n=14, weight=0.207)"
"if V60 > 8.36 & V442 > 1.11 then 0.571 (n=14, weight=0.036)"
"if V122 > 16 & V481 > 56 then 0.143 (n=14, weight=0.12)"
"if V122 > 16 & V52 < 189 then 0.133 (n=15, weight=0.38)"
"if V104 < -0.00865 & V435 > 19.7 then 0.263 (n=19, weight=0.027)"
"if V60 < 4.96 & V122 > 16 then 0.304 (n=23, weight=0.027)"
"if V65 > 30.7 & V449 < 0.207 then 0.522 (n=23, weight=0.071)"
"if V60 > 8.41 & V521 < 1.3 then 0.462 (n=26, weight=0.08)"
"if V60 < 4.97 & V572 < 1.21 then 0.258 (n=31, weight=0.019)"
"if V60 < 8.04 & V65 > 33.3 then 0.294 (n=34, weight=0.223)"
"if V60 < 8.14 & V349 > 0.0443 then 0.257 (n=35, weight=0.129)"
"if V60 > 4.96 & V342 > 4.13 then 0.436 (n=39, weight=0.027)"
"if V60 > 8.14 & V334 > 6.76 then 0.475 (n=40, weight=0.352)"
"if V65 > 30.7 & V449 > 0.207 then 0.14 (n=43, weight=0.071)"
"if V65 > 30.7 & V116 > 722 then 0.173 (n=52, weight=0.207)"
"if V60 > 4.95 & V334 > 6.76 then 0.389 (n=54, weight=0.019)"
"if V104 > -0.00865 & V542 > 11.4 then 0.305 (n=82, weight=0.027)"
"if V511 > 105 & V206 < 14.1 then 0.108 (n=83, weight=0.074)"
"if V60 > 8.41 & V588 > 0.0161 then 0.292 (n=106, weight=0.106)"
"if V60 > 8.41 & V588 < 0.0161 then 0.106 (n=132, weight=0.106)"
"if V60 > 8.14 & V334 < 6.76 then 0.132 (n=204, weight=0.129)"
"if V60 > 8.04 & V334 < 6.76 then 0.136 (n=206, weight=0.223)"
"if V60 > 8.41 & V521 > 1.3 then 0.156 (n=212, weight=0.08)"
"if V60 > 8.41 & V171 < 0.971 then 0.165 (n=224, weight=0.036)"
"if V511 < 105 & V60 > 5.49 then 0.156 (n=269, weight=0.074)"
"if V60 > 4.97 & V334 < 6.76 then 0.125 (n=288, weight=0.019)"
"if V60 > 4.96 & V342 < 4.13 then 0.132 (n=304, weight=0.027)"
"if V104 > -0.00865 & V542 < 11.4 then 0.093 (n=388, weight=0.027)"
"if V104 < -0.00865 & V435 < 19.7 then 0.035 (n=1078, weight=0.027)"
"if V60 < 4.97 & V572 > 1.21 then 0.033 (n=1194, weight=0.019)"
"if V60 < 4.96 & V122 < 16 then 0.033 (n=1201, weight=0.027)"
"if V511 < 105 & V60 < 5.49 then 0.037 (n=1202, weight=0.074)"
"if V60 < 8.04 & V65 < 33.3 then 0.037 (n=1287, weight=0.223)"
"if V60 < 8.14 & V349 < 0.0443 then 0.038 (n=1288, weight=0.129)"
"if V60 < 8.41 & V122 < 16 then 0.039 (n=1304, weight=0.222)"
"if V65 < 30.7 & V122 < 16 then 0.053 (n=1476, weight=0.278)"
```

```
"if V104 > -0.00665 & V334 > 7.37 then 0.667 (n=12, weight=0.019)"
"if V65 > 30.7 & V457 < 8.81 then 0.692 (n=13, weight=0.322)"
"if V407 > 14 then 0.385 (n=13, weight=0.017)"
"if V60 < 8.08 & V430 > 10.4 then 0.333 (n=15, weight=0.171)"
"if V60 > 8.08 & V170 > 0.584 then 0.562 (n=16, weight=0.124)"
"if V17 > 10.8 then 0.278 (n=18, weight=0.019)"
"if V60 > 8.08 & V521 < 1.09 then 0.526 (n=19, weight=0.012)"
"if V60 < 8.08 & V122 > 16 then 0.292 (n=24, weight=0.096)"
"if V66 < 36.8 & V122 > 16 then 0.32 (n=25, weight=0.066)"
"if V133 < 2.21 then 0.296 (n=27, weight=0.129)"
"if V60 < 5.02 & V572 < 1.2 then 0.258 (n=31, weight=0.078)"
"if V66 > 36.4 & V342 > 3.19 then 0.455 (n=33, weight=0.066)"
"if V60 < 9.02 & V349 > 0.0438 then 0.25 (n=40, weight=0.124)"
"if V60 > 8.08 & V334 > 6.76 then 0.475 (n=40, weight=0.378)"
"if V60 < 8.94 & V65 > 31.7 then 0.255 (n=47, weight=0.124)"
"if V65 > 30.7 & V457 > 8.81 then 0.17 (n=53, weight=0.322)"
"if V66 > 36.4 & V342 < 3.19 then 0.104 (n=77, weight=0.066)"
"if V60 > 5.02 & V478 > 8.1 then 0.314 (n=86, weight=0.077)"
"if V65 < 30.7 & V60 > 10.9 then 0.189 (n=201, weight=0.176)"
"if V60 > 8.14 & V334 < 6.76 then 0.132 (n=204, weight=0.124)"
"if V60 > 8.08 & V334 < 6.76 then 0.137 (n=205, weight=0.145)"
"if V60 > 8.08 & V521 > 1.09 then 0.164 (n=226, weight=0.012)"
"if V60 > 6.54 & V334 < 6.76 then 0.131 (n=229, weight=0.109)"
"if V104 > -0.00665 & V334 < 7.37 then 0.13 (n=230, weight=0.019)"
```

SIRUS: **S**table and **I**nterpretable **RU**le **S**et

Algorithm principle: Extraction of rules from a random forest

SECOM open dataset - **591** variables / **1567** data points

Average failure rate $p_f = 6.6\%$					
if	$X^{(60)} < 5.51$	**then**	$p_f = 4.2\%$	**else**	$p_f = 16.6\%$
if	$X^{(104)} < -0.01$	**then**	$p_f = 4.0\%$	**else**	$p_f = 13.0\%$
if	$X^{(349)} < 0.04$	**then**	$p_f = 5.4\%$	**else**	$p_f = 17.8\%$
if	$X^{(206)} < 12.7$	**then**	$p_f = 5.4\%$	**else**	$p_f = 17.8\%$
if	$X^{(65)} < 26.1$	**then**	$p_f = 5.5\%$	**else**	$p_f = 17.2\%$
if	$X^{(60)} < 5.51$ **& $X^{(349)} < 0.04$**	**then**	$p_f = 3.5\%$	**else**	$p_f = 16.4\%$

- Predictivity close to Random Forests
- 4 to 5 stable rules

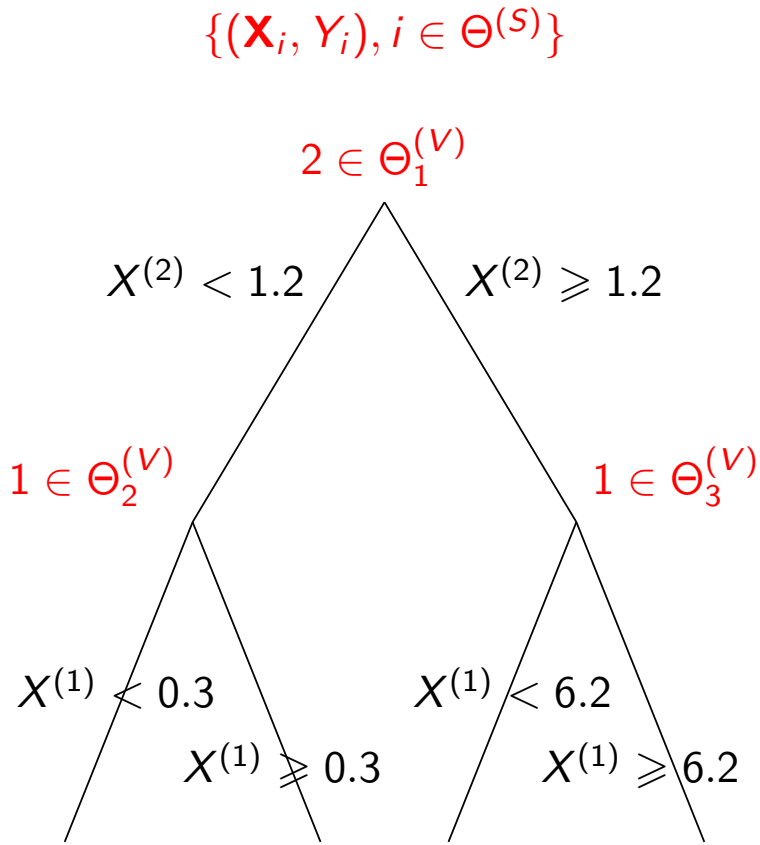
SIRUS - Tree construction

- Classification setting
  $\mathcal{D}_n = \{(\mathbf{X}_i, Y_i), i = 1, \dots, n\}$, $\mathbf{X}_i \in \mathbb{R}^p$,
  $Y \in \{0, 1\}$, $(\mathbf{X}_i, Y_i) \sim \mathbb{P}_{\mathbf{X},Y}$.

SIRUS - Tree construction

- Classification setting
  $\mathcal{D}_n = \{(\mathbf{X}_i, Y_i), i = 1, \dots, n\}$, $\mathbf{X}_i \in \mathbb{R}^p$,
  $Y \in \{0, 1\}$, $(\mathbf{X}_i, Y_i) \sim \mathbb{P}_{\mathbf{X},Y}$.

- Random Forest
  Aggregation of Θ-random trees
  $\Theta = (\Theta^{(S)}, \Theta^{(V)})$

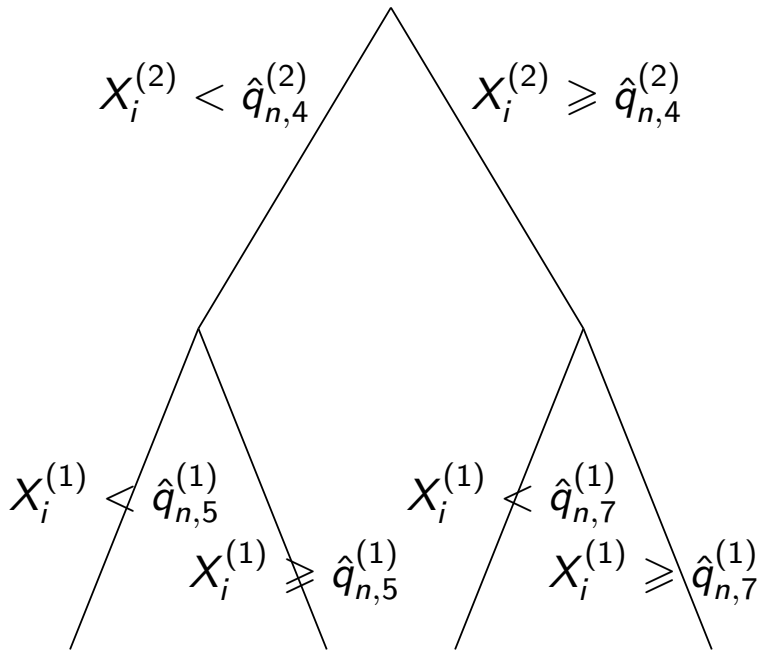


SIRUS - Tree construction

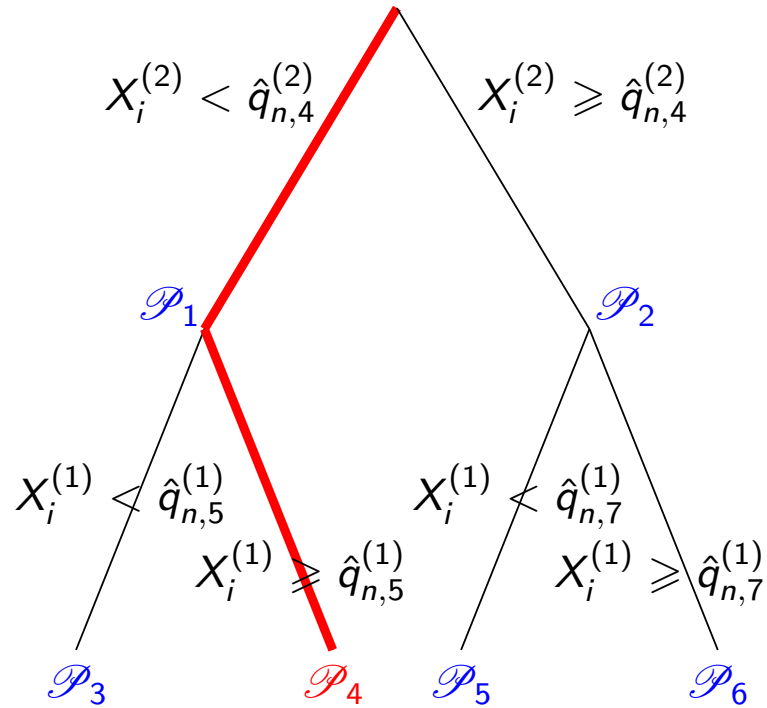
- Classification setting
  $\mathcal{D}_n = \{(\mathbf{X}_i, Y_i), i = 1, \dots, n\}$, $\mathbf{X}_i \in \mathbb{R}^p$,
  $Y \in \{0, 1\}$, $(\mathbf{X}_i, Y_i) \sim \mathbb{P}_{\mathbf{X},Y}$.

- Random Forest
  Aggregation of Θ-random trees
  $\Theta = (\Theta^{(S)}, \Theta^{(V)})$

- Modifications to Breiman's forest
  - Tree depth limited to 2
  - Splits restricted to empirical q-quantiles $\hat{q}_{n,r}^{(j)}$ (typically $q = 10$)

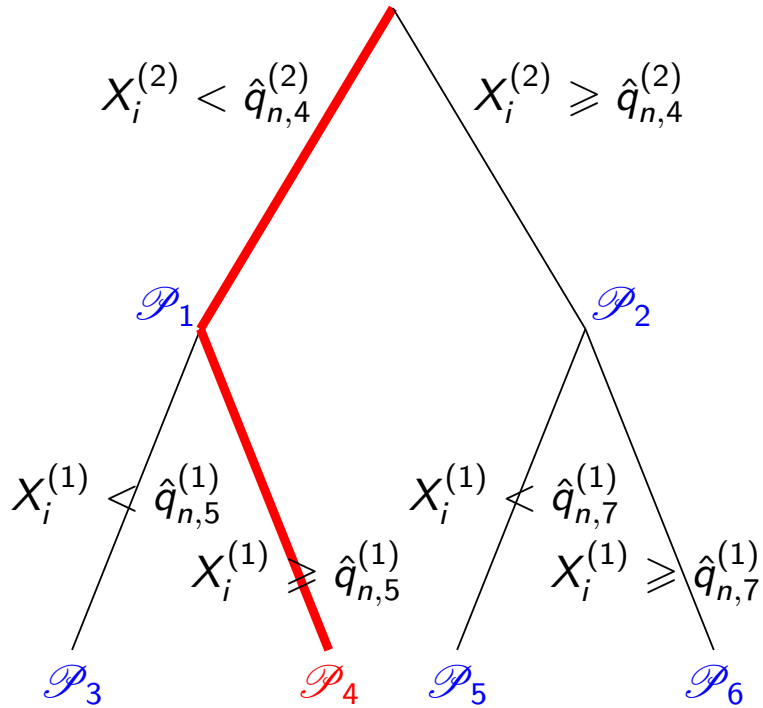


SIRUS - Path


$X_i^{(2)} < \hat{q}_{n,4}^{(2)}$
$X_i^{(2)} \geqslant \hat{q}_{n,4}^{(2)}$
$\mathscr{P}_1$
$\mathscr{P}_2$
$X_i^{(1)} < \hat{q}_{n,5}^{(1)}$
$X_i^{(1)} \geqslant \hat{q}_{n,5}^{(1)}$
$X_i^{(1)} < \hat{q}_{n,7}^{(1)}$
$X_i^{(1)} \geqslant \hat{q}_{n,7}^{(1)}$
$\mathscr{P}_3$
$\mathscr{P}_4$
$\mathscr{P}_5$
$\mathscr{P}_6$

SIRUS - Path

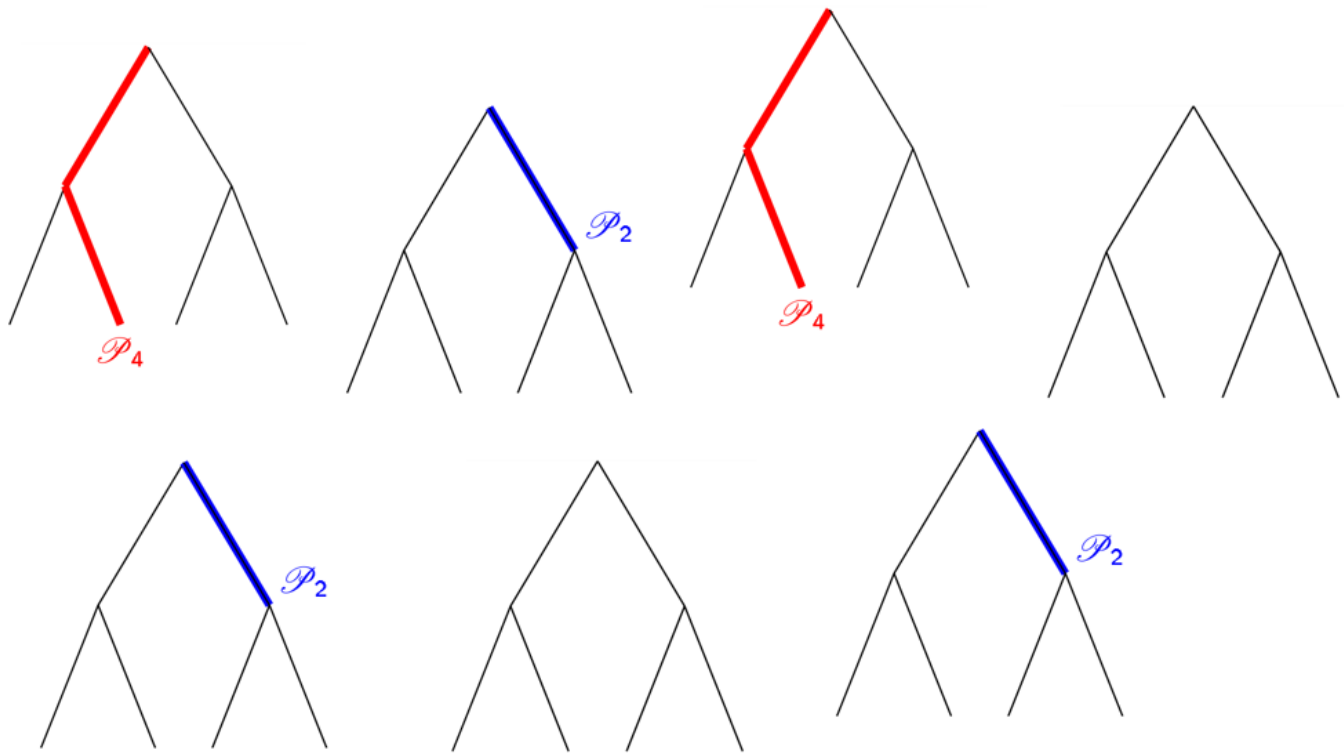
- $T(\Theta, \mathcal{D}_n) = \{\mathscr{P}_1, ..., \mathscr{P}_6\}$: set of paths extracted from a Θ-random tree.
