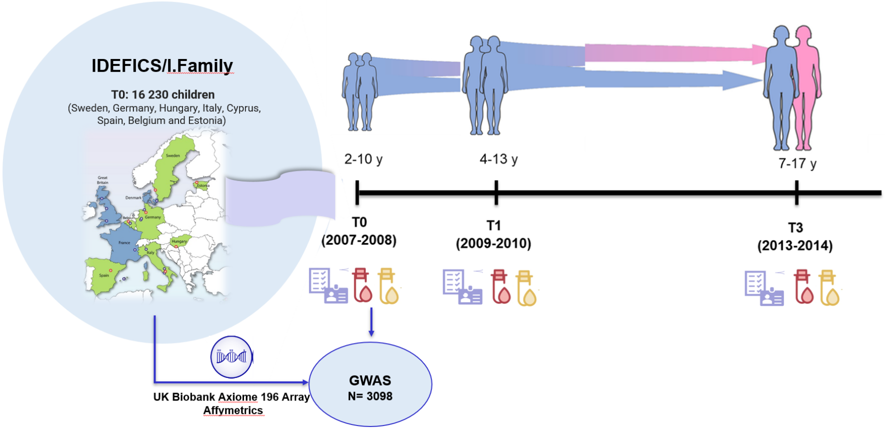
Functional Decomposition of Tree-Based Models

Marvin N. Wright

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

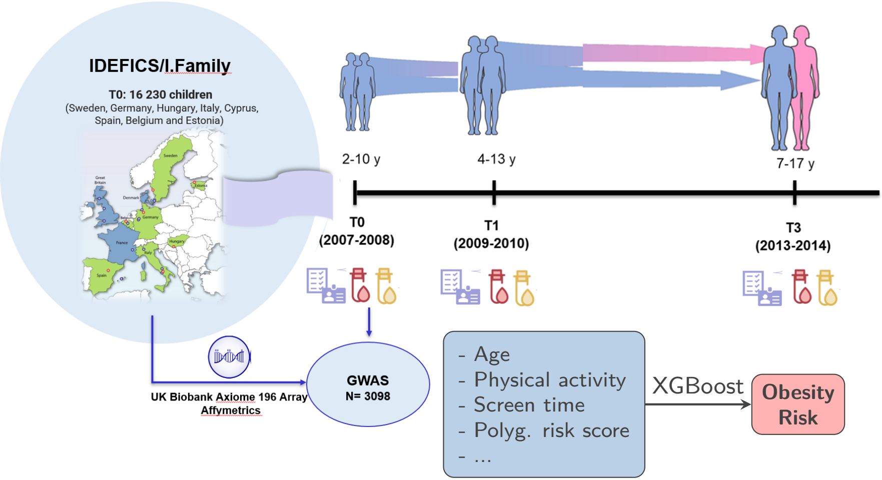
Amsterdam, February 2026

Motivating Example



Ahrens et al. 2011 Int J Obes 35:3 • Ahrens et al. 2017 Int J Epidemiol 46:1394 • Hüls et al. 2021 Int J Obes 45:1321

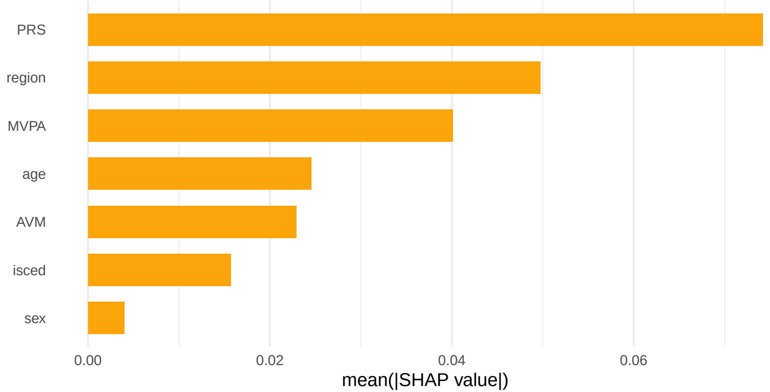
Motivating Example



Motivating Example

SHAP values

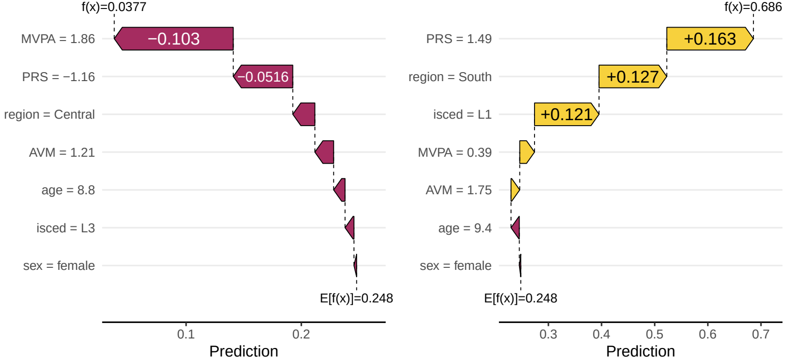
Feature Importance (global)



