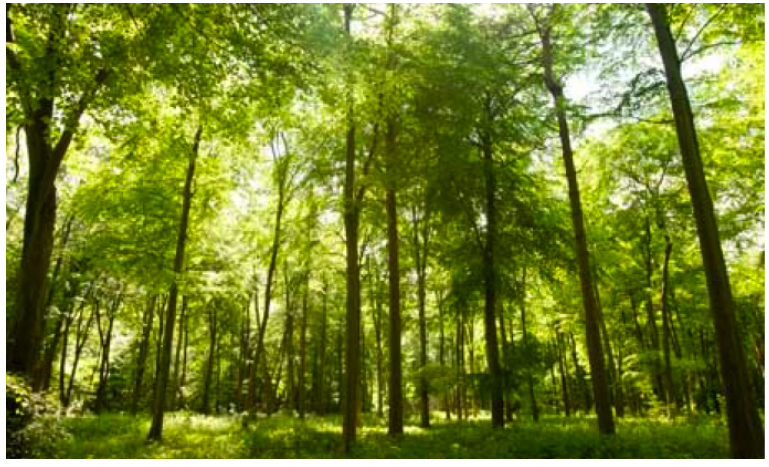
Early Stopping for Regression Trees

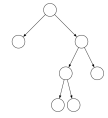
Ratmir Miftachov
Markus Reiß

Seminar Mathematische Statistik
WIAS Berlin
April 30, 2025

Classification and Regression Trees (CART)

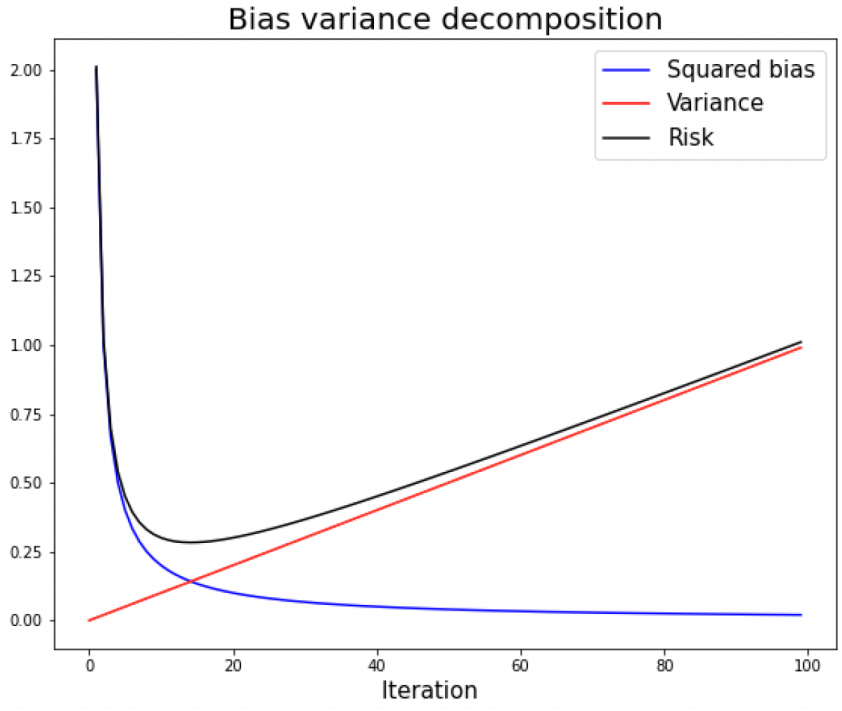
- Introduced by Breiman et al. (1984); nowadays widely popular in data science within industry and academia
- Most Kaggle winning solutions of 2024 on tabular data are tree-based models
- Foundation of enhanced algorithms, like bagging, random forest, and boosted decision trees (AdaBoost, XGBoost); see Friedman (2001, AoS)

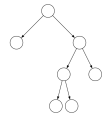




Early stopping

- Find a fully data-driven iteration that neither over- nor underfits the data
- Can we handle computational and statistical complexity at the same time?



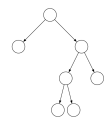


When to stop an iterative algorithm?

- Iterative procedures are essential to machine learning
  - training of neural networks
  - L2 boosting
  - gradient descent with mini-batching
- Procedures should not be run indefinitely, as this may lead to overfitting and high computational costs
- **Solution:** *