  ($T(\Theta, \mathcal{D}_n) \subset \Pi$, the set of all possible paths)



SIRUS - Path selection

Principle

Frequent paths in random trees = strong and robust patterns in the data.



SIRUS - Path selection

Principle

Frequent paths in random trees = strong and robust patterns in the data.

Frequency of occurence of a given path $\mathscr{P} \in \Pi$ in the random forest with M trees parametrized by $\Theta_1, ..., \Theta_M$

$$\hat{p}_{M,n}(\mathscr{P}) = \frac{1}{M} \sum_{\ell=1}^{M} \mathbb{1}_{\mathscr{P} \in T(\Theta_\ell, \mathcal{D}_n)}$$

SIRUS - Path selection

Principle

Frequent paths in random trees = strong and robust patterns in the data.

Frequency of occurence of a given path $\mathscr{P} \in \Pi$ in the random forest with M trees parametrized by $\Theta_1, ..., \Theta_M$

$$\hat{p}_{M,n}(\mathscr{P}) = \frac{1}{M} \sum_{\ell=1}^{M} \mathbb{1}_{\mathscr{P} \in T(\Theta_\ell, \mathcal{D}_n)}$$

$\hat{p}_{M,n}(\mathscr{P})$ is the Monte-Carlo estimate of the probability that a Θ-random tree contains a given path $\mathscr{P} \in \Pi$

$$p_n(\mathscr{P}) = \mathbb{P}(\mathscr{P} \in T(\Theta, \mathcal{D}_n) | \mathcal{D}_n)$$

SIRUS - Path selection

Principle

Frequent paths in random trees = strong and robust patterns in the data.

Frequency of occurence of a given path $\mathscr{P} \in \Pi$ in the random forest with M trees parametrized by $\Theta_1, ..., \Theta_M$

$$\hat{p}_{M,n}(\mathscr{P}) = \frac{1}{M} \sum_{\ell=1}^{M} \mathbb{1}_{\mathscr{P} \in T(\Theta_\ell, \mathcal{D}_n)}$$

$\hat{p}_{M,n}(\mathscr{P})$ is the Monte-Carlo estimate of the probability that a Θ-random tree contains a given path $\mathscr{P} \in \Pi$

$$p_n(\mathscr{P}) = \mathbb{P}(\mathscr{P} \in T(\Theta, \mathcal{D}_n) | \mathcal{D}_n)$$

Selected paths

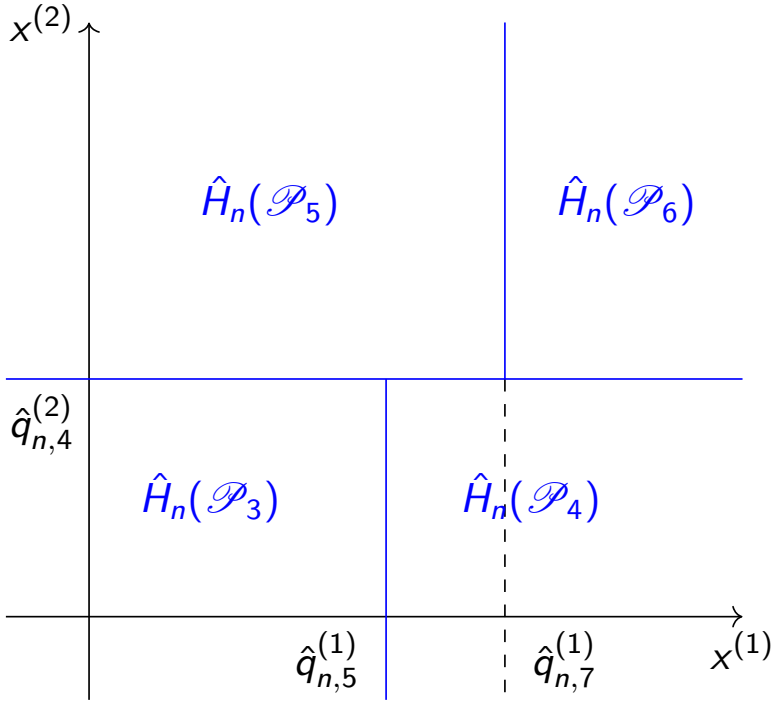
$$\boxed{\hat{\mathscr{P}}_{M,n,p_0} = \{\mathscr{P} \in \Pi : \hat{p}_{M,n}(\mathscr{P}) > p_0\}}$$

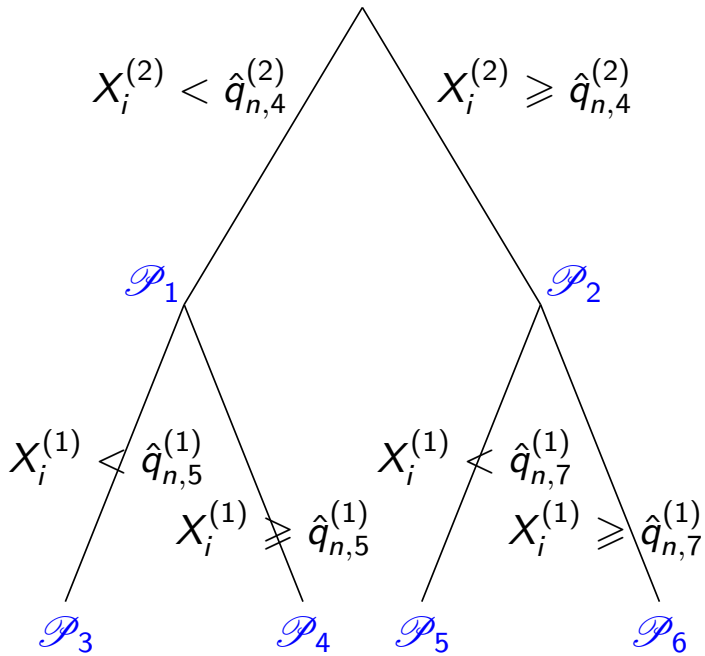
SIRUS - Rule

How to recover a rule from a path ?

SIRUS - Rule

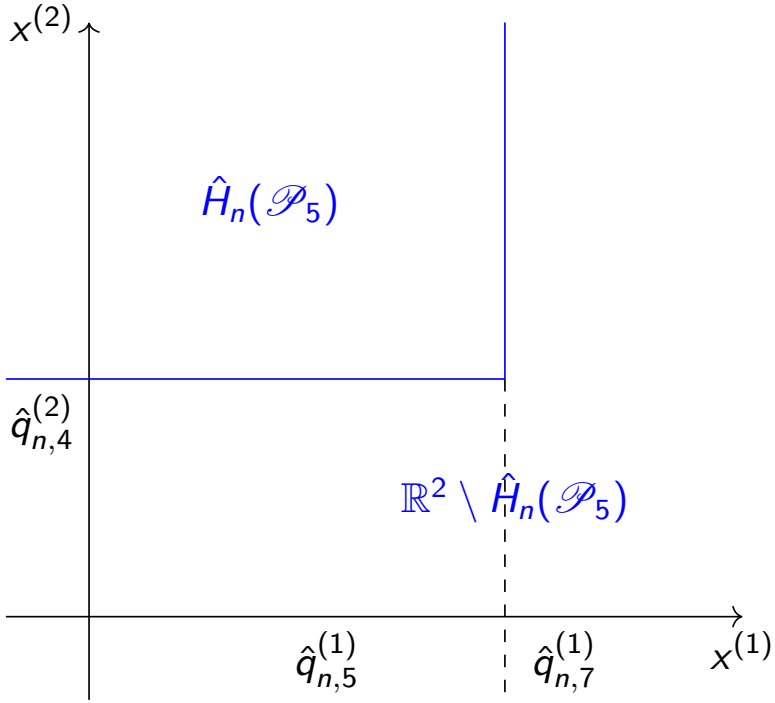
How to recover a rule from a path ?

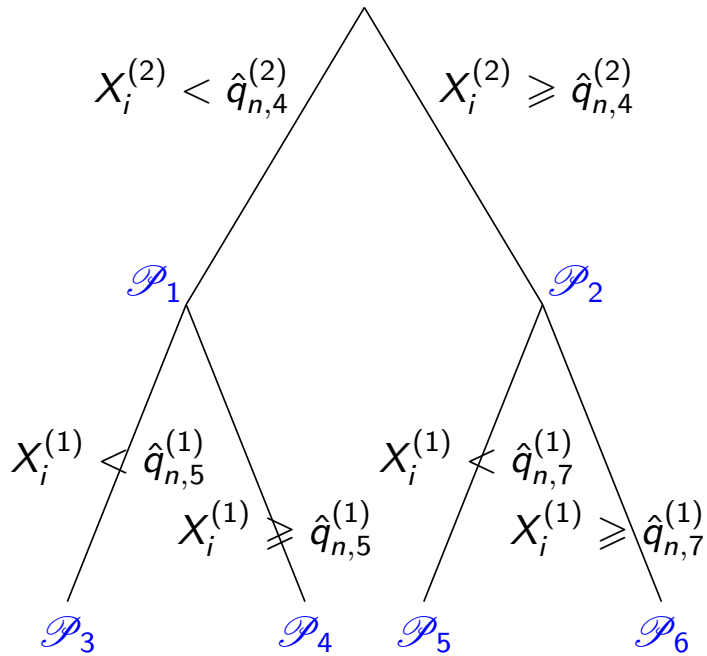




SIRUS - Rule

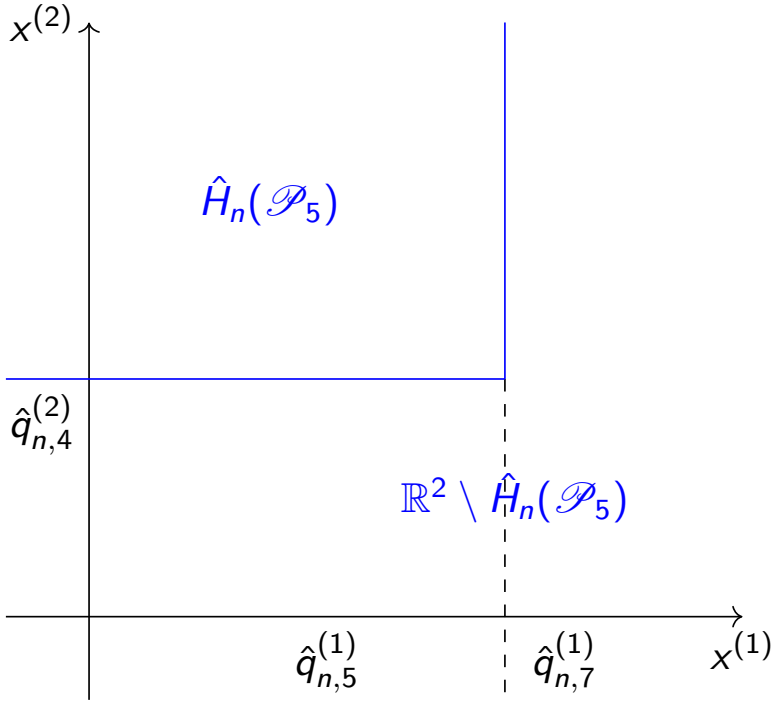
How to recover a rule from a path ?

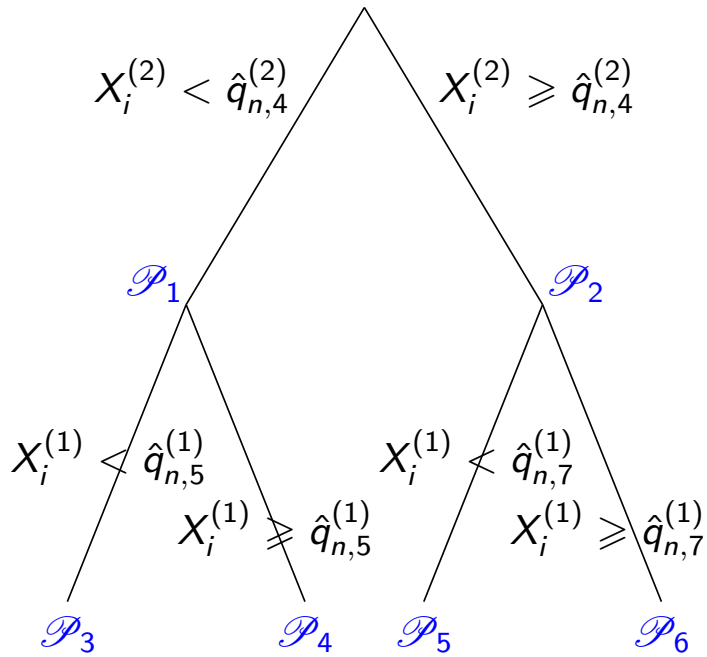




SIRUS - Rule

How to recover a rule from a path ?





$$\forall \mathbf{x} \in \mathbb{R}^p, \quad \hat{g}_{n,\mathscr{P}}(\mathbf{x}) = \begin{cases} \frac{1}{N_n(\hat{H}_n(\mathscr{P}))} \sum_{i=1}^n Y_i \mathbb{1}_{\mathbf{x}_i \in \hat{H}_n(\mathscr{P})} & \text{if } \mathbf{x} \in \hat{H}_n(\mathscr{P}) \\ \frac{1}{n - N_n(\hat{H}_n(\mathscr{P}))} \sum_{i=1}^n Y_i \mathbb{1}_{\mathbf{x}_i \notin \hat{H}_n(\mathscr{P})} & \text{otherwise.} \end{cases}$$

SIRUS - Classifier

Final aggregated estimate of $\eta(\mathbf{x}) = \mathbb{P}(Y = 1|X = x)$

$$\hat{\eta}_{M,n,p_0}(\mathbf{x}) = \frac{1}{|\hat{\mathscr{P}}_{M,n,p_0}|} \sum_{\mathscr{P} \in \hat{\mathscr{P}}_{M,n,p_0}} \hat{g}_{n,\mathscr{P}}(\mathbf{x}).$$

SIRUS - Classifier

Final aggregated estimate of $\eta(\mathbf{x}) = \mathbb{P}(Y = 1|X = x)$

$$\hat{\eta}_{M,n,p_0}(\mathbf{x}) = \frac{1}{|\hat{\mathscr{P}}_{M,n,p_0}|} \sum_{\mathscr{P} \in \hat{\mathscr{P}}_{M,n,p_0}} \hat{g}_{n,\mathscr{P}}(\mathbf{x}).$$

Classification procedure:

- $\hat{Y} = 1$ if $\hat{\eta}_{M,n,p_0}(\mathbf{x}) > s$ $(s \in \mathbb{R})$
- $\hat{Y} = 0$ otherwise.

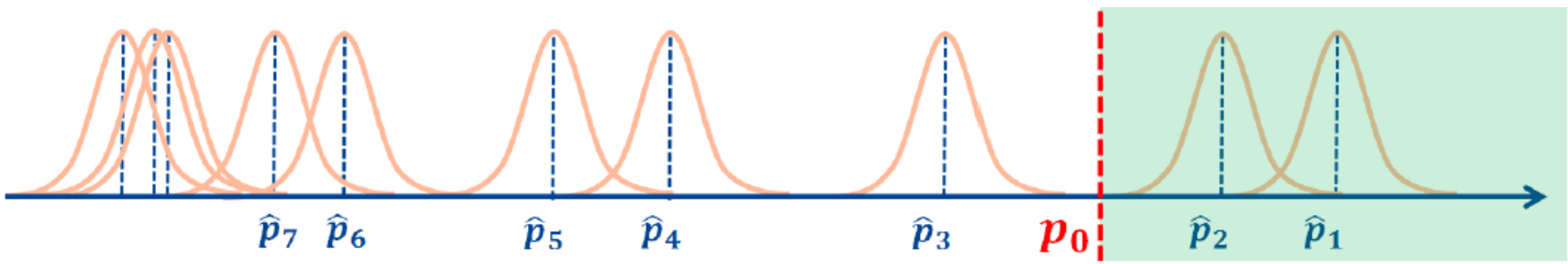
Stability

Define

- $\mathcal{D}'_n$, Θ' independent copies of $\mathcal{D}_n$ and Θ
- $\hat{p}'_{M,n}(\mathscr{P})$, $\hat{\mathscr{P}}'_{M,n,p_0}$ built with $\mathcal{D}'_n$, Θ'

Dice-Sorensen index

$$\hat{S}_{M,n,p_0} = \frac{2\left|\hat{\mathscr{P}}_{M,n,p_0} \cap \hat{\mathscr{P}}'_{M,n,p_0}\right|}{\left|\hat{\mathscr{P}}_{M,n,p_0}\right| + \left|\hat{\mathscr{P}}'_{M,n,p_0}\right|}.$$



Asymptotic Stability

(A1) The subsampling rate a_n satisfies $\lim_{n\to\infty} a_n = \infty$ and $\lim_{n\to\infty} \frac{a_n}{n} = 0$.

(A2) The number of trees M_n satisfies $\lim_{n\to\infty} M_n = \infty$.

(A3) $\mathbf{X}$ has a density f with respect to the Lebesgue measure, continuous, bounded, and strictly positive.

Theorem 1

Assume that Assumptions (A1)-(A3) are satisfied. Then, provided $p_0 \in [0, \max_{\mathscr{P}\in\Pi} p^\star(\mathscr{P})] \setminus \{p^\star(\mathscr{P}) : \mathscr{P} \in \Pi\}$, we have

$$\lim_{n\to\infty} \hat{S}_{M_n,n,p_0} = 1, \quad \text{in probability.}$$

UCI Datasets

- UCI datasets

Dataset	Random Forest	CART	RuleFit	Node Harvest	BRL	SIRUS
Haberman	0.32	0.42	**0.35**	**0.35**	**0.36**	**0.38**
Diabetes	0.17	0.21	**0.19**	**0.20**	0.25	**0.20**
Heart Statlog	0.10	0.17	**0.13**	**0.15**	0.23	**0.13**
Liver Disorders	0.23	0.40	**0.27**	0.31	0.44	0.35
Heart C2	0.10	0.19	**0.11**	**0.11**	0.24	**0.12**
Heart H2	0.12	0.17	**0.11**	**0.11**	0.17	**0.12**
Credit German	0.21	0.31	**0.23**	**0.25**	0.34	**0.26**
Credit Approval	0.07	0.10	**0.07**	**0.07**	0.11	**0.08**
Ionosphere	0.03	0.10	**0.04**	0.07	0.11	0.07

1-AUC (10-fold cross-validation)

UCI Datasets

- UCI datasets

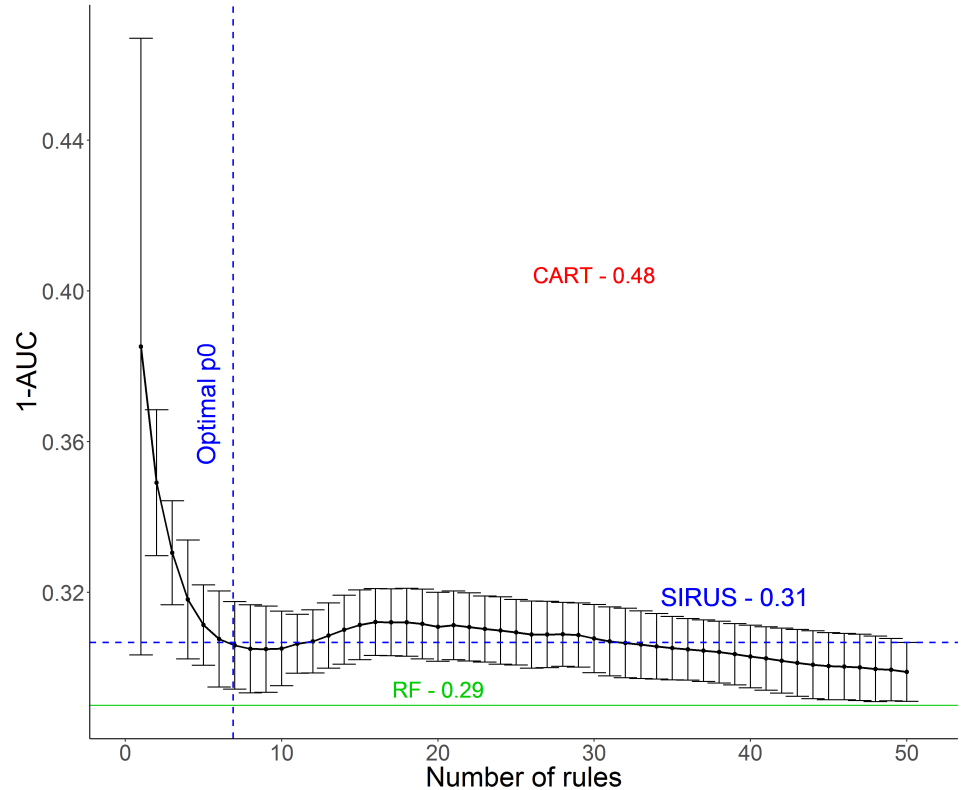
Dataset	RuleFit	Node Harvest	BRL	SIRUS
Haberman	0.57	0.35	**0.71**	0.62
Diabetes	0.21	0.38	**0.80**	**0.74**
Heart Statlog	0.18	0.31	0.34	**0.51**
Liver Disorders	0.19	0.31	0.48	**0.57**
Heart C2	0.28	0.53	**0.66**	**0.64**
Heart H2	0.23	0.37	0.61	**0.75**
Credit German	0.12	0.46	0.33	**0.75**
Credit Approval	0.17	0.26	0.32	**0.44**
Ionosphere	0.06	0.25	**0.78**	0.53

Mean number of rules shared by 2 distinct models in a 10-fold cross-validation

- Preditivity close to RF and stability improved over state of the art algorithms.