Motivating Example

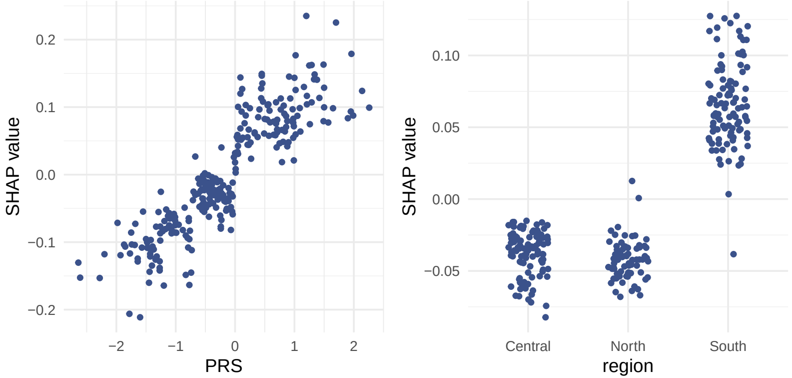
SHAP values

Waterfall Plots (local)



SHAP values

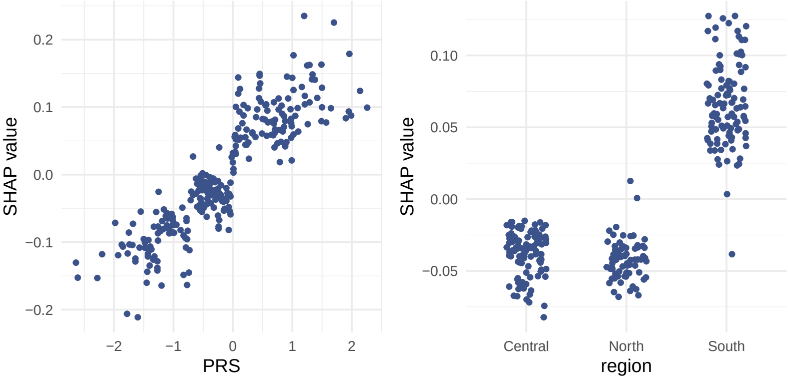
Dependence Plot (global)



Motivating Example

SHAP values

Dependence Plot (global)



What about interactions?

Local vs. Global Explanations

Local explanation

Each subset of features $S \subseteq \{1, \ldots, d\}$ receives a descriptive function $\phi_S : \mathbb{R}^d \to \mathbb{R}$ which may depend on all values of $x \in \mathbb{R}^d$
Example: Shapley values

Global explanation

Each feature $j \in \{1, \ldots, d\}$ receives a single descriptive value $v_j \in \mathbb{R}$ which does not depend on $x \in \mathbb{R}^d$
Example: Feature importance

Local vs. Global Explanations

Local explanation

Each subset of features $S \subseteq \{1, \ldots, d\}$ receives a descriptive function $\phi_S : \mathbb{R}^d \to \mathbb{R}$ which may depend on all values of $x \in \mathbb{R}^d$
Example: Shapley values

Global explanation

Each feature $j \in \{1, \ldots, d\}$ receives a single descriptive value $v_j \in \mathbb{R}$ which does not depend on $x \in \mathbb{R}^d$
Example: Feature importance

Local vs. Global Explanations

Local explanation

Each subset of features $S \subseteq \{1, \ldots, d\}$ receives a descriptive function $\phi_S : \mathbb{R}^d \to \mathbb{R}$ which may depend on all values of $x \in \mathbb{R}^d$
Example: Shapley values

Single-value global explanation

Each feature $j \in \{1, \ldots, d\}$ receives a single descriptive value $v_j \in \mathbb{R}$ which does not depend on $x \in \mathbb{R}^d$
Example: Feature importance

Local vs. Global Explanations

Local explanation

Each subset of features $S \subseteq \{1, \dots, d\}$ receives a descriptive function $\phi_S : \mathbb{R}^d \to \mathbb{R}$ which may depend on all values of $x \in \mathbb{R}^d$

Example: Shapley values

Single-value global explanation

Each feature $j \in \{1, \dots, d\}$ receives a single descriptive value $v_j \in \mathbb{R}$ which does not depend on $x \in \mathbb{R}^d$

Example: Feature importance

(Truly) Global explanation

Each subset of features $S \subseteq \{1, \dots, d\}$ receives a descriptive function $m_S : \mathbb{R}^S \to \mathbb{R}$ which only depends on values $x_S = \{x_k : k \in S\}$ and not on other values $x_{-S} = \{x_j : j \notin S\}$.

Local vs. Global Explanations

Local explanation

Each subset of features $S \subseteq \{1, \ldots, d\}$ receives a descriptive function $\phi_S : \mathbb{R}^d \to \mathbb{R}$ which may depend on all values of $x \in \mathbb{R}^d$

Example: Shapley values

Single-value global explanation

Each feature $j \in \{1, \ldots, d\}$ receives a single descriptive value $v_j \in \mathbb{R}$ which does not depend on $x \in \mathbb{R}^d$

Example: Feature importance

(Truly) Global explanation

Each subset of features $S \subseteq \{1, \ldots, d\}$ receives a descriptive function $m_S : \mathbb{R}^S \to \mathbb{R}$ which only depends on values $x_S = \{x_k : k \in S\}$ and not on other values $x_{-S} = \{x_j : j \notin S\}$.

Local vs. Global Explanations

"A local explanation that explicitly considers all interactions is a global explanation."

Functional Decomposition

Functional decomposition of model $\hat{m}(x)$

$$\begin{aligned}\hat{m}(x) &= \hat{m}_0 + \sum_{k=1}^{d} \hat{m}_k(x_k) + \sum_{k<l} \hat{m}_{kl}(x_k, x_l) + \cdots + \hat{m}_{1,\ldots,d}(x) \\ &= \sum_{S \subseteq \{1,\ldots,d\}} \hat{m}_S(x_S).\end{aligned}$$

Functional Decomposition

Functional decomposition of model $\hat{m}(x)$

$$\begin{aligned}\hat{m}(x) &= \hat{m}_0 + \sum_{k=1}^{d} \hat{m}_k(x_k) + \sum_{k<l} \hat{m}_{kl}(x_k, x_l) + \cdots + \hat{m}_{1,\ldots,d}(x) \\ &= \sum_{S \subseteq \{1,\ldots,d\}} \hat{m}_S(x_S).\end{aligned}$$

Huge number of components

$2^7 = 128$ for the example above

$2^{20} = 1,048,576$ for 20 features

...

Functional Decomposition

Functional decomposition of model $\hat{m}(x)$

$$\hat{m}(x) = \hat{m}_0 + \sum_{k=1}^{d} \hat{m}_k(x_k) + \sum_{k<l} \hat{m}_{kl}(x_k, x_l) + \cdots + \hat{m}_{1,\ldots,d}(x)$$
$$= \sum_{S \subseteq \{1,\ldots,d\}} \hat{m}_S(x_S).$$

Gradient-boosted trees

Ensemble of low-dimensional structures

$q << d$ with $q = \max\{|S| : m_S \neq 0\}$

Functional Decomposition

Functional decomposition of model $\hat{m}(x)$

$$\begin{aligned}\hat{m}(x) &= \hat{m}_0 + \sum_{k=1}^{d} \hat{m}_k(x_k) + \sum_{k<l} \hat{m}_{kl}(x_k, x_l) + \cdots + \hat{m}_{1,\ldots,d}(x) \\ &= \sum_{S \subseteq \{1,\ldots,d\}} \hat{m}_S(x_S).\end{aligned}$$

Right-hand side not identified

Possible to change components on the right without altering the left-hand side

Functional Decomposition

Functional decomposition of model $\hat{m}(x)$

$$\begin{aligned}\hat{m}(x) &= \hat{m}_0 + \sum_{k=1}^{d} \hat{m}_k(x_k) + \sum_{k<l} \hat{m}_{kl}(x_k, x_l) + \cdots + \hat{m}_{1,\ldots,d}(x) \\ &= \sum_{S \subseteq \{1,\ldots,d\}} \hat{m}_S(x_S).\end{aligned}$$

Different identifications proposed

Baseline,
Marginal (aka interventional),
Conditional (aka observational)

Connection to SHAP values

SHAP values

$$\hat{m}(x_0) = \phi_0 + \sum_{k=1}^{d} \phi_k(x_0)$$

with $x_0 \in \mathbb{R}^d$ and constants $\phi_0, \phi_1(x_0), \ldots, \phi_d(x_0)$

Connection to SHAP values

SHAP values

$$\hat{m}(x_0) = \phi_0 + \sum_{k=1}^{d} \phi_k(x_0)$$

with $x_0 \in \mathbb{R}^d$ and constants $\phi_0, \phi_1(x_0), \dots, \phi_d(x_0)$

Right-hand side not identified

Different identifications lead to different SHAP values with different value functions, e.g. interventional SHAP or observational SHAP

Connection to SHAP values

SHAP values

$$\hat{m}(x_0) = \phi_0 + \sum_{k=1}^{d} \phi_k(x_0)$$

with $x_0 \in \mathbb{R}^d$ and constants $\phi_0, \phi_1(x_0), \ldots, \phi_d(x_0)$

Where are the interactions?