SECOM: production data

- Manufacturing process of semi-conductors (public and real data)
  591 variables - **1567** data points
  Unbalanced data: 104 fails - Failure rate: 6.64%



- Stability
  Across a 10-fold cross-validation, **4** to **5** persistent rules between two folds in average.

Conclusion

Average survival rate $p_s = 39\%$.

if	`sex` is male	**then**	$p_s = 19\%$	**else**	$p_s = 74\%$
if	1^{st} or 2^{nd} `class`	**then**	$p_s = 56\%$	**else**	$p_s = 24\%$
if	1^{st} or 2^{nd} `class` & `sex` is female	**then**	$p_s = 95\%$	**else**	$p_s = 25\%$
if	`fare` $< 10.5£$	**then**	$p_s = 20\%$	**else**	$p_s = 50\%$
if	no `parents or children aboard`	**then**	$p_s = 35\%$	**else**	$p_s = 51\%$
if	2^{st} or 3^{nd} `class` & `sex` is male	**then**	$p_s = 14\%$	**else**	$p_s = 64\%$
if	`sex` is male & `age` ≥ 15	**then**	$p_s = 16\%$	**else**	$p_s = 72\%$

- SIRUS behaves well on many datasets
- R/C++ package *sirus* on **CRAN**.
- Can also be applied to regression settings.

Stability

Stability. The problem of defining rules without the else clause lies in the rule selection. Indeed, rules associated with left (L) and right (R) nodes at the first level of a tree are identical:

$$\hat{g}_{n,L}(\mathbf{x}) = \hat{g}_{n,R}(\mathbf{x}) = \bar{Y}_L \mathbf{1}_{\mathbf{x} \in L} + \bar{Y}_R \mathbf{1}_{\mathbf{x} \in R}.$$

Without the else clause, these two rules become different estimates:

$$\hat{h}_{n,L}(\mathbf{x}) = (\bar{Y}_L - \bar{Y}_R)\mathbf{1}_{\mathbf{x} \in L}, \qquad \hat{h}_{n,R}(\mathbf{x}) = (\bar{Y}_R - \bar{Y}_L)\mathbf{1}_{\mathbf{x} \in R}.$$

However, $\hat{h}_{n,L}$ and $\hat{h}_{n,R}$ are linearly dependent, since

$$\hat{h}_{n,L}(\mathbf{x}) - \hat{h}_{n,R}(\mathbf{x}) = \bar{Y}_L - \bar{Y}_R.$$

This linear dependence between predictors makes the linear aggregation of the rules ill-defined. One of two rule could be removed randomly, but this would strongly hurt stability.

Summary

1. Explainability and random forests

2. Decision rules

3. Variable importance
   - A first variable importance in random forests: MDI
   - A second variable importance in random forests: MDA
   - Shapley values via random forests

Variable importance via random forests

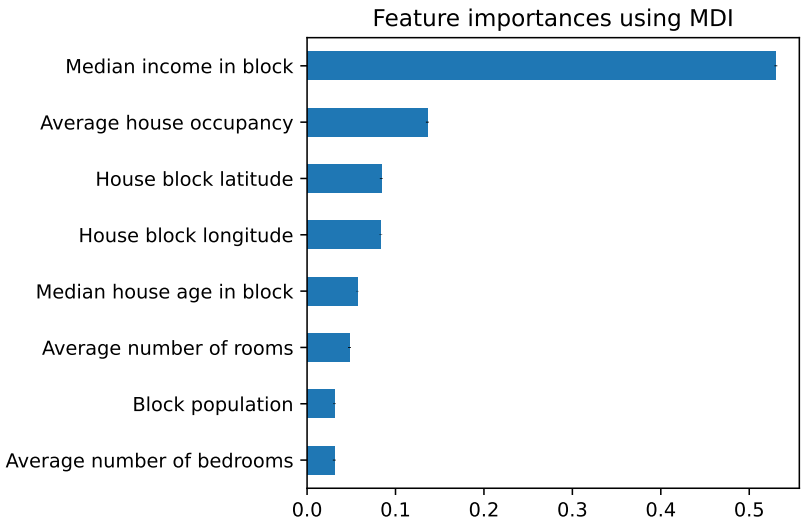


Figure: One of the two variable importance measure, Mean Decrease in Impurity (MDI) computed on the California housing data set.

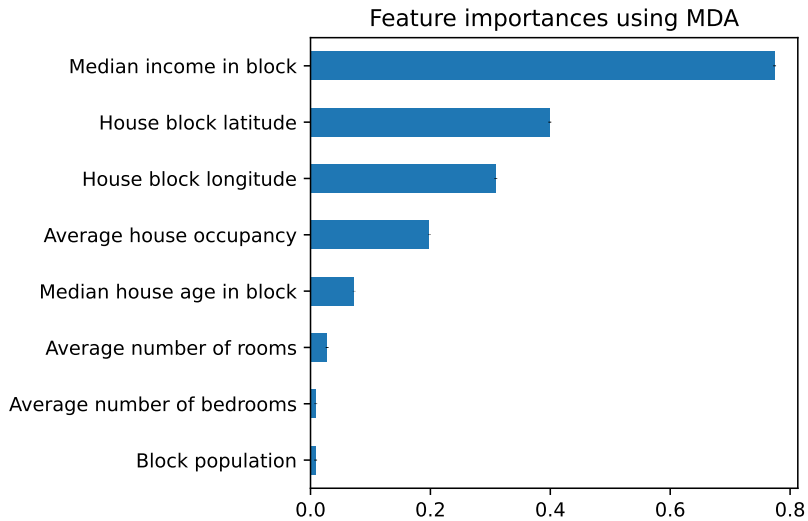


Figure: One of the two variable importance measure, Mean Decrease in Accuracy (MDA) computed on the California housing data set.

Variable importance via random forests

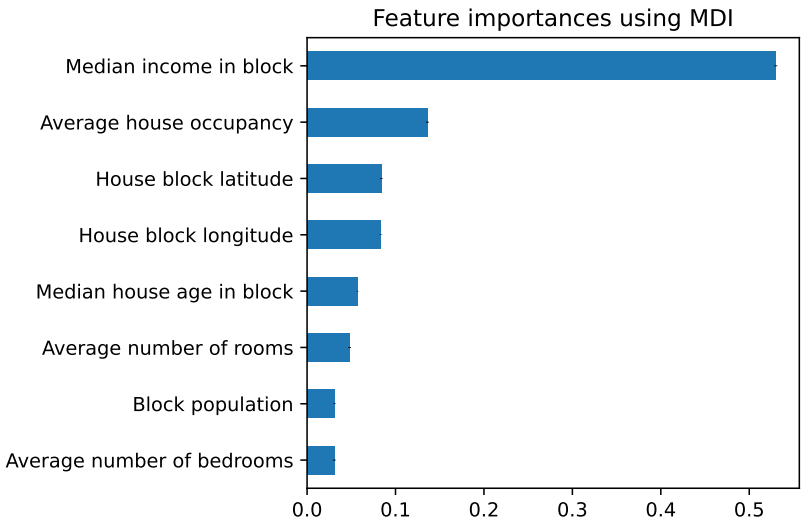


Figure: One of the two variable importance measure, Mean Decrease in Impurity (MDI) computed on the California housing data set.

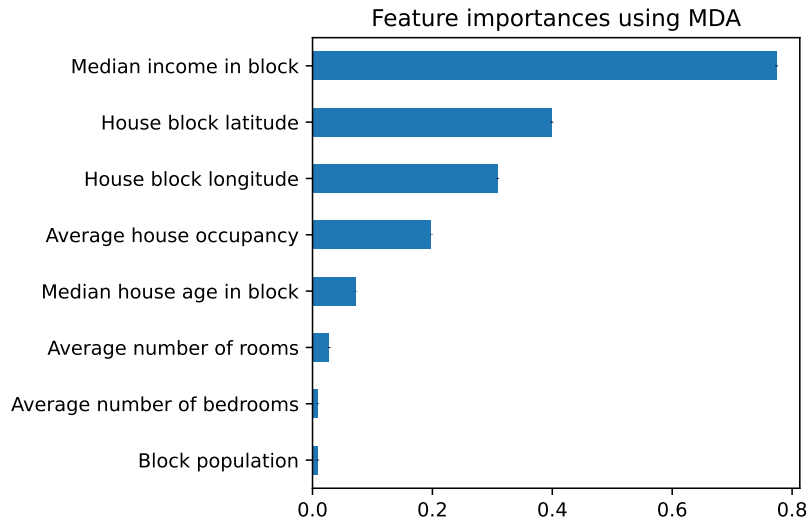


Figure: One of the two variable importance measure, Mean Decrease in Accuracy (MDA) computed on the California housing data set.

- Going beyond prediction to understand the black-box model
- Finding the input variables that are the most "linked" to the output
- Here the variable ranking is not exactly the same across these two different measures.

Variable importance - to what aim?

One single good variable importance measure does not exist. It always depend on what it is used for.

A simple example. Assume that $X \in \mathbb{R}^{10}$, $Y \in \mathbb{R}$ and $Y = X_1$ with $X_1 = g(X_2, \ldots, X_{10})$ for some function g.

- (Variable selection) If one is interested in finding the smallest set of variables leading to good predictive performance, the associated variable importance should be large for X_1 and null for $X_2, \ldots, X_{10}$.
- (Link identification) If one is interested in finding all variables linked to the output, the associated variable importance should be large for $X_1, \ldots, X_d$.

The quality of a variable importance measure depends on its final use (variable selection or link identification).

Variable importance in random forests

Two different measures often computed with random forests:

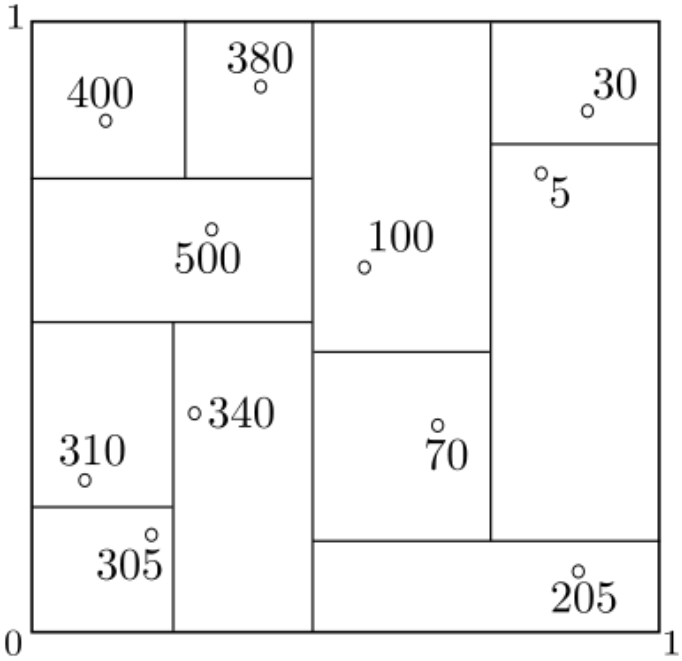
- Mean Decrease Impurity (MDI) [Breiman, 2002]
  - Tailored for decision tree methods
  - Use the decrease in impurity in each node to compute an aggregated variable importance
- Mean Decrease Accuracy (MDA) [also called *permutation importance*, see Breiman, 2001b]
  - Can be used with any supervised learning algorithm (not tree specific)
  - Permute the values of a given feature in the test set and compare the resulting decrease in predictive performance.

1. Explainability and random forests

2. Decision rules

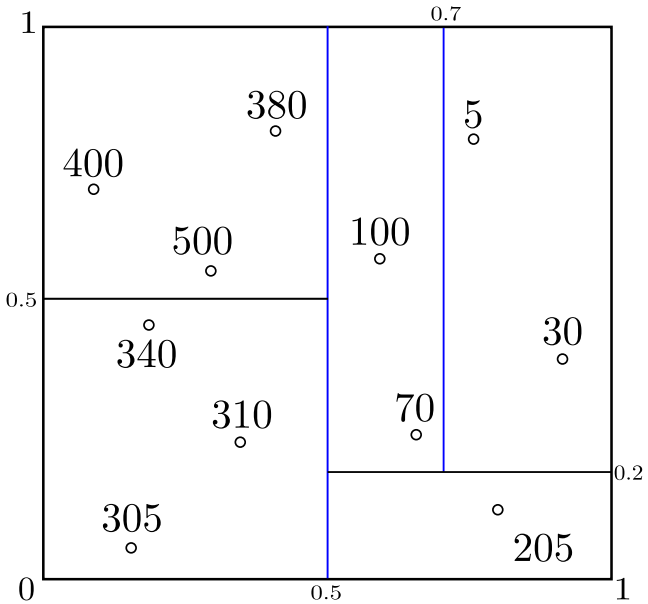
3. Variable importance
   - A first variable importance in random forests: MDI
   - A second variable importance in random forests: MDA
   - Shapley values via random forests

Mean Decrease in impurity



For this given trained tree $\mathcal{T}$, we want to evaluate the MDI of $X^{(1)}$.

Mean Decrease in impurity

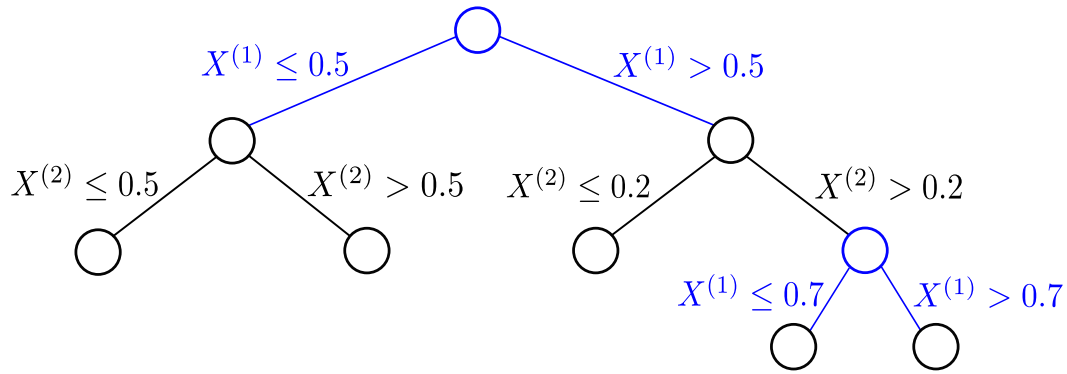


$k = 0$

$k = 1$

$k = 2$

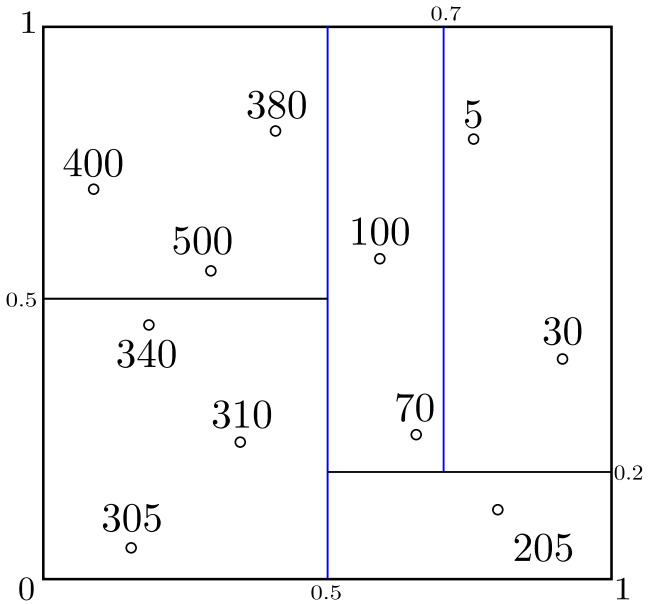
$k = 3$

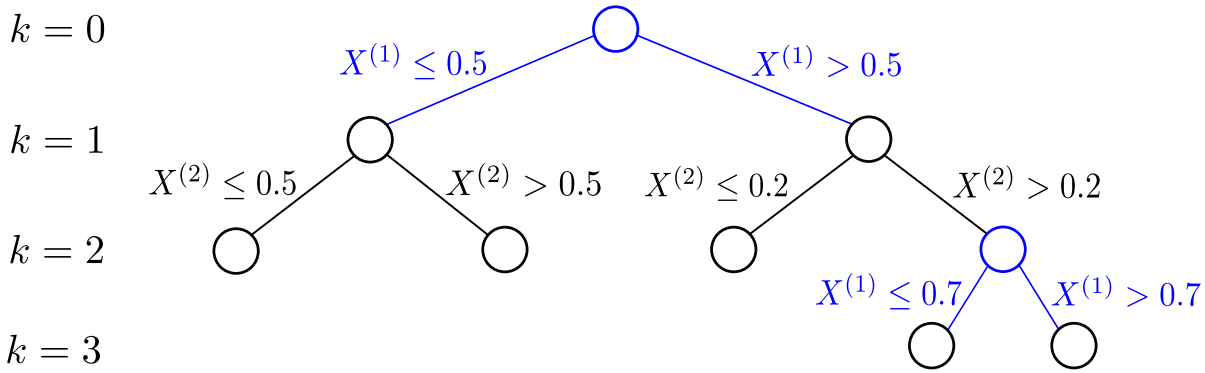


For this given trained tree $\mathcal{T}$, we want to evaluate the MDI of $X^{(1)}$. We proceed as follows:

- Identify all splits that involve variable $X^{(1)}$

Mean Decrease in impurity





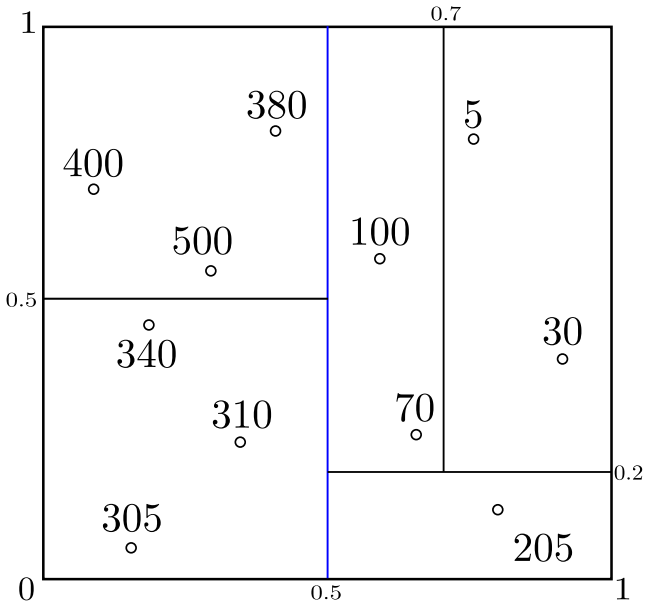
For this given trained tree $\mathcal{T}$, we want to evaluate the MDI of $X^{(1)}$. We proceed as follows:

- Identify all splits that involve variable $X^{(1)}$
- For each split, compute the decrease in impurity between the parent node A and the two resulting nodes A_L and A_R:

$$\Delta Imp_n(A) = Imp_n(A) - p_{L,n} Imp_n(A_L) - p_{R,n} Imp_n(A_R),$$

where $p_{L,n}$ (resp. $p_{R,n}$) is the fraction of observations in A that fall into A_L (resp. A_R). For example, $Imp_n(A) = \mathbb{V}_n[Y|X \in A]$.

Mean Decrease in impurity

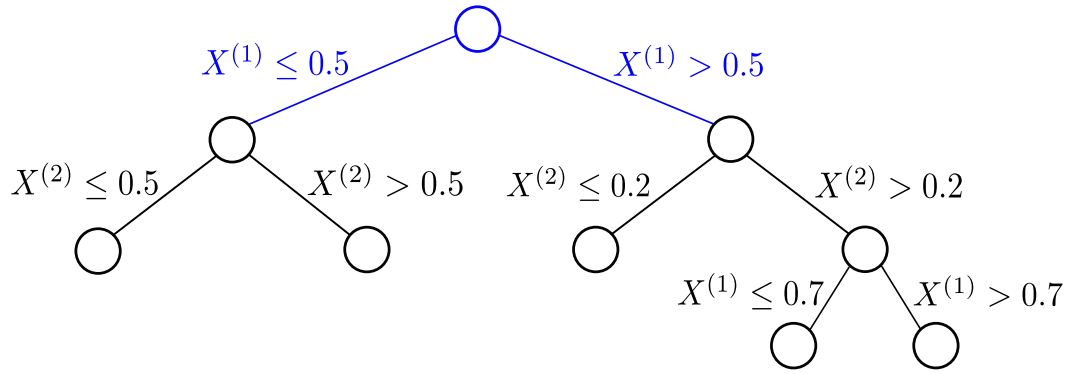


$k = 0$

$k = 1$

$k = 2$

$k = 3$



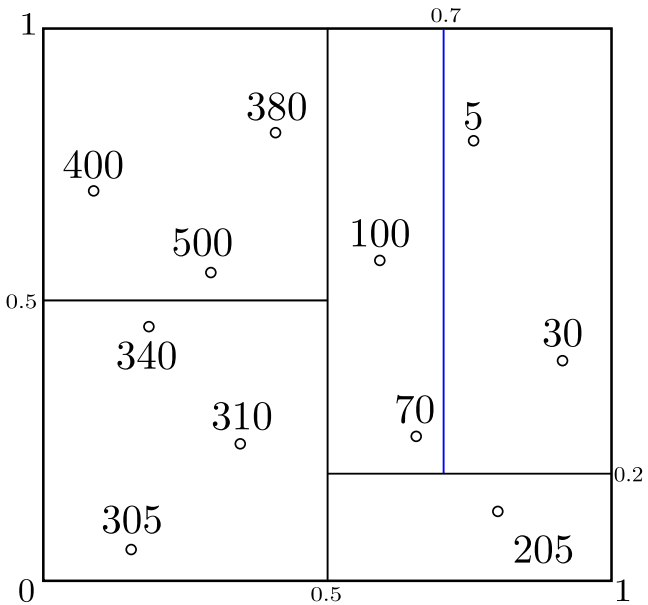
For this given trained tree $\mathcal{T}$, we want to evaluate the MDI of $X^{(1)}$. We proceed as follows:

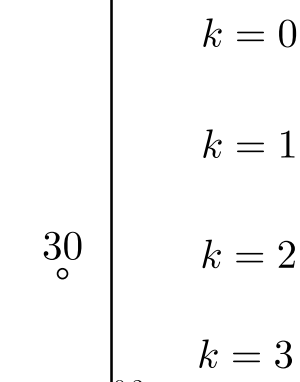
- Identify all splits that involve variable $X^{(1)}$
- For each split, compute the decrease in impurity between the parent node A and the two resulting nodes A_L and A_R:

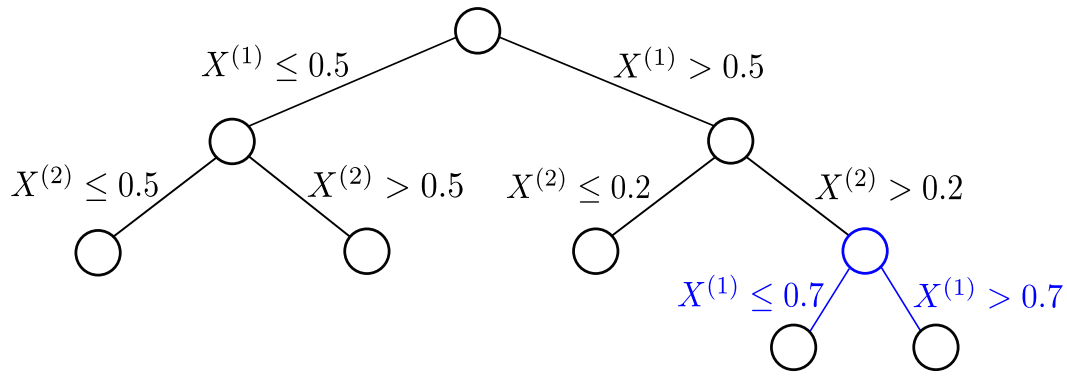
$$\Delta Imp_n(A) = Imp_n(A) - p_{L,n} Imp_n(A_L) - p_{R,n} Imp_n(A_R),$$

where $p_{L,n}$ (resp. $p_{R,n}$) is the fraction of observations in A that fall into A_L (resp. A_R). For example, $Imp_n(A) = \mathbb{V}_n[Y|X \in A]$.

Mean Decrease in impurity







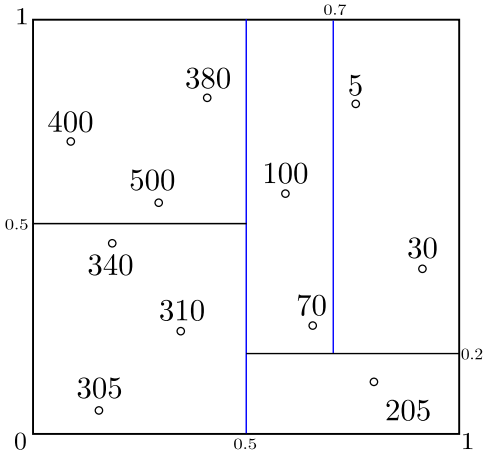
For this given trained tree $\mathcal{T}$, we want to evaluate the MDI of $X^{(1)}$. We proceed as follows:

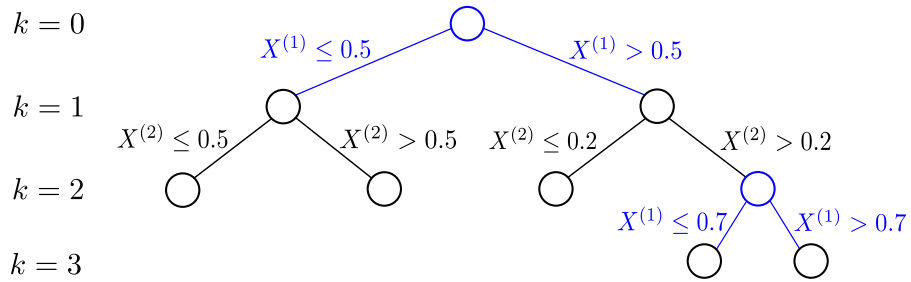
- Identify all splits that involve variable $X^{(1)}$
- For each split, compute the decrease in impurity between the parent node A and the two resulting nodes A_L and A_R:

$$\Delta Imp_n(A) = Imp_n(A) - p_{L,n} Imp_n(A_L) - p_{R,n} Imp_n(A_R),$$

where $p_{L,n}$ (resp. $p_{R,n}$) is the fraction of observations in A that fall into A_L (resp. A_R). For example, $Imp_n(A) = \mathbb{V}_n[Y|X \in A]$.

Mean Decrease in impurity





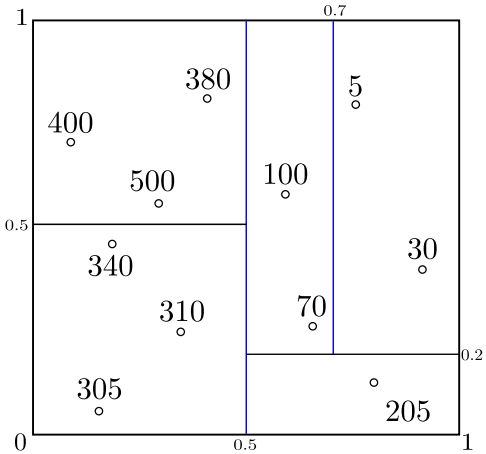
For this given trained tree $\mathcal{T}$, we want to evaluate the MDI of $X^{(1)}$. We proceed as follows:

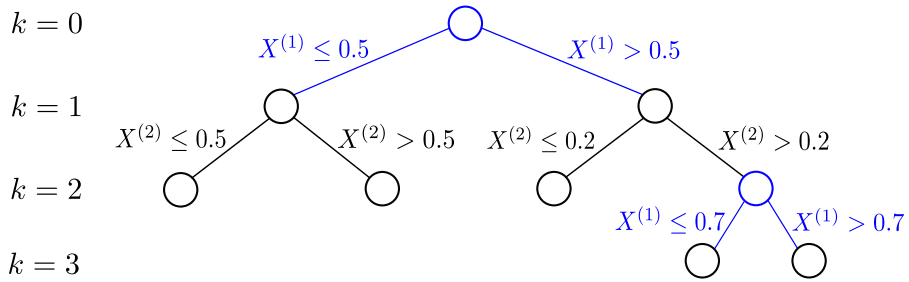
- Identify all splits that involve variable $X^{(1)}$
- For each split, compute the decrease in impurity $\Delta Imp_n(A)$ between the parent node A and the two resulting nodes A_L and A_R
- The MDI of $X^{(1)}$ computed via this tree $\mathcal{T}$ is

$$\widehat{\mathrm{MDI}}_{\mathcal{T}}(X^{(1)}) = \sum_{\substack{A \in \mathcal{T} \\ j_{n,A}=1}} p_{n,A}\ \Delta Imp_n(A), \tag{1}$$

where the sum ranges over all cells A in $\mathcal{T}$ that are split along variable j and $p_{A,n}$ is the fraction of observations falling into A

Mean Decrease in impurity





For this given trained tree $\mathcal{T}$, we want to evaluate the MDI of $X^{(1)}$. We proceed as follows:

- Identify all splits that involve variable $X^{(1)}$
- For each split, compute the decrease in impurity $\Delta Imp_n(A)$ between the parent node A and the two resulting nodes A_L and A_R
- The MDI of $X^{(1)}$ computed via this tree $\mathcal{T}$ is

$$\widehat{\mathrm{MDI}}_{\mathcal{T}}(X^{(1)}) = \sum_{\substack{A \in \mathcal{T} \\ j_{n,A}=1}} p_{n,A}\ \Delta Imp_n(A) \tag{1}$$

- The MDI of $X^{(1)}$ output by a forest is the average of the MDI of $X^{(1)}$ of each tree.

Literature

Empirically known flaws of MDI:

- biased towards variables with many categories [see, e.g., Strobl et al., 2007, Nicodemus, 2011]
- biased towards variables that possess high-category frequency [Nicodemus, 2011, Boulesteix et al., 2011]
- biased in presence of correlated features [Nicodemus and Malley, 2009]

Designing new tree building procedure:

- Select splits via a permutation test [Strobl et al., 2008, 2009].

Theoretical work on MDI:

- Louppe et al. [2013] study of theoretical MDI when all variables are categorical.
- Bias related to in-sample estimation [Li et al., 2019, Zhou and Hooker, 2019]

First result

Proposition [Scornet, 2022]

Let $\mathcal{T}_n$ be the CART tree, based on the data set $\mathcal{D}_n$. Then,

$$\widehat{\mathbb{V}[Y]} = \sum_{j=1}^{d} \widehat{\mathrm{MDI}}_{\mathcal{T}_n}(X^{(j)}) + R_n(\hat{m}_{\mathcal{T}_n}), \tag{2}$$

where $\hat{m}_{\mathcal{T}_n}$ is the estimate associated to $\mathcal{T}_n$.

- Valid for many tree building processes (telescopic sums)
- Relation between $\widehat{\mathrm{MDI}}$ and R^2:

$$R^2 = \frac{\sum_{j=1}^{d} \widehat{\mathrm{MDI}}_{\mathcal{T}_n}(X^{(j)})}{\widehat{\mathbb{V}[Y]}}$$

- MDI, computed with fully-grown trees is positively biased:

$$\lim_{n\to\infty} \sum_{j=1}^{d} \widehat{\mathrm{MDI}}_{\mathcal{T}_n}(X^{(j)}) = \mathbb{V}[m(\mathbf{X})] + \sigma^2.$$

Additive models

Definition: Additive model

The regression model writes $Y = \sum_{j=1}^{d} m_j(X^{(j)}) + \varepsilon$, where each m_j is continuous; ε is a Gaussian noise $\mathcal{N}(0, \sigma^2)$, independent of $\mathbf{X}$; and $\mathbf{X} \sim \mathcal{U}([0,1]^d)$.

Theorem [Additive model, Scornet, 2022]

Assume that the Additive Model holds. Let $\mathcal{T}_n$ be the empirical CART tree. Then, for all $\gamma > 0, \rho \in (0,1]$, there exists K such that, for all $k > K$, for all n large enough, with probability at least $1 - \rho$, for all j,

$$\left| \widehat{\mathrm{MDI}}_{\mathcal{T}_{n,k}}(X^{(j)}) - \mathbb{V}[m_j(X^{(j)})] \right| \leqslant \gamma.$$

Additive model - theoretical results

Theorem [Additive model, Scornet, 2022]

Assume that the Additive Model holds. Let $\mathcal{T}_n$ be the empirical CART tree. Then, for all $\gamma > 0, \rho \in (0, 1]$, there exists K such that, for all $k > K$, for all n large enough, with probability at least $1 - \rho$, for all j,

$$\left| \widehat{\mathrm{MDI}}_{\mathcal{T}_{n,k}}(X^{(j)}) - \mathbb{V}[m_j(X^{(j)})] \right| \leqslant \gamma.$$

- MDI targets the same value as MDA (up to a constant 2).
- MDI targets the right quantity in an additive model with independent features.
  $\rightarrow$ MDI can be used to rank and select variables in this context
- MDI is consistent when computed with shallow trees.

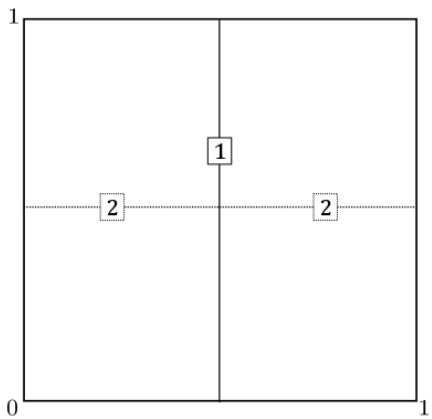
Moving beyond additivity

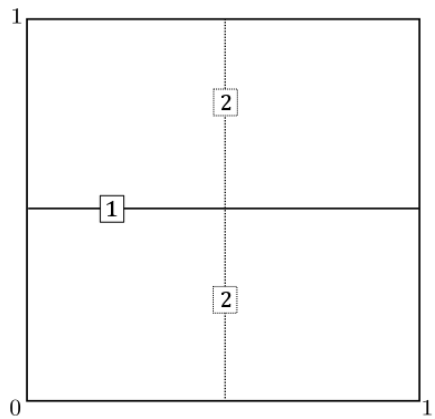
Model (Multiplicative model)

Let $\alpha \in \mathbb{R}$. *The regression model writes* $Y = 2^d \alpha \prod_{j=1}^{d} X^{(j)} + \varepsilon$, *where* $\alpha \in \mathbb{R}$; ε *is a Gaussian noise* $\mathcal{N}(0, \sigma^2)$, *independent of* $\mathbf{X}$; *and* $\mathbf{X} \sim \mathcal{U}([0,1]^d)$.

- This model contains interactions between all input variables
- There exists many theoretical trees

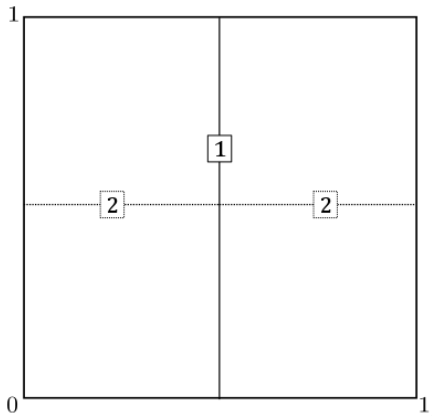
An example in dimension two:

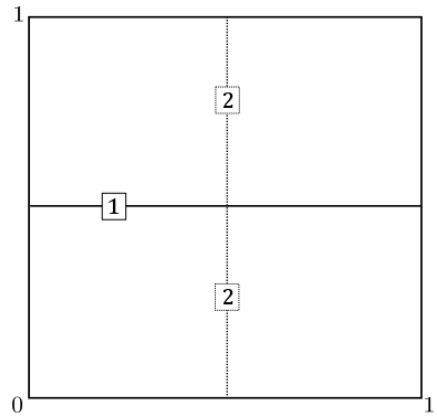




A negative result in presence of interactions

Two theoretical trees in the previous multiplicative model:

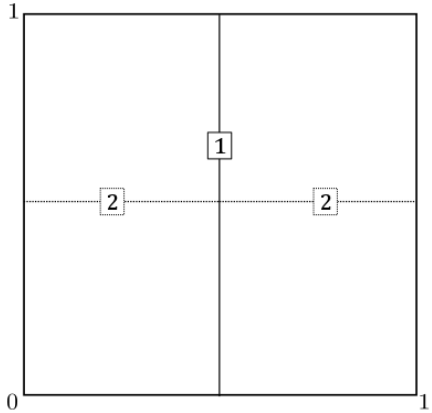


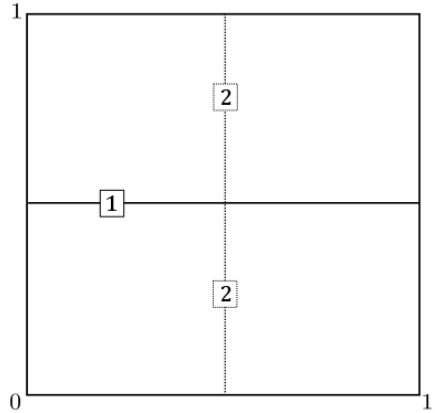


- In this example, the splits in the second level are associated with larger decreases in impurity/variance.