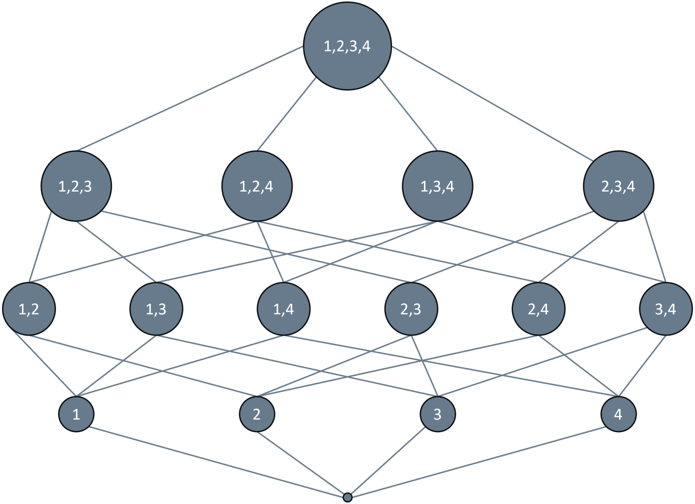
Connection to SHAP values

SHAP values can be calculated from decomposition

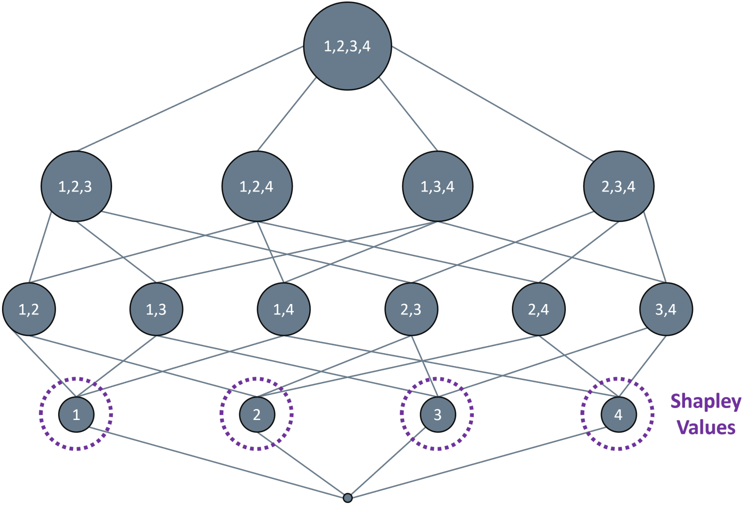
SHAP values are weighted averages of the corresponding components, where an interaction component is equally split to all involved features

$$\phi_k(x) = \hat{m}^*_k(x_k) + \frac{1}{2}\sum_j \hat{m}^*_{kj}(x_{kj}) + \cdots + \frac{1}{d}\hat{m}^*_{1,\dots,d}(x_{1,\dots,d})$$

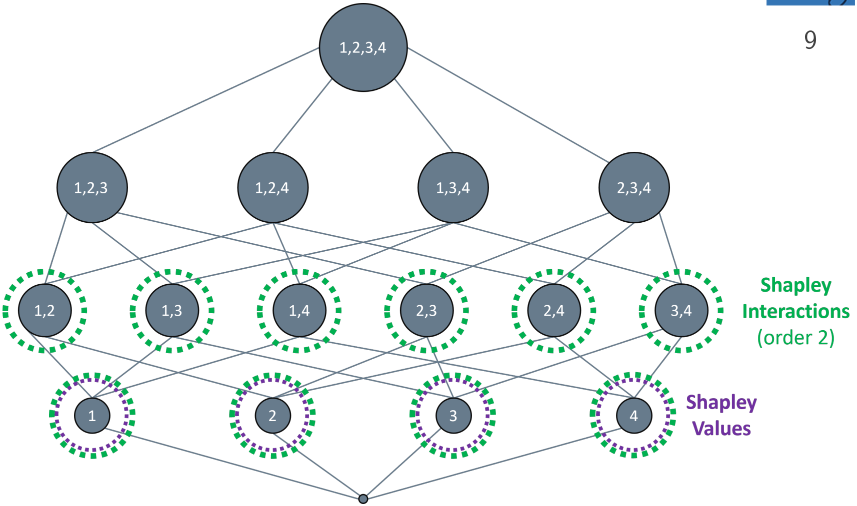
Connection to SHAP values



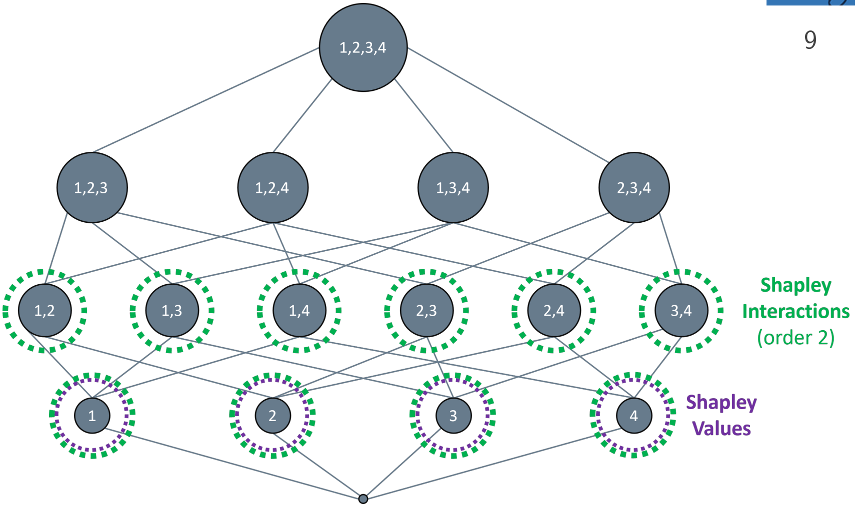
Connection to SHAP values



Connection to SHAP values



Connection to SHAP values



d-interaction SHAP is the full functional decomposition

Connection to Partial Dependence

Partial dependence function

$$\xi_S = \sum_{U \subseteq S} \hat{m}^*_U$$

where ξ_S is be the partial dependence plot for a set of features S.

Single feature $S = \{k\}$

$$\xi_k(x_k) = \hat{m}^*_0 + \hat{m}^*_k(x_k)$$

Connection to Partial Dependence

Partial dependence function

$$\xi_S = \sum_{U \subseteq S} \hat{m}^*_U$$

where ξ_S is be the partial dependence plot for a set of features S.

Single feature $S = \{k\}$

$$\xi_k(x_k) = \hat{m}^*_0 + \hat{m}^*_k(x_k)$$

Interventional SHAP

SHAP with PD value function = interventional SHAP

TreeSHAP algorithms

Algorithm for (gradient-boosted) trees

Decomposition can be calculated from tree structures with a single recursion through each tree

TreeSHAP algorithm

- Leaf: Return leaf prediction
- Internal node:
  - If split feature in subset U: Apply splitting criterion of node and continue with respective child node
  - If not: Continue in both the left and right children nodes, each weighted by the coverage

TreeSHAP algorithms

Method	FD	Consistent	Complexity
VanillaPD	✓	✓	$O(R2^d n_\mathtt{e} n_\mathtt{b})$
Friedman (2001)	✓	✗	$O(2^d 2^D n_e)$
TreeSHAP-path	✗	✗	$O(D^2 2^D n_\mathtt{e})$
TreeSHAP-int	✗	✓	$O(D2^D n_\mathtt{e} n_\mathtt{b})$
Zern (2023)	✗	✓	$O(2^D n_b + 3^D D n_e)$
FastPD	✓	✓	$O(2^{D+F}(n_\mathtt{e} + n_\mathtt{b}))$

n_e: evaluation samples, n_b: background samples, d: number of features, D: depth of tree, F: number of features the tree splits on, R: model evaluations

TreeSHAP algorithms

Method	FD	Consistent	Complexity
`VanillaPD`	✓	✓	$O(R2^d n_\mathbf{e} n_\mathbf{b})$ Slow
Friedman (2001)	✓	✗	$O(2^d 2^D n_e)$ Fast
`TreeSHAP-path`	✗	✗	$O(D^2 2^D n_\mathbf{e})$ Fast
`TreeSHAP-int`	✗	✓	$O(D2^D n_\mathbf{e} n_\mathbf{b})$ Slow
Zern (2023)	✗	✓	$O(2^D n_b + 3^D D n_e)$ Fast
`FastPD`	✓	✓	$O(2^{D+F}(n_\mathbf{e} + n_\mathbf{b}))$ Fast

n_e: evaluation samples, n_b: background samples, d: number of features, D: depth of tree, F: number of features the tree splits on, R: model evaluations

TreeSHAP algorithms

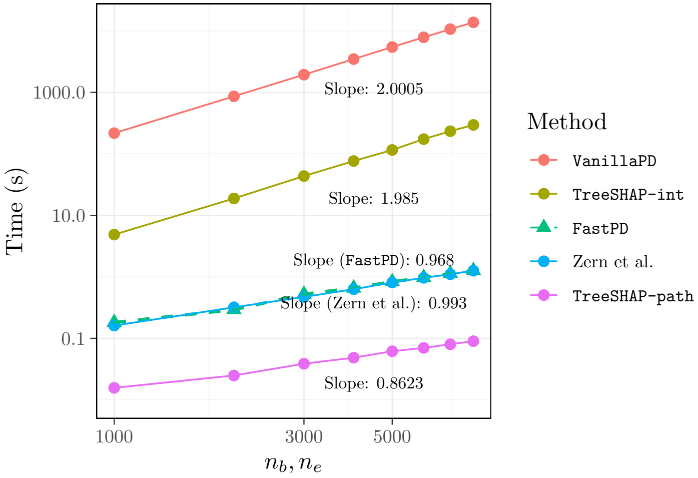
Method	FD	Consistent	Complexity
`VanillaPD`	✓	✓	$O(R2^d n_\mathtt{e} n_\mathtt{b})$ Slow
Friedman (2001)	✓	✗	$O(2^d 2^D n_e)$ Fast
`TreeSHAP-path`	✗	✗	$O(D^2 2^D n_\mathtt{e})$ Fast
`TreeSHAP-int`	✗	✓	$O(D2^D n_\mathtt{e} n_\mathtt{b})$ Slow
Zern (2023)	✗	✓	$O(2^D n_b + 3^D D n_e)$ Fast
`FastPD`	✓	✓	$O(2^{D+F}(n_\mathtt{e} + n_\mathtt{b}))$ Fast

n_e: evaluation samples, n_b: background samples, d: number of features, D: depth of tree, F: number of features the tree splits on, R: model evaluations

FastPD the only algorithm that provides a full functional decomposition, is consistent and fast