- In presence of interactions, the splits with the largest decreases in variance are not always in the first level of the tree!

A negative result in presence of interactions

Two theoretical trees in the previous multiplicative model:





Recall that Model 1 corresponds to $Y = 2^d \alpha \prod_{j=1}^{d} X^{(j)} + \varepsilon$.

Lemma [Scornet, 2022]

Assume that Model 1 holds. Then, there exists two theoretical trees $\mathcal{T}_1$ and $\mathcal{T}_2$ such that

$$\lim_{k \to \infty} \left(\mathrm{MDI}^{\star}_{\mathcal{T}_{2,k}}(X^{(1)}) - \mathrm{MDI}^{\star}_{\mathcal{T}_{1,k}}(X^{(1)}) \right) = \alpha^2/16.$$

- MDI computed with a single tree is ill-defined

A correlation framework

Correlated Model

Let $\beta \in \mathbb{N}$. Assume that $Y = X^{(1)} + X^{(2)} + \alpha X^{(3)} + \varepsilon$, where $(X^{(1)}, X^{(2)}) \sim \mathcal{U}^{\otimes 2^\beta}$, $X^{(3)} \sim \mathcal{U}([0,1])$ is independent of $(X^{(1)}, X^{(2)})$, and ε is an independent noise distributed as $\mathcal{N}(0, \sigma^2)$.

The distribution $\mathcal{U}^{\otimes 2^\beta}$ is defined as $\mathcal{U}^{\otimes 2^\beta} = \mathcal{U}\left(\cup_{j=0}^{2^\beta - 1} \left[\frac{j}{2^\beta}, \frac{j+1}{2^\beta}\right]^2\right)$

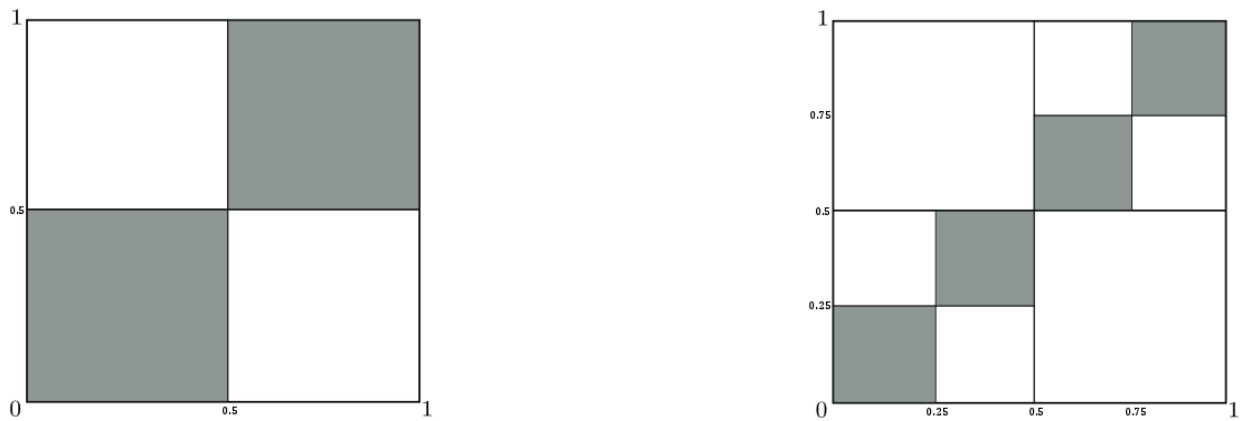


Figure: Illustration of $\mathcal{U}^{\otimes 2^\beta}$, with $\beta = 1$ (left) and $\beta = 2$ (right).

Correlated Model

Let $\beta \in \mathbb{N}$. Assume that $Y = X^{(1)} + X^{(2)} + \alpha X^{(3)} + \varepsilon$, where $(X^{(1)}, X^{(2)}) \sim \mathcal{U}^{\otimes 2^\beta}$, $X^{(3)} \sim \mathcal{U}([0,1])$ is independent of $(X^{(1)}, X^{(2)})$, and ε is an independent noise distributed as $\mathcal{N}(0, \sigma^2)$.

Lemma

Let $\beta \in \{0, \dots, 5\}$. Assume that the Correlated Model holds. Then, there exists two theoretical trees $\mathcal{T}_1$ and $\mathcal{T}_2$ such that

$$\lim_{k \to \infty} \left(MDI^\star_{\mathcal{T}_{2,k}}(X^{(1)}) - MDI^\star_{\mathcal{T}_{1,k}}(X^{(1)}) \right) = \frac{1}{3} - \frac{1}{3}\left(\frac{1}{4}\right)^\beta.$$

- Many theoretical trees exist.
- MDI computed with a single tree is ill-defined in this model (correlated design).

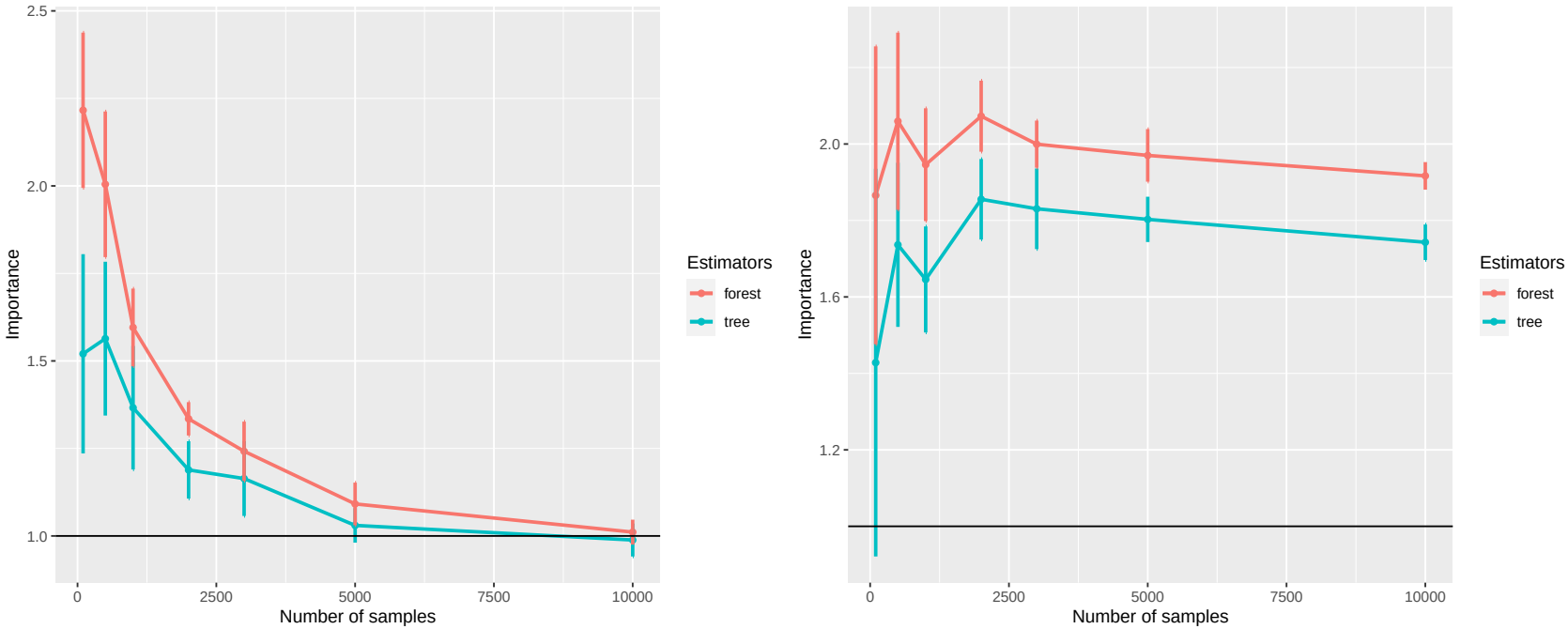
Experiments

We let $Y = \alpha_1 X^{(1)} + \alpha_2 X^{(2)} + \alpha_3 X^{(3)} + \varepsilon$, where ε is an independent noise, distributed as $\mathcal{N}(0, \sigma^2)$ and $(X^{(1)}, X^{(2)}, X^{(3)})$ is distributed as $\mathcal{N}(0, \Sigma)$ where

$$\Sigma = \begin{pmatrix} 1 & \rho & 0 \\ \rho & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

For all j, we let $\alpha_j = \sqrt{j}$:

- The variable importance of the j-th component is j (for $\rho = 0$)
- Studying the impact of the noise σ^2 is easier in this setting.

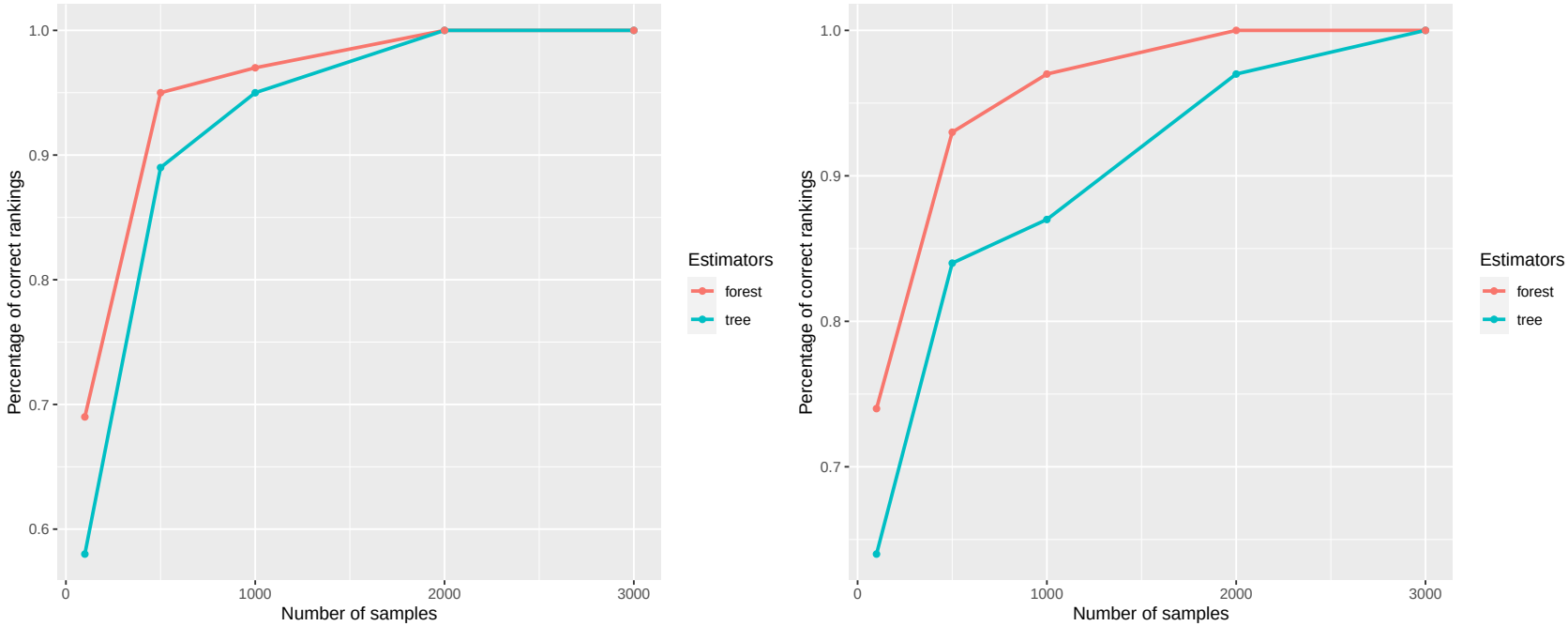


((a)) ((b))

Figure: Importance of the first variable in the previous simulated model, with $\sigma^2 = 3, \rho = 0$ and, from left to right $\texttt{maxnodes} = \lfloor n^{0.6} \rfloor, n$

In presence of noise, the MDI of the first variable is

- positively biased if computed with a fully-grown tree/forest.
- unibased if computed with an early-stopped tree/forest



((a)) ((b))

Figure: Percent of correct ranking in the previous simulated model, with $\sigma^2 = 12$ and, from left to right $\texttt{maxnodes} = \lfloor n^{0.6} \rfloor, n$

- Despite the fact that the MDIs are biased, the correct order is accurately retrieved.
- An early-stopped tree/forest produces more accurate rankings than a fully grown tree/forest.

Take-home messages on MDI

- If input variables are independent and in absence of interactions, using MDI to rank variable is ok:
  - Proved for an early-stopped tree/forest
  - Empirically correct for fully-grown tree/forest
- In presence of correlation or interaction, the empirical MDI computed with a single tree does not converge, and therefore should not be used.
- In presence of correlation or interaction, the empirical MDI computed with a forest targets a quantity which is currently unknown.

Take-home messages on MDI

- If input variables are independent and in absence of interactions, using MDI to rank variable is ok:
  - Proved for an early-stopped tree/forest
  - Empirically correct for fully-grown tree/forest
- In presence of correlation or interaction, the empirical MDI computed with a single tree does not converge, and therefore should not be used.
- In presence of correlation or interaction, the empirical MDI computed with a forest targets a quantity which is currently unknown.
- If fully-grown trees/forests are used, the sum of MDI does not converge to the explained variance of the model.
- Experimentally, in an additive model with no correlation, the variance due to the noise seems to be split equitably between all MDIs which does not affect the variable ranking.

Take-home message on MDI

Do not use MDI!

We do not know what quantity is targeted.

Alternatives that circumvent some flaws have been proposed:

- Out-of-sample estimation [Li et al., 2019, Zhou and Hooker, 2021, Loecher, 2022] with code in python:

  `https://github.com/ZhengzeZhou/unbiased-feature-importance`

- Better to compute it with shallow trees (by default, trees in RF are very deep).

Anyway, remember to check the predictive performance of a model: it it is low, the model is useless and variable importances are misleading.

1. Explainability and random forests

2. Decision rules

3. Variable importance
   - A first variable importance in random forests: MDI
   - A second variable importance in random forests: MDA
   - Shapley values via random forests

MDA

- MDA [Breiman, 2001a]
  - Built-in variable importance algorithm for random forests

MDA

- MDA [Breiman, 2001a]
  - Built-in variable importance algorithm for random forests
  - MDA principle: decrease of accuracy of the forest when a variable is noised up

MDA

- MDA [Breiman, 2001a]
  - Built-in variable importance algorithm for random forests
  - MDA principle: decrease of accuracy of the forest when a variable is noised up
  - MDA is used intensively (intuitive and fast)

MDA

- MDA [Breiman, 2001a]
  - Built-in variable importance algorithm for random forests
  - MDA principle: decrease of accuracy of the forest when a variable is noised up
  - MDA is used intensively (intuitive and fast)

- MDA has flaws
  - Poor understanding of the MDA: what is estimated?
  - Empirical studies show that the MDA is biased for dependent inputs [Strobl et al., 2007, Gregorutti et al., 2017, Hooker and Mentch, 2019].

MDA

- MDA [Breiman, 2001a]
  - Built-in variable importance algorithm for random forests
  - MDA principle: decrease of accuracy of the forest when a variable is noised up
  - MDA is used intensively (intuitive and fast)

- MDA has flaws
  - Poor understanding of the MDA: what is estimated?
  - Empirical studies show that the MDA is biased for dependent inputs [Strobl et al., 2007, Gregorutti et al., 2017, Hooker and Mentch, 2019].

- Our objective is twofold:
  - Theoretical analysis of the MDA
    - Existing results only for simplified MDA [Ishwaran, 2007, Zhu et al., 2015]
    - Theoretical understanding of MDA bias
  - Design of Sobol-MDA algorithm to fix the MDA flaws

MDA illustration

$X^{(1)}$	$X^{(2)}$	$\ldots$	$X^{(j)}$	$\ldots$	$X^{(p)}$	Y
2.1	4.3	$\ldots$	0.1	$\ldots$	2.6	2.3
1.7	4.1	$\ldots$	9.2	$\ldots$	3.8	0.4
3.4	9.2	$\ldots$	3.2	$\ldots$	3.6	10.2
5.6	1.2	$\ldots$	8.2	$\ldots$	4.2	9.1
8.9	6.8	$\ldots$	6.7	$\ldots$	2.9	4.5

Table: Example of the permutation of a dataset $\mathcal{D}_n$ for $n = 5$.

MDA illustration

$X^{(1)}$	$X^{(2)}$	$\ldots$	$X^{(j)}$	$\ldots$	$X^{(p)}$	Y
2.1	4.3	$\ldots$	0.1	$\ldots$	2.6	2.3
1.7	4.1	$\ldots$	9.2	$\ldots$	3.8	0.4
3.4	9.2	$\ldots$	3.2	$\ldots$	3.6	10.2
5.6	1.2	$\ldots$	8.2	$\ldots$	4.2	9.1
8.9	6.8	$\ldots$	6.7	$\ldots$	2.9	4.5

Table: Example of the permutation of a dataset $\mathcal{D}_n$ for $n = 5$.

MDA illustration

$X^{(1)}$	$X^{(2)}$	...	$X^{(j)}$	...	$X^{(p)}$	Y
2.1	4.3	...	0.1	...	2.6	2.3
1.7	4.1	...	9.2	...	3.8	0.4
3.4	9.2	...	3.2	...	3.6	10.2
5.6	1.2	...	8.2	...	4.2	9.1
8.9	6.8	...	6.7	...	2.9	4.5

$X^{(1)}$	$X^{(2)}$	...	$X^{(j)}$	...	$X^{(p)}$	Y
2.1	4.3	...	6.7	...	2.6	2.3
1.7	4.1	...	3.2	...	3.8	0.4
3.4	9.2	...	9.2	...	3.6	10.2
5.6	1.2	...	0.1	...	4.2	9.1
8.9	6.8	...	8.2	...	2.9	4.5

Table: Example of the permutation of a dataset $\mathcal{D}_n$ for $n = 5$.

MDA illustration

$X^{(1)}$	$X^{(2)}$	...	$X^{(j)}$	...	$X^{(p)}$	Y
2.1	4.3	...	0.1	...	2.6	2.3
1.7	4.1	...	9.2	...	3.8	0.4
3.4	9.2	...	3.2	...	3.6	10.2
5.6	1.2	...	8.2	...	4.2	9.1
8.9	6.8	...	6.7	...	2.9	4.5

$X^{(1)}$	$X^{(2)}$	...	$X^{(j)}$	...	$X^{(p)}$	Y
2.1	4.3	...	6.7	...	2.6	2.3
1.7	4.1	...	3.2	...	3.8	0.4
3.4	9.2	...	9.2	...	3.6	10.2
5.6	1.2	...	0.1	...	4.2	9.1
8.9	6.8	...	8.2	...	2.9	4.5

Table: Example of the permutation of a dataset $\mathcal{D}_n$ for $n = 5$.