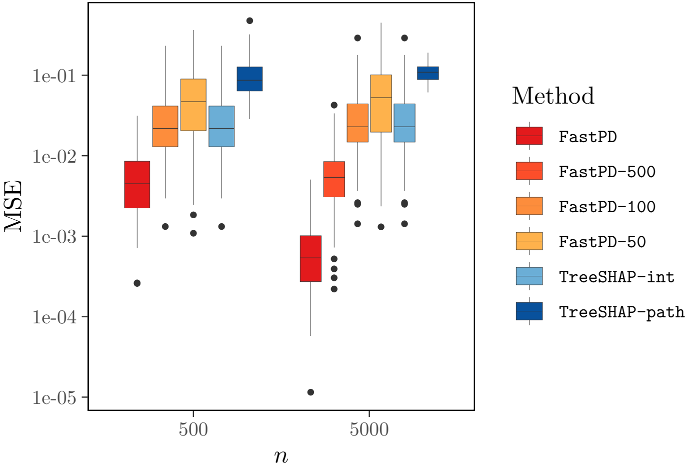
TreeSHAP algorithms

Runtime



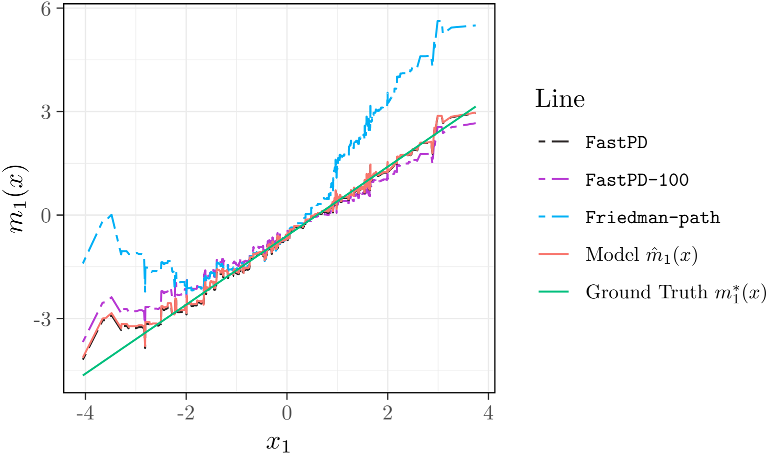
TreeSHAP algorithms

Consistency



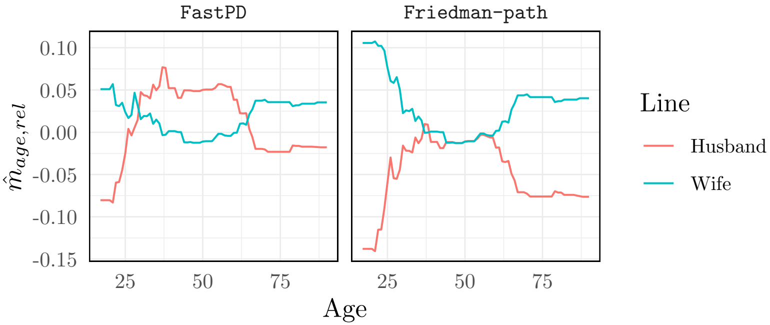
TreeSHAP algorithms

Consistency – Simulation



TreeSHAP algorithms

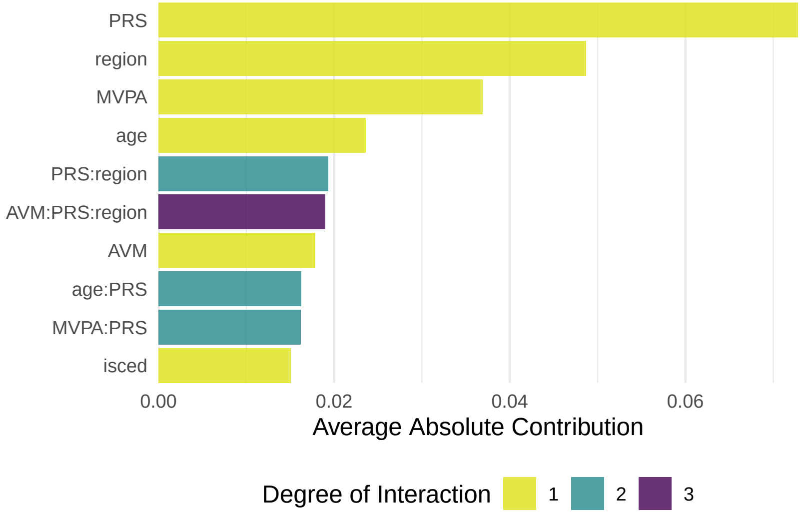
Consistency – Example



Example from the `adult` dataset

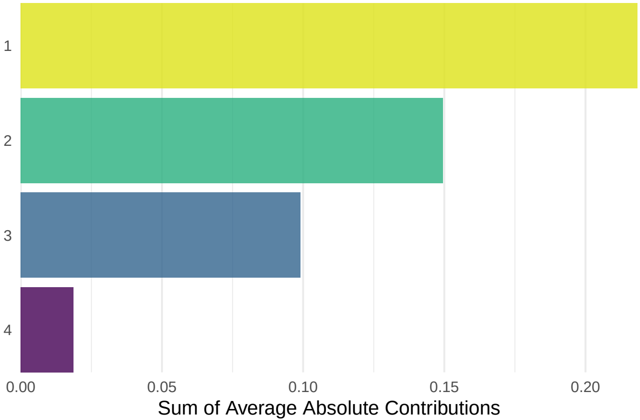
Application

Feature importance (single-value global)

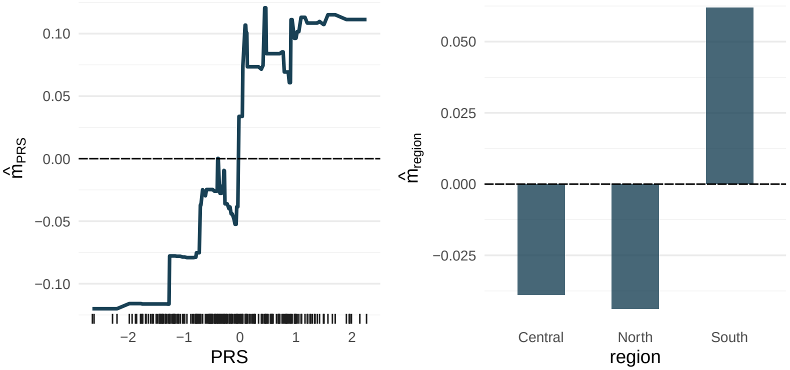


Application

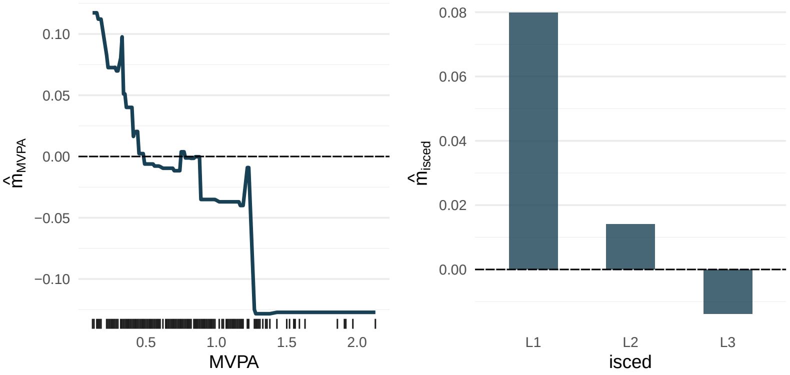
Feature importance (single-value global), by degree



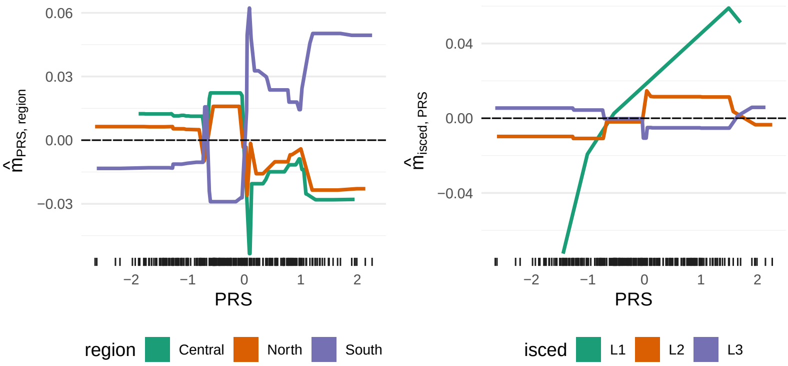
Decomposition - main effects (global)



Decomposition - main effects (global)

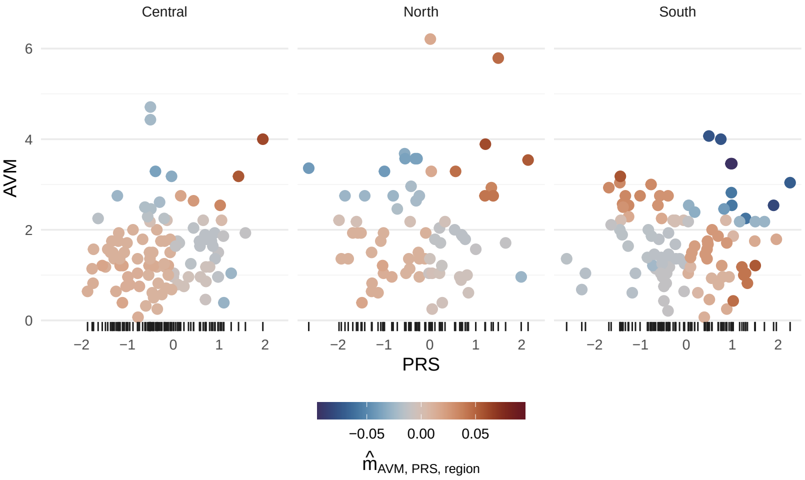


Decomposition - 2-way interactions (global)



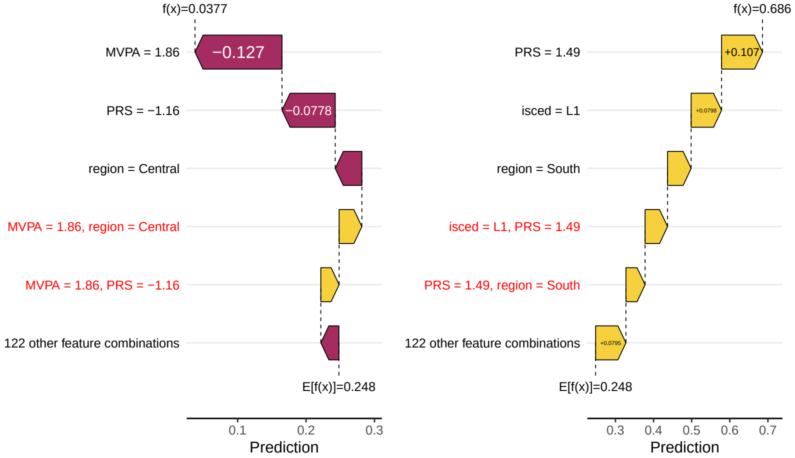
Application

Decomposition - 3-way interactions (global)



Application

Waterfall Plots (local)



Conclusion & Outlook

Conclusion

- Proposed a marginal identification that connects functional decomposition, SHAP values and partial dependence
- Global explanation by functional decomposition
- New perspective on Shapley values without game theory
- Fast algorithm and implementations for gradient-boosted trees (e.g. XGBoost), random forest and random planted forest

Outlook

- Further application: Post-hoc feature removal
  $\rightarrow$ Plug-in debiasing
- Implementations for other learning algorithms
- Implementation in SHAP-IQ