quadratic error $= 13.7$ quadratic error $= 16.4$

$$\mathrm{MDA}(X^{(j)}) = 16.4 - 13.7 = 2.7$$

MDA illustration

$X^{(1)}$	$X^{(2)}$	...	$X^{(j)}$	...	$X^{(p)}$	Y
2.1	4.3	...	0.1	...	2.6	2.3
1.7	4.1	...	9.2	...	3.8	0.4
3.4	9.2	...	3.2	...	3.6	10.2
5.6	1.2	...	8.2	...	4.2	9.1
8.9	6.8	...	6.7	...	2.9	4.5

$X^{(1)}$	$X^{(2)}$	...	$X^{(j)}$	...	$X^{(p)}$	Y
2.1	4.3	...	6.7	...	2.6	2.3
1.7	4.1	...	3.2	...	3.8	0.4
3.4	9.2	...	9.2	...	3.6	10.2
5.6	1.2	...	0.1	...	4.2	9.1
8.9	6.8	...	8.2	...	2.9	4.5

Table: Example of the permutation of a dataset $\mathcal{D}_n$ for $n = 5$.

quadratic error $= 13.7$ quadratic error $= 16.4$

$$\mathrm{MDA}(X^{(j)}) = 16.4 - 13.7 = 2.7$$

- $\mathrm{MDA}(X^{(j)}) = 0 \longrightarrow$ no influence of $X^{(j)}$
- $\mathrm{MDA}(X^{(j)})$ is high $\longrightarrow$ strong influence of $X^{(j)}$

MDA illustration

$X^{(1)}$	$X^{(2)}$	...	$X^{(j)}$	...	$X^{(p)}$	Y
2.1	4.3	...	0.1	...	2.6	2.3
1.7	4.1	...	9.2	...	3.8	0.4
3.4	9.2	...	3.2	...	3.6	10.2
5.6	1.2	...	8.2	...	4.2	9.1
8.9	6.8	...	6.7	...	2.9	4.5

$X^{(1)}$	$X^{(2)}$	...	$X^{(j)}$	...	$X^{(p)}$	Y
2.1	4.3	...	6.7	...	2.6	2.3
1.7	4.1	...	3.2	...	3.8	0.4
3.4	9.2	...	9.2	...	3.6	10.2
5.6	1.2	...	0.1	...	4.2	9.1
8.9	6.8	...	8.2	...	2.9	4.5

Table: Example of the permutation of a dataset $\mathcal{D}_n$ for $n = 5$.

quadratic error $= 13.7$ quadratic error $= 16.4$

$$\mathrm{MDA}(X^{(j)}) = 16.4 - 13.7 = 2.7$$

$\mathcal{D}_n$ used to fit the forest and compute accuracy: overfitting and inflated accuracy

MDA versions

The explained variance estimate of MDA algorithms differ across implementations

Train-Test MDA: train data to fit the forest, and test data for accuracy

MDA versions

The explained variance estimate of MDA algorithms differ across implementations

Train-Test MDA: train data to fit the forest, and test data for accuracy

Out-of-bag (OOB) samples: $\mathcal{D}_n$ is bootstrap prior to the construction of each tree, leaving aside a portion of $\mathcal{D}_n$, which is not involved in the tree growing and defines the "out-of-bag" sample.

$X^{(1)}$	$X^{(2)}$	$\ldots$	$X^{(j)}$	$\ldots$	$X^{(p)}$	Y
2.1	4.3	$\ldots$	0.1	$\ldots$	2.6	2.3
1.7	4.1	$\ldots$	9.2	$\ldots$	3.8	0.4
3.4	9.2	$\ldots$	3.2	$\ldots$	3.6	10.2
5.6	1.2	$\ldots$	8.2	$\ldots$	4.2	9.1
8.9	6.8	$\ldots$	6.7	$\ldots$	2.9	4.5

Selected samples: $\Theta_\ell^{(S)} = \{1, 3, 4\}$

MDA versions

The explained variance estimate of MDA algorithms differ across implementations

Train-Test MDA: train data to fit the forest, and test data for accuracy

Out-of-bag (OOB) samples: $\mathcal{D}_n$ is bootstrap prior to the construction of each tree, leaving aside a portion of $\mathcal{D}_n$, which is not involved in the tree growing and defines the "out-of-bag" sample.

$X^{(1)}$	$X^{(2)}$	$\ldots$	$X^{(j)}$	$\ldots$	$X^{(p)}$	Y
2.1	4.3	$\ldots$	0.1	$\ldots$	2.6	2.3
1.7	4.1	$\ldots$	9.2	$\ldots$	3.8	0.4
3.4	9.2	$\ldots$	3.2	$\ldots$	3.6	10.2
5.6	1.2	$\ldots$	8.2	$\ldots$	4.2	9.1
8.9	6.8	$\ldots$	6.7	$\ldots$	2.9	4.5

OOB samples: $\{1, \ldots, n\} \setminus \Theta_\ell^{(S)} = \{2, 5\}$

MDA versions

The explained variance estimate of MDA algorithms differ across implementations

Train-Test MDA: train data to fit the forest, and test data for accuracy

Out-of-bag (OOB) samples: $\mathcal{D}_n$ is bootstrap prior to the construction of each tree, leaving aside a portion of $\mathcal{D}_n$, which is not involved in the tree growing and defines the "out-of-bag" sample.

MDA Version	Package	Error	Data
Train-Test	`scikit-learn` `randomForestSRC`	Forest	Testing dataset
Breiman-Cutler	`randomForest` (normalized) `ranger` / `randomForestSRC`	Tree	OOB sample
Ishwaran-Kogalur	`randomForestSRC`	Forest	OOB sample

Table: Summary of the different MDA algorithms.

Assumptions

(A1)

The response $Y \in \mathbb{R}$ *follows*

$$Y = m(\mathbf{X}) + \varepsilon$$

where

- $\mathbf{X} = (X^{(1)}, \ldots, X^{(p)}) \in [0,1]^p$
- $\mathbf{X}$ *admits a density* f *such that* $c_1 < f(\mathbf{x}) < c_2$, *with constants* $c_1, c_2 > 0$
- m *is continuous*
- *the noise* ε *is sub-Gaussian and centered*

Assumptions

(A2): the theoretical tree is consistent

Assumptions

(A2): the theoretical tree is consistent

(A2)

The randomized theoretical CART tree built with the distribution of $(\mathbf{X}, Y)$ *is consistent, that is, for all* $\mathbf{x} \in [0,1]^p$*, almost surely,*

$$\lim_{k \to \infty} \Delta(m, A_k^\star(\mathbf{x}, \Theta)) = 0.$$

Assumptions

(A2): the theoretical tree is consistent

(A2)

The randomized theoretical CART tree built with the distribution of $(\mathbf{X}, Y)$ *is consistent, that is, for all* $\mathbf{x} \in [0,1]^p$*, almost surely,*

$$\lim_{k \to \infty} \Delta(m, A_k^\star(\mathbf{x}, \Theta)) = 0.$$

(A3): tree partition is not too complex with respect to n

Assumptions

(A2): the theoretical tree is consistent

(A2)

The randomized theoretical CART tree built with the distribution of $(\mathbf{X}, Y)$ *is consistent, that is, for all* $\mathbf{x} \in [0,1]^p$*, almost surely,*

$$\lim_{k \to \infty} \Delta(m, A_k^\star(\mathbf{x}, \Theta)) = 0.$$

(A3): tree partition is not too complex with respect to n

(A3)

The asymptotic regime of a_n*, the size of the subsampling without replacement, and the number of terminal leaves* t_n *is such that* $a_n \leqslant n-2$, $a_n/n < 1-\kappa$ *for a fixed* $\kappa > 0$, $\lim_{n\to\infty} a_n = \infty$, $\lim_{n\to\infty} t_n = \infty$, *and* $\lim_{n\to\infty} t_n \frac{(\log(a_n))^9}{a_n} = 0$.

MDA Convergence

Theorem (Bénard et al. [2022a])

If Assumptions (A1), (A2), and (A3) are satisfied, then, for all $M \in \mathbb{N}^\star$ *and* $j \in \{1, \dots, p\}$ *we have*

$$\widehat{MDA}^{(BC)}_{M,n}(X^{(j)}) \xrightarrow{\mathbb{L}^1} \mathbb{E}[(m(\mathbf{X}) - m(\mathbf{X}_{\pi_j}))^2]$$

$\mathbf{X}_{\pi_j}$: $\mathbf{X}$ where the j-th component is replaced by an independent copy, i.e. $\mathbf{X}_{\pi_j} = (X^{(1)}, \dots, X'^{(j)}, \dots, X^{(p)})$

Limit interpretation?

MDA Decomposition

Total Sobol index [Sobol, 1993]

$$ST^{(j)} = \frac{\mathbb{E}[\mathbb{V}(m(\mathbf{X})|\mathbf{X}^{(-j)})]}{\mathbb{V}(Y)}$$

MDA Decomposition

Total Sobol index [Sobol, 1993]

$$ST^{(j)} = \frac{\mathbb{E}[\mathbb{V}(m(\mathbf{X})|\mathbf{X}^{(-j)})]}{\mathbb{V}(Y)}$$

Marginal total Sobol index

$$ST^{(j)}_{mg} = \frac{\mathbb{E}[\mathbb{V}(m(\mathbf{X}_{\pi_j})|\mathbf{X}^{(-j)})]}{\mathbb{V}(Y)}$$

MDA Decomposition

Total Sobol index [Sobol, 1993]

$$ST^{(j)} = \frac{\mathbb{E}[\mathbb{V}(m(\mathbf{X})|\mathbf{X}^{(-j)})]}{\mathbb{V}(Y)}$$

Marginal total Sobol index

$$ST^{(j)}_{mg} = \frac{\mathbb{E}[\mathbb{V}(m(\mathbf{X}_{\pi_j})|\mathbf{X}^{(-j)})]}{\mathbb{V}(Y)}$$

Proposition (Bénard et al. [2022a])

If Assumptions (A1), (A2) and (A3) are satisfied, then for all $M \in \mathbb{N}^\star$ and $j \in \{1, \ldots, p\}$ we have

$$\widehat{MDA}^{(BC)}_{M,n}(X^{(j)}) \xrightarrow{\mathbb{L}^1} \mathbb{V}[Y] \times ST^{(j)} + \mathbb{V}[Y] \times ST^{(j)}_{mg} + MDA^{\star(j)}_3.$$

The term $\mathrm{MDA}^{\star(j)}_3$ is not an importance measure and is defined by

$$\mathrm{MDA}^{\star(j)}_3 = \mathbb{E}[(\mathbb{E}[m(\mathbf{X})|\mathbf{X}^{(-j)}] - \mathbb{E}[m(\mathbf{X}_{\pi_j})|\mathbf{X}^{(-j)}])^2].$$

MDA Decomposition

Proposition (Bénard et al. [2022a])

If Assumptions (A1), (A2) and (A3) are satisfied, then for all $M \in \mathbb{N}^\star$ *and* $j \in \{1, \dots, p\}$ *we have*

$$(i) \quad \widehat{MDA}^{(TT)}_{M,n}(X^{(j)}) \xrightarrow{\mathbb{L}^1} \mathbb{V}[Y] \times ST^{(j)} + \mathbb{V}[Y] \times ST^{(j)}_{mg} + MDA^{\star(j)}_3$$

$$(ii) \quad \widehat{MDA}^{(BC)}_{M,n}(X^{(j)}) \xrightarrow{\mathbb{L}^1} \mathbb{V}[Y] \times ST^{(j)} + \mathbb{V}[Y] \times ST^{(j)}_{mg} + MDA^{\star(j)}_3.$$

If additionally $M \longrightarrow \infty$, *then*

$$(iii) \quad \widehat{MDA}^{(IK)}_{M,n}(X^{(j)}) \xrightarrow{\mathbb{L}^1} \mathbb{V}[Y] \times ST^{(j)} + MDA^{\star(j)}_3.$$

Independent inputs

If inputs X are independent: $\mathrm{MDA}_3^{\star(j)} = 0$ and $ST^{(j)} = ST_{mg}^{(j)}$.

Corollary (Bénard et al. [2022a])

If $\mathbf{X}$ *has independent components, and if Assumptions (A1)-(A3) are satisfied, for all* $M \in \mathbb{N}^\star$ *and* $j \in \{1, \ldots, p\}$ *we have*

$$\widehat{MDA}_{M,n}^{(TT)}(X^{(j)}) \xrightarrow{\mathbb{L}^1} 2\mathbb{V}[Y] \times ST^{(j)}$$
$$\widehat{MDA}_{M,n}^{(BC)}(X^{(j)}) \xrightarrow{\mathbb{L}^1} 2\mathbb{V}[Y] \times ST^{(j)}.$$

If additionally $M \longrightarrow \infty$, *then*

$$\widehat{MDA}_{M,n}^{(IK)}(X^{(j)}) \xrightarrow{\mathbb{L}^1} \mathbb{V}[Y] \times ST^{(j)}.$$

This Corollary completes the result from [Gregorutti, 2015].

MDA summary

- When inputs $\mathbf{X}$ are dependent and have interactions, the MDA is artificially inflated by the term MDA_3 and is therefore misleading.

MDA summary

- When inputs $\mathbf{X}$ are dependent and have interactions, the MDA is artificially inflated by the term MDA_3 and is therefore misleading.
- For variable selection, the total Sobol index is the relevant component

$$\mathbb{V}[Y] \times ST^{(j)} + \cancel{\mathbb{V}[Y] \times ST_{mg}^{(j)}} + \cancel{\mathrm{MDA}_3^{*(j)}}$$

MDA summary

- When inputs $\mathbf{X}$ are dependent and have interactions, the MDA is artificially inflated by the term MDA_3 and is therefore misleading.
- For variable selection, the total Sobol index is the relevant component

$$\mathbb{V}[Y] \times ST^{(j)} + \cancel{\mathbb{V}[Y] \times ST_{mg}^{(j)}} + \cancel{\mathrm{MDA}_3^{*(j)}}$$

- We develop the Sobol-MDA: a fast and consistent estimate of $ST^{(j)}$ for random forests

Sobol-MDA

Principle: **project** the partition of each tree along the j-th direction to remove $X^{(j)}$ from the prediction process.

Sobol-MDA

Principle: **project** the partition of each tree along the j-th direction to remove $X^{(j)}$ from the prediction process.

$$\widehat{\text{S-MDA}}_{M,n}(X^{(j)}) = \frac{1}{\hat{\sigma}^2_Y} \frac{1}{n} \sum_{i=1}^{n} \left[Y_i - m^{(-j,OOB)}_{M,n}(\mathbf{X}_i^{(-j)}, \boldsymbol{\Theta}_M) \right]^2 - \left[Y_i - m^{(OOB)}_{M,n}(\mathbf{X}_i, \boldsymbol{\Theta}_M) \right]^2$$

Sobol-MDA

Principle: **project** the partition of each tree along the j-th direction to remove $X^{(j)}$ from the prediction process.

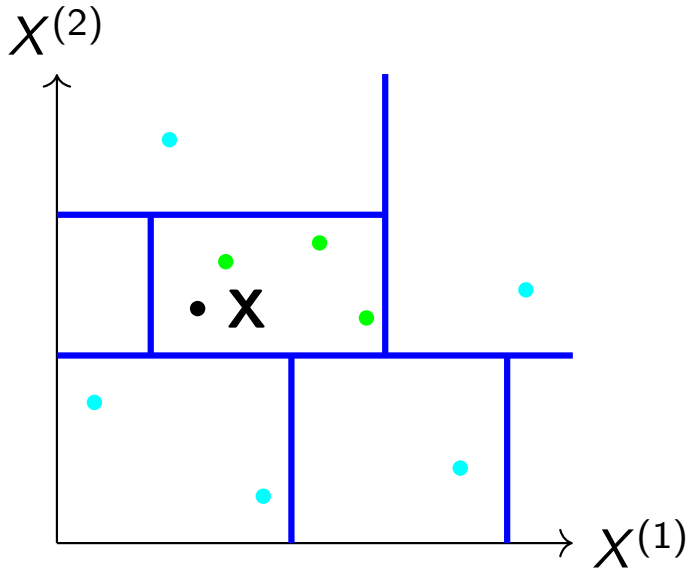


Figure: Partition of $[0,1]^2$ by a random tree (left side) projected on the subspace span by $\mathbf{X}^{(-2)} = X^{(1)}$ (right side), for $p = 2$ and $j = 2$.

$$\widehat{\text{S-MDA}}_{M,n}(X^{(j)}) = \frac{1}{\hat{\sigma}_Y^2} \frac{1}{n} \sum_{i=1}^{n} \left[Y_i - m_{M,n}^{(-j,OOB)}(\mathbf{X}_i^{(-j)}, \boldsymbol{\Theta}_M) \right]^2 - \left[Y_i - m_{M,n}^{(OOB)}(\mathbf{X}_i, \boldsymbol{\Theta}_M) \right]^2$$

Sobol-MDA

Principle: **project** the partition of each tree along the j-th direction to remove $X^{(j)}$ from the prediction process.

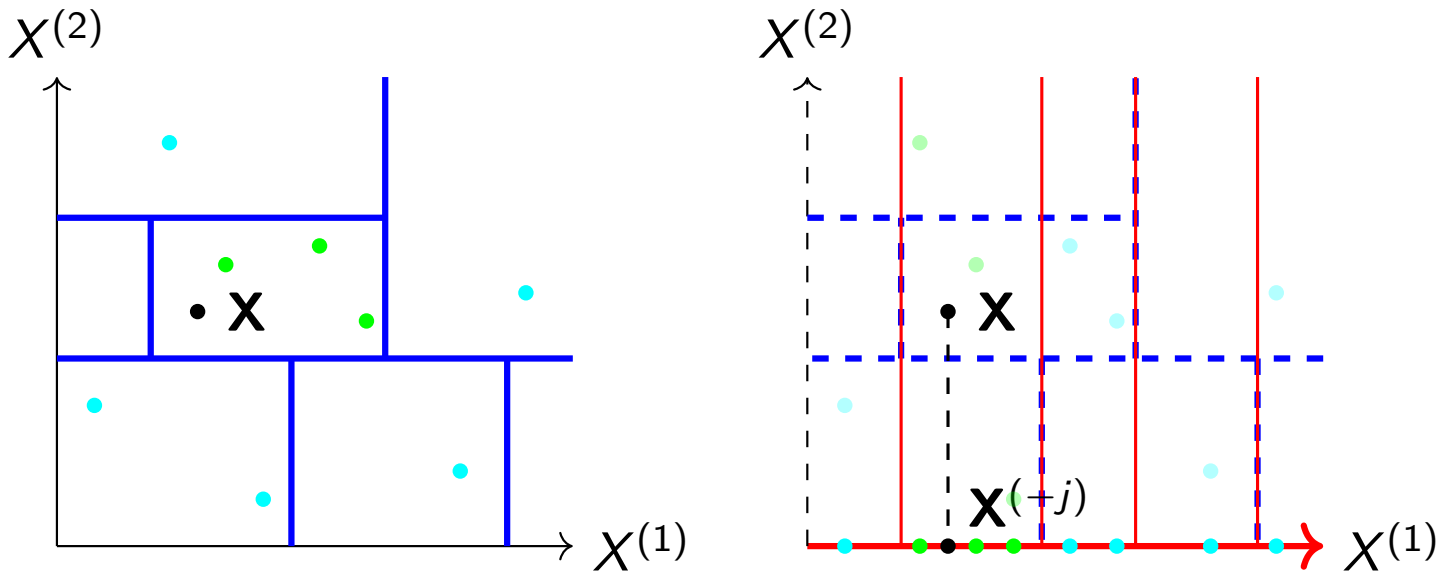


Figure: Partition of $[0,1]^2$ by a random tree (left side) projected on the subspace span by $\mathbf{X}^{(-2)} = X^{(1)}$ (right side), for $p = 2$ and $j = 2$.