Literature & Software

Literature

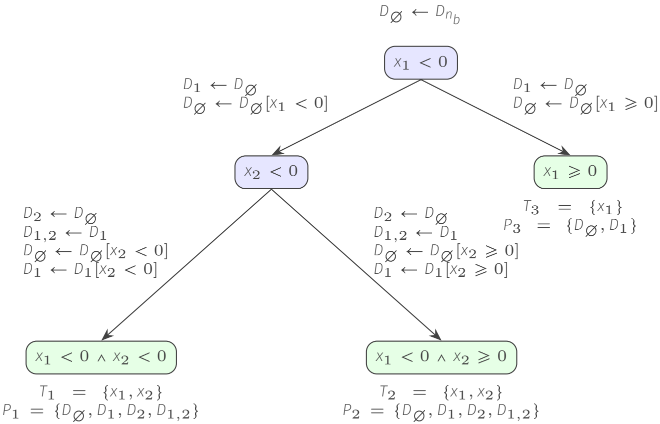
- Hiabu M, Meyer JT & Wright MN (2023). Unifying local and global model explanations by functional decomposition of low dimensional structures. AISTATS 2023. https://proceedings.mlr.press/v206/hiabu23a.html.
- Liu J, Steensgaard T, Wright MN, Pfister N & Hiabu M (2025). Fast estimation of partial dependence functions using trees. ICML 2025. https://proceedings.mlr.press/v267/liu25bm.html.

Software

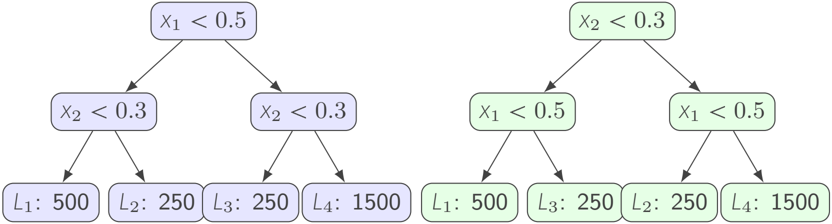
- R (+Rcpp): https://github.com/PlantedML/glex
- Python (+Rust): https://github.com/jyliuu/glex-rust (experimental)

Backup slides

FastPD algorithm



Inconsistency of TreeSHAP-path



The two trees have the same leaves hence predict the same values, but their explanations differ when obtained via `TreeSHAP-path`. The number on each leaf is the number of observations landing in that leaf.

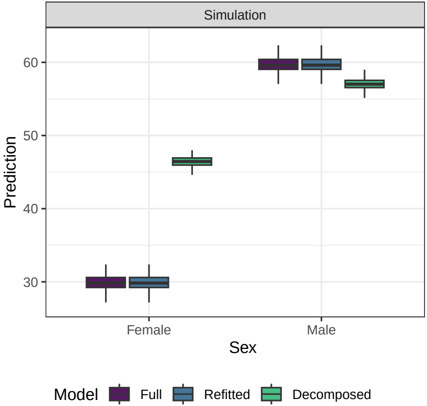
Application: Post-hoc Feature Removal

Simulation

Predict a person's salary, based on sex and weekly working hours.

Simulation: Average of 40 hours for men and 30 hours for women

$y = 20 \cdot x_{\text{sex}} + 1 \cdot x_{\text{working hours}} + \mathcal{N}(0, 1)$



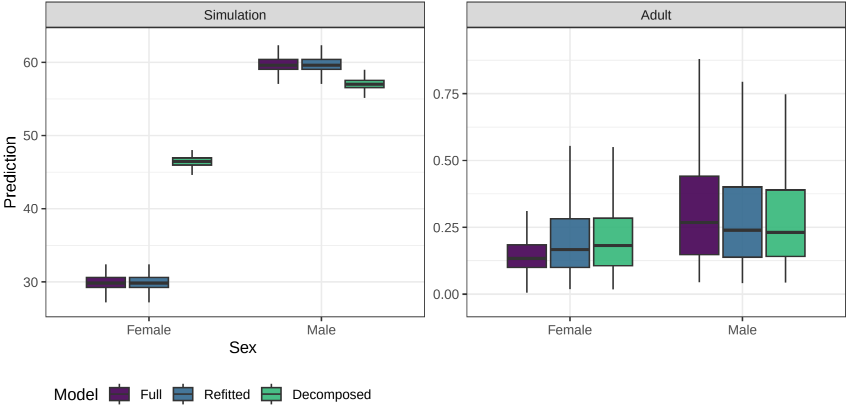
Application: Post-hoc Feature Removal

Simulation

Predict a person's salary, based on sex and weekly working hours.

Simulation: Average of 40 hours for men and 30 hours for women

$y = 20 \cdot x_{\text{sex}} + 1 \cdot x_{\text{working hours}} + \mathcal{N}(0, 1)$



Motivating Example

Predict childhood obesity

- Data from the IDEFICS/I.Family cohort
- Children from 7 European countries
- Aged 2-9 years at baseline
- Sample size: 828 (552 train, 276 test)
- Predict overweight/obesity after 4 years
- XGBoost, AUC 0.93 (train), 0.65 (test)

Features

- Age
- Sex
- European region
- Parental education
- Physical activity (MVPA)
- Screentime (AVM)
- Polygenic risk score (PRS)

Ahrens et al. 2011 Int J Obes 35:3 • Ahrens et al. 2017 Int J Epidemiol 46:1394 • Hüls et al. 2021 Int J Obes 45:1321