$$\widehat{\text{S-MDA}}_{M,n}(X^{(j)}) = \frac{1}{\hat{\sigma}^2_Y} \frac{1}{n} \sum_{i=1}^{n} \left[Y_i - m^{(-j,OOB)}_{M,n}(\mathbf{X}_i^{(-j)}, \mathbf{\Theta}_M) \right]^2 - \left[Y_i - m^{(OOB)}_{M,n}(\mathbf{X}_i, \mathbf{\Theta}_M) \right]^2$$

Consistency of the Sobol-MDA

The Sobol-MDA recovers the appropriate theoretical counterpart for variable selection: the total Sobol index

Theorem (Bénard et al. [2022a])

If Assumptions (A1), (A2'), and (A3') are satisfied, for all $M \in \mathbb{N}^\star$ *and* $j \in \{1, \ldots, p\}$

$$\widehat{S\text{-}MDA}_{M,n}(X^{(j)}) \xrightarrow{p} ST^{(j)}.$$

Consistency of the Sobol-MDA

The Sobol-MDA recovers the appropriate theoretical counterpart for variable selection: the total Sobol index

Theorem (Bénard et al. [2022a])

If Assumptions (A1), (A2'), and (A3') are satisfied, for all $M \in \mathbb{N}^\star$ *and* $j \in \{1, \ldots, p\}$

$$\widehat{S\text{-}MDA}_{M,n}(X^{(j)}) \xrightarrow{p} ST^{(j)}.$$

- Dependent inputs $\mathbf{X}$
- Mild assumption for m (continuous)

Sobol-MDA Experiments

Settings [Archer and Kimes, 2008, Gregorutti et al., 2017]

- $p = 200$ input variables
- 5 independent groups of 40 variables
- each group is a Gaussian vector, strongly correlated

Sobol-MDA Experiments

Settings [Archer and Kimes, 2008, Gregorutti et al., 2017]

- $p = 200$ input variables
- 5 independent groups of 40 variables
- each group is a Gaussian vector, strongly correlated
- 1 variable from each group involved in m

$$m(\mathbf{X}) = 2X^{(1)} + X^{(41)} + X^{(81)} + X^{(121)} + X^{(161)}.$$

- independent Gaussian noise with $\mathbb{V}[\varepsilon] = 10\%\mathbb{V}[Y]$

$$Y = m(\mathbf{X}) + \varepsilon$$

- $n = 1000$ observations
- $M = 300$ trees

Sobol-MDA Experiments

$\widehat{\text{S-MDA}}$	
$\mathbf{X}^{(1)}$	0.035
$\mathbf{X}^{(161)}$	0.005
$\mathbf{X}^{(81)}$	0.004
$\mathbf{X}^{(121)}$	0.004
$\mathbf{X}^{(41)}$	0.002
$\mathbf{X}^{(179)}$	0.002
$\mathbf{X}^{(13)}$	0.001
$\mathbf{X}^{(25)}$	0.001
$\mathbf{X}^{(73)}$	0.001
$\mathbf{X}^{(155)}$	0.001

$\widehat{\text{BC-MDA}}/2\mathbb{V}[Y]$	
$\mathbf{X}^{(1)}$	0.048
$\mathbf{X}^{(25)}$	0.010
$\mathbf{X}^{(31)}$	0.008
$\mathbf{X}^{(14)}$	0.008
$\mathbf{X}^{(40)}$	0.007
$\mathbf{X}^{(3)}$	0.007
$\mathbf{X}^{(17)}$	0.006
$\mathbf{X}^{(26)}$	0.006
$\mathbf{X}^{(41)}$	0.006
$\mathbf{X}^{(121)}$	0.006

$\widehat{\text{IK-MDA}}/\mathbb{V}[Y]$	
$\mathbf{X}^{(1)}$	0.056
$\mathbf{X}^{(5)}$	0.009
$\mathbf{X}^{(81)}$	0.007
$\mathbf{X}^{(41)}$	0.005
$\mathbf{X}^{(161)}$	0.005
$\mathbf{X}^{(15)}$	0.005
$\mathbf{X}^{(121)}$	0.005
$\mathbf{X}^{(7)}$	0.005
$\mathbf{X}^{(4)}$	0.004
$\mathbf{X}^{(28)}$	0.004

Table: Sobol-MDA, normalized BC-MDA, and normalized IK-MDA estimates with influential variables in blue.

Take-home message on MDI and MDA

Do not use MDI or MDA!

We do not know what quantity they are targeting

Alternatives that circumvent some of their flaws have been proposed:

- MDI
  - Out-of-sample estimation [Li et al., 2019, Zhou and Hooker, 2021, Loecher, 2022] with code in python:

    `https://github.com/ZhengzeZhou/unbiased-feature-importance`
- MDA
  - Rerun the model without a given covariate (expensive). Work for any predictive model [Williamson et al., 2021]
  - Use the tree structure to remove a variable from the model without needing to rerun it [Bénard et al., 2022a]

Anyway, remember to check the predictive performance of a model: it it is low, the model is useless and variable importances are misleading.

1. Explainability and random forests

2. Decision rules

3. Variable importance
   - A first variable importance in random forests: MDI
   - A second variable importance in random forests: MDA
   - Shapley values via random forests

Definition of Shapley effects

- Originally defined in game theory [Shapley, 1953]
- Attribute the value produced by a joint team to its individual members

Definition of Shapley effects

- Originally defined in game theory [Shapley, 1953]
- Attribute the value produced by a joint team to its individual members
- Difference of produced value between a subset of the team and the same subteam with an additional member (averaged over all possible subteams).

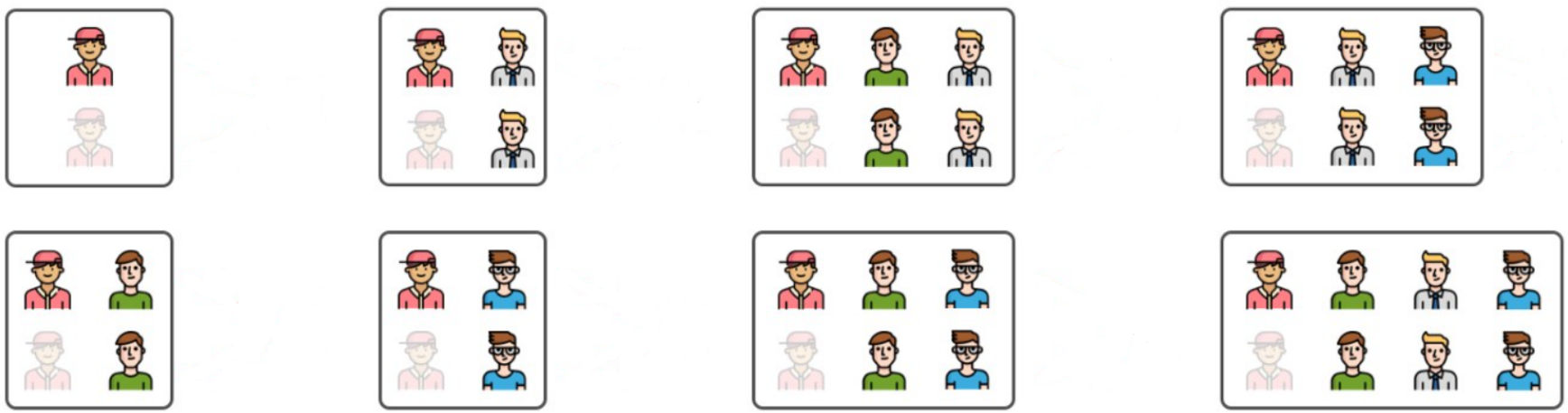


Figure: Illustration of Shapley effects [Lopez, 2021]

Definition of Shapley effects

- Originally defined in game theory [Shapley, 1953]
- Attribute the value produced by a joint team to its individual members
- Difference of produced value between a subset of the team and the same subteam with an additional member (averaged over all possible subteams).

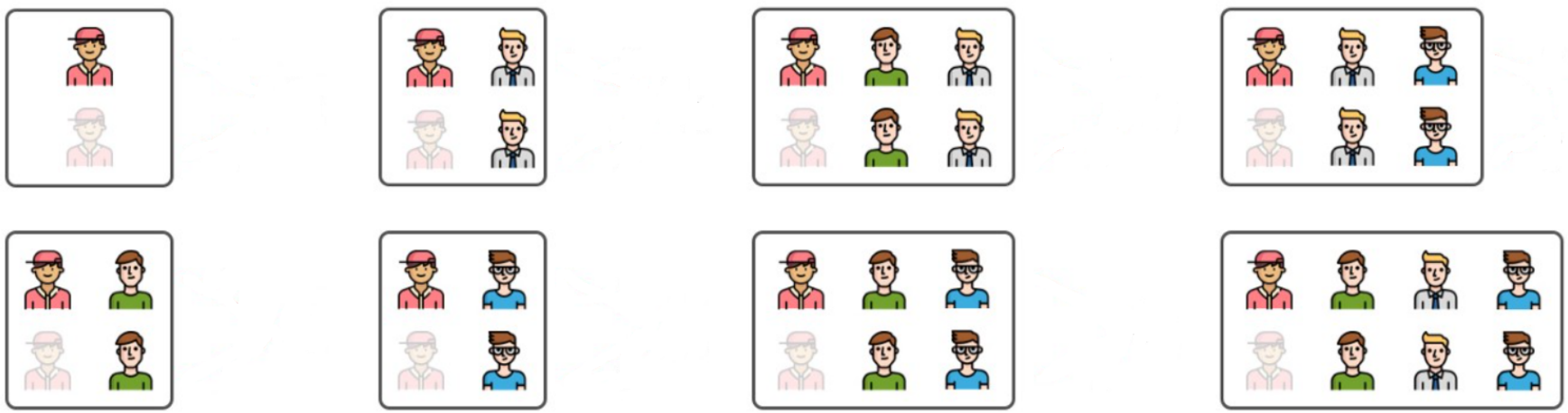


Figure: Illustration of Shapley effects [Lopez, 2021]

Adapted by Owen [2014] to variable importance in machine learning:

- member of the team = input variable
- value function = explained output variance

Definition of Shapley effects

Formally, the Shapley effect of the j-th variable is defined by

$$Sh^{\star}(X^{(j)}) = \sum_{U \subset \{1,\dots,p\} \setminus \{j\}} \frac{1}{p} \binom{p-1}{|U|}^{-1} \frac{\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U \cup \{j\})}]] - \mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]}{\mathbb{V}[Y]}.$$

Definition of Shapley effects

Formally, the Shapley effect of the j-th variable is defined by

$$Sh^\star(X^{(j)}) = \sum_{U \subset \{1,\dots,p\} \setminus \{j\}} \frac{1}{p} \binom{p-1}{|U|}^{-1} \frac{\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U\cup\{j\})}]] - \mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]}{\mathbb{V}[Y]}.$$

Main property: equitably allocate contributions due to dependence and interactions across input variables

Definition of Shapley effects

Formally, the Shapley effect of the j-th variable is defined by

$$Sh^\star(X^{(j)}) = \sum_{U \subset \{1,\dots,p\} \setminus \{j\}} \frac{1}{p} \binom{p-1}{|U|}^{-1} \frac{\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U\cup\{j\})}]] - \mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]}{\mathbb{V}[Y]}.$$

Main property: equitably allocate contributions due to dependence and interactions across input variables

Two obstacles arise to estimate Shapley effects:

1. the computational complexity is exponential with the dimension p

Definition of Shapley effects

Formally, the Shapley effect of the j-th variable is defined by

$$Sh^\star(X^{(j)}) = \sum_{U \subset \{1,\dots,p\} \setminus \{j\}} \frac{1}{p} \binom{p-1}{|U|}^{-1} \frac{\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U\cup\{j\})}]] - \mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]}{\mathbb{V}[Y]}.$$

Main property: equitably allocate contributions due to dependence and interactions across input variables

Two obstacles arise to estimate Shapley effects:

1. the computational complexity is exponential with the dimension p
2. $\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]$ requires a fast and accurate estimate for all variable subsets $U \subset \{1, \dots, p\}$

Definition of Shapley effects

Formally, the Shapley effect of the j-th variable is defined by

$$Sh^\star(X^{(j)}) = \sum_{U \subset \{1,\dots,p\} \setminus \{j\}} \frac{1}{p} \binom{p-1}{|U|}^{-1} \frac{\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U\cup\{j\})}]] - \mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]}{\mathbb{V}[Y]}.$$

Main property: equitably allocate contributions due to dependence and interactions across input variables

Two obstacles arise to estimate Shapley effects:

1. the computational complexity is exponential with the dimension p
   Literature: Monte-Carlo methods
2. $\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]$ requires a fast and accurate estimate for all variable subsets $U \subset \{1, \dots, p\}$

Definition of Shapley effects

Formally, the Shapley effect of the j-th variable is defined by

$$Sh^\star(X^{(j)}) = \sum_{U \subset \{1,\dots,p\} \setminus \{j\}} \frac{1}{p} \binom{p-1}{|U|}^{-1} \frac{\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U\cup\{j\})}]] - \mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]}{\mathbb{V}[Y]}.$$

Main property: equitably allocate contributions due to dependence and interactions across input variables

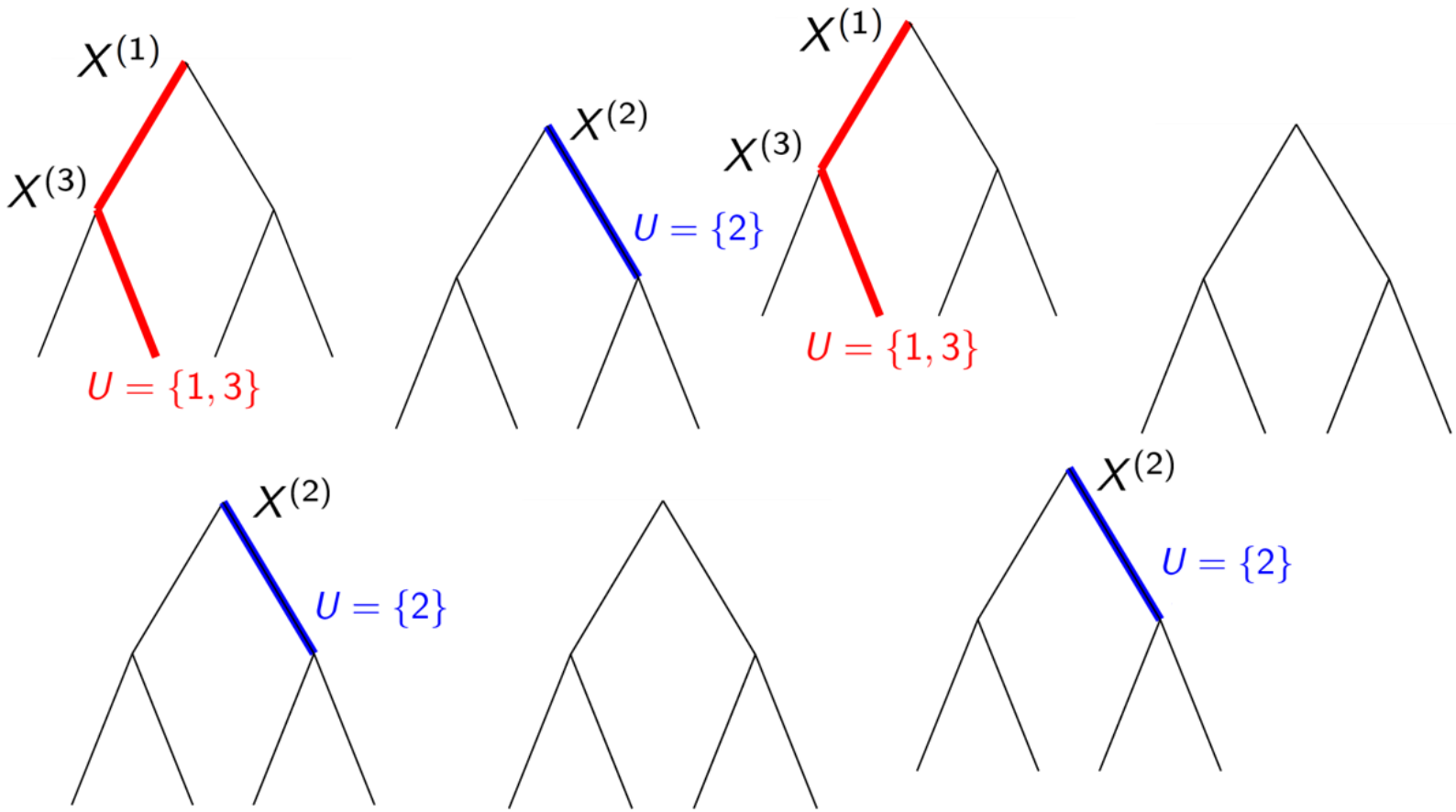
Two obstacles arise to estimate Shapley effects:

1. the computational complexity is exponential with the dimension p
   Literature: Monte-Carlo methods
2. $\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]$ requires a fast and accurate estimate for all variable subsets $U \subset \{1,\dots,p\}$
   Literature: strong approximation of the conditional distributions

SHAFF: SHApley efFects via random Forests

SHAFF proceeds in three steps:

1. sample many subsets U, typically a few hundreds, based on their occurrence frequency $\hat{p}_{M,n}(U)$ in the random forest



SHAFF: SHApley efFects via random Forests

SHAFF proceeds in three steps:

1. sample many subsets U, typically a few hundreds, based on their occurrence frequency $\hat{p}_{M,n}(U)$ in the random forest
2. estimate $\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]$ with the projected forest algorithm for all selected U and their complementary sets $\{1,\dots,p\} \setminus U$: $\hat{v}_{M,n}(U)$

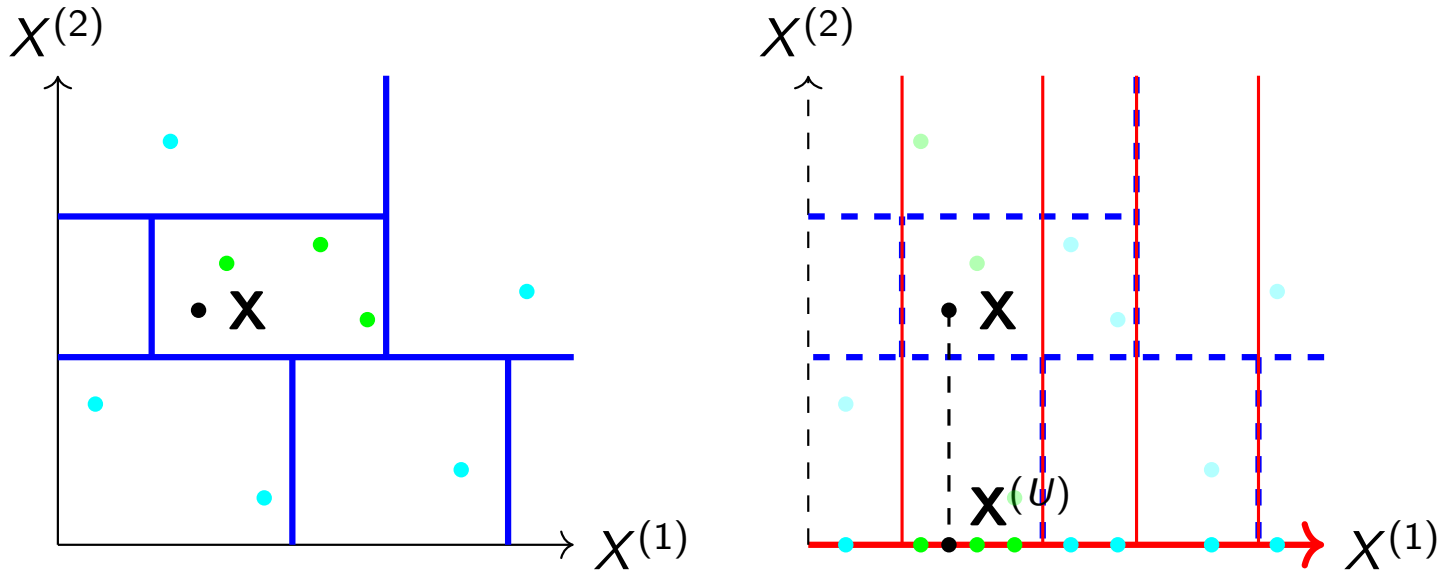


Figure: Partition of $[0,1]^2$ by a random tree (left side) projected on the subspace span by $\mathbf{X}^{(U)} = X^{(1)}$ (right side), for $p = 2$ and $U = \{1\}$.

SHAFF: SHApley efFects via random Forests

SHAFF proceeds in three steps:

1. sample many subsets U, typically a few hundreds, based on their occurrence frequency $\hat{p}_{M,n}(U)$ in the random forest

2. estimate $\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]$ with the projected forest algorithm for all selected U and their complementary sets $\{1,\dots,p\} \setminus U$: $\hat{v}_{M,n}(U)$

3. solve a weighted linear regression problem to recover Shapley effects $\hat{\mathbf{Sh}}_{M_n,n}$ by minimizing in β

$$\ell_{M,n}(\beta) = \frac{1}{K} \sum_{U \in \mathcal{U}_{n,K}} \frac{w(U)}{\hat{p}_{M,n}(U)} (\hat{v}_{M,n}(U) - \beta^T I(U))^2,$$

where $w(U) = \frac{p-1}{\binom{p}{|U|}|U|(p-|U|)}$ and $I(U)$ is the binary vector of dimension p where the j-th component takes the value 1 if $j \in U$ and 0 otherwise.

SHAFF consistency

(A4)

The number of Monte-Carlo sampling K_n and the number of trees M_n grow with n, such that $M_n \longrightarrow \infty$ and $n.M_n/K_n \longrightarrow 0$.

Theorem

If Assumptions (A1), (A2'), (A3'), and (A4) are satisfied, then **SHAFF** *is consistent, that is*

$$\hat{\mathbf{Sh}}_{M_n,n} \xrightarrow{p} \mathbf{Sh}^{\star}.$$

Experiments

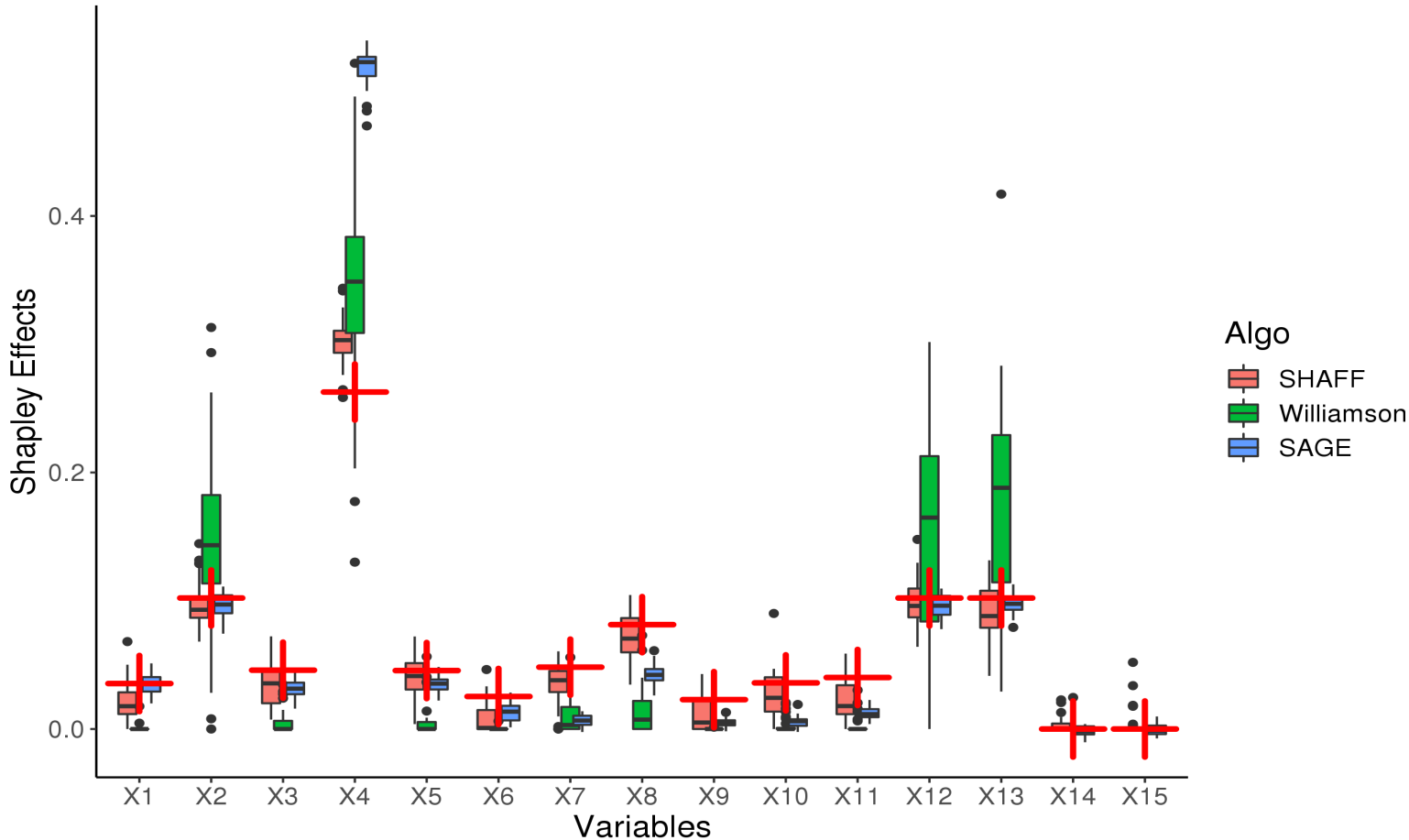


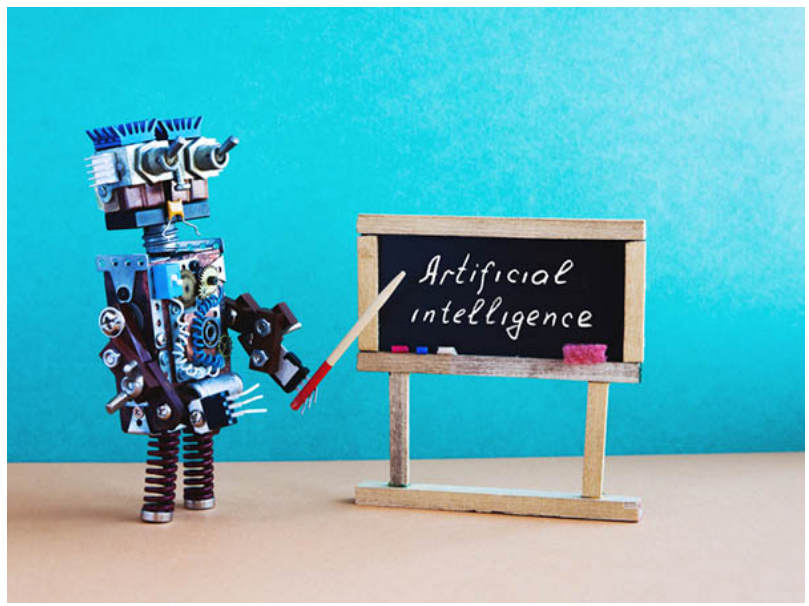
Figure: Shapley effects for a linear case. Red crosses are the theoretical Shapley effects.

Overall take-home messages

If you want to do variable selection via variable importance in random forests:

- Do not use the implemented MDI
  - we do not know toward what quantity it converges, if it converges at all;
- Do not use the implemented MDA
  - it converges to incorrect quantities;
- If you want to build a model with a high accuracy and a small number of variables
  - use a solution that estimates the total Sobol index, as our solution Sobol-MDA;
- If you want to find all variables that are linked to the output
  - use a solution that estimates the Shapley effects, as SHAFF (based on random forests).

Thank you!



References I

K.J. Archer and R.V. Kimes. Empirical characterization of random forest variable importance measures. *Computational Statistics & Data Analysis*, 52:2249–2260, 2008.

S. Arlot and R. Genuer. Analysis of purely random forests bias. *arXiv:1407.3939*, 2014.

C. Bénard, G. Biau, S. Da Veiga, and E. Scornet. Sirus: making random forests interpretable. *arXiv preprint arXiv:1908.06852*, 2019.

C. Bénard, S. Da Veiga, and E. Scornet. Mda for random forests: inconsistency, and a practical solution via the sobol-mda. *accepted for publication in Biometrika*, 2022a.

Clément Bénard, Gérard Biau, Sébastien Da Veiga, and Erwan Scornet. Sirus: Stable and interpretable rule set for classification. *Electronic Journal of Statistics*, 15(1):427–505, 2021a.

Clément Bénard, Gérard Biau, Sébastien Veiga, and Erwan Scornet. Interpretable random forests via rule extraction. In *International Conference on Artificial Intelligence and Statistics*, pages 937–945. PMLR, 2021b.

References II

Clément Bénard, Gérard Biau, Sébastien Da Veiga, and Erwan Scornet. Shaff: Fast and consistent shapley effect estimates via random forests. *AISTAT*, 2022b.

G. Biau. Analysis of a random forests model. *Journal of Machine Learning Research*, 13:1063–1095, 2012.

G. Biau and E. Scornet. A random forest guided tour. *Test*, 25:197–227, 2016.

G. Biau, L. Devroye, and G. Lugosi. Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9:2015–2033, 2008.

A.-L. Boulesteix, A. Bender, J. Lorenzo Bermejo, and C. Strobl. Random forest gini importance favours snps with large minor allele frequency: impact, sources and recommendations. *Briefings in Bioinformatics*, 13:292–304, 2011.

A.-L. Boulesteix, S. Janitza, J. Kruppa, and I.R. König. Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2:493–507, 2012.

L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001a.

L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001b.

References III

L. Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science*, 16:199–231, 2001c.

L. Breiman. Manual on setting up, using, and understanding random forests v3. 1. *Statistics Department University of California Berkeley, CA, USA*, 1:58, 2002.

A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 7:81–227, 2011.

F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning. *arXiv:1702.08608*, 2017.

Jerome H Friedman and Bogdan E Popescu. Predictive learning via rule ensembles. 2008.

R. Genuer. Variance reduction in purely random forests. *Journal of Nonparametric Statistics*, 24(3):543–562, 2012.

B. Gregorutti. *Random forests and variable selection : analysis of the flight data recorders for aviation safety*. PhD thesis, Université Pierre et Marie Curie - Paris VI, 2015.

References IV

B. Gregorutti, B. Michel, and P. Saint-Pierre. Correlation and variable importance in random forests. *Statistics and Computing*, 27:659–678, 2017.

G. Hooker and L. Mentch. Please stop permuting features: an explanation and alternatives. *arXiv preprint arXiv:1905.03151*, 2019.

H. Ishwaran. Variable importance in binary regression trees and forests. *Electronic Journal of Statistics*, 1:519–537, 2007.

J. Klusowski. Sharp analysis of a simple model for random forests. In *AISTAT*, 2021.

J. Klusowski and P. Tian. Large scale prediction with decision trees. *Journal of the American Statistical Association*, 119(545):525–537, 2024.

X. Li, Y. Wang, S. Basu, K. Kumbier, and B. Yu. A debiased mdi feature importance measure for random forests. In *Advances in Neural Information Processing Systems*, pages 8049–8059, New York, 2019.

Z.C. Lipton. The mythos of model interpretability. *arXiv:1606.03490*, 2016.

Markus Loecher. Unbiased variable importance for random forests. *Communications in Statistics-Theory and Methods*, 51(5):1413–1425, 2022.

References V

F. Lopez. Shap: Shapley additive explanations, 2021. URL `https://towardsdatascience.com/shap-shapley-additive-explanations-5a2a271ed9c3`.

G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts. Understanding variable importances in forests of randomized trees. In *Advances in Neural Information Processing Systems*, pages 431–439, 2013.

Nicolai Meinshausen. Node harvest. *The Annals of Applied Statistics*, pages 2049–2072, 2010.

L. Mentch and G. Hooker. Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *Journal of Machine Learning Research*, 17:841–881, 2016.

J. Mourtada, S. Gaiffas, and E. Scornet. Minimax optimal rates for mondrian trees and forests. *The Annals of Statistics*, 2020.

W.J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu. Interpretable machine learning: Definitions, methods, and applications. *arXiv:1901.04592*, 2019.

References VI

K. K. Nicodemus. Letter to the editor: On the stability and ranking of predictors from random forest variable importance measures. *Briefings in bioinformatics*, 12(4):369–373, 2011.

K.K. Nicodemus and J.D. Malley. Predictor correlation impacts machine learning algorithms: implications for genomic studies. *Bioinformatics*, 25:1884–1890, 2009.

A.B. Owen. Sobol'indices and shapley value. *SIAM/ASA Journal on Uncertainty Quantification*, 2:245–251, 2014.

E. Scornet. Trees, forests, and impurity-based variable importance. *Annales de l'IHP*, 2022.

E. Scornet, G. Biau, and J.-P. Vert. Consistency of random forests. *The Annals of Statistics*, 43:1716–1741, 2015.

Erwan Scornet. Random forests and kernel methods. *IEEE Transactions on Information Theory*, 62(3):1485–1500, 2016.

Erwan Scornet and Giles Hooker. Theory of random forests: A review. 2025.

L.S. Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2:307–317, 1953.

References VII

I.M. Sobol. Sensitivity estimates for nonlinear mathematical models. *Mathematical Modelling and Computational Experiments*, 1:407–414, 1993.

C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn. Bias in random forest variable importance measures: illustrations, sources and a solution. *BMC bioinformatics*, 8:25, 2007.

C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis. Conditional variable importance for random forests. *BMC Bioinformatics*, 9:307, 2008.

C. Strobl, T. Hothorn, and A. Zeileis. Party on! 2009.

S. Wager and S. Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113:1228–1242, 2018.

S. Wager and G. Walther. Adaptive concentration of regression trees, with application to random forests. 2015.

Brian D Williamson, Peter B Gilbert, Noah R Simon, and Marco Carone. A general framework for inference on algorithm-agnostic variable importance. *Journal of the American Statistical Association*, pages 1–14, 2021.

B. Yu. Stability. *Bernoulli*, 19:1484–1500, 2013.

References VIII

Z. Zhou and G. Hooker. Unbiased measurement of feature importance in tree-based methods. *arXiv preprint arXiv:1903.05179*, 2019.

Zhengze Zhou and Giles Hooker. Unbiased measurement of feature importance in tree-based methods. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(2):1–21, 2021.

R. Zhu, D. Zeng, and M. R. Kosorok. Reinforcement learning trees. *Journal of the American Statistical Association*, 110:1770–1784, 2015.

SIRUS: **S**table and **I**nterpretable **RU**le **S**et

An example: SIRUS output on Titanic data set [Bénard et al., 2019]

Average survival rate $p_s = 39\%$.

if	`sex` is male	**then**	$p_s = 19\%$	**else**	$p_s = 74\%$
if	1^{st} or 2^{nd} `class`	**then**	$p_s = 56\%$	**else**	$p_s = 24\%$
if	1^{st} or 2^{nd} `class` **&** `sex` is female	**then**	$p_s = 95\%$	**else**	$p_s = 25\%$
if	`fare` < 10.5£	**then**	$p_s = 20\%$	**else**	$p_s = 50\%$
if	no `parents or children aboard`	**then**	$p_s = 35\%$	**else**	$p_s = 51\%$
if	2^{st} or 3^{nd} `class` **&** `sex` is male	**then**	$p_s = 14\%$	**else**	$p_s = 64\%$
if	`sex` is male **&** age $\geqslant 15$	**then**	$p_s = 16\%$	**else**	$p_s = 72\%$

SIRUS

Principle

- Build a random forests and extract all decisions rules from all trees
- Select the rules that appear with a frequence larger than p_0
- Aggregate the rules to obtain the final estimator.

Principle

Frequent paths in random trees = strong and robust patterns in the data.

Technical detail

- Preprocessing: discretize features based on their quantiles
- Random forests: building trees of depth 2

Probability that a Θ-random tree contains a given path $\mathscr{P} \in \Pi$

$$p_n(\mathscr{P}) = \mathbb{P}(\mathscr{P} \in T(\Theta, \mathcal{D}_n) | \mathcal{D}_n)$$

Selected paths

$$\boxed{\hat{\mathscr{P}}_{M,n,p_0} = \{\mathscr{P} \in \Pi : \hat{p}_{M,n}(\mathscr{P}) > p_0\}}$$

where

$$\hat{p}_{M,n}(\mathscr{P}) = \frac{1}{M} \sum_{\ell=1}^{M} \mathbb{1}_{\mathscr{P} \in T(\Theta_\ell, \mathcal{D}_n)}$$

is the Monte-Carlo estimate, directly computed using the random forest with M trees parametrized by $\Theta_1, ..., \Theta_M$.

Stability - definition

Define

- $\mathcal{D}'_n$, Θ' independent copies of $\mathcal{D}_n$ and Θ
- $\hat{p}'_{M,n}(\mathscr{P})$, $\hat{\mathscr{P}}'_{M,n,p_0}$ built with $\mathcal{D}'_n$, Θ'

Dice-Sorensen index

$$\hat{S}_{M,n,p_0} = \frac{2\left|\hat{\mathscr{P}}_{M,n,p_0} \cap \hat{\mathscr{P}}'_{M,n,p_0}\right|}{\left|\hat{\mathscr{P}}_{M,n,p_0}\right| + \left|\hat{\mathscr{P}}'_{M,n,p_0}\right|}.$$

Stability - a theoretical result

(A1) The subsampling rate a_n satisfies $\lim_{n\to\infty} a_n = \infty$ and $\lim_{n\to\infty} \frac{a_n}{n} = 0$.

(A2) The number of trees M_n satisfies $\lim_{n\to\infty} M_n = \infty$.

(A3) $\mathbf{X}$ has a density f with respect to the Lebesgue measure, continuous, bounded, and strictly positive.

Let $\mathcal{U}^\star = \{p^\star(\mathscr{P}), \mathscr{P} \in \Pi\}$ be the set of all theoretical probabilities of appearance of all paths.

Proposition Bénard et al. [2019]

Assume that Assumptions (A1)-(A3) are satisfied. Then, provided $p_0 \in [0,1]\backslash\mathcal{U}^\star$, we have

$$\lim_{n\to\infty} \hat{S}_{M_n,n,p_0} = 1, \quad \text{in probability.}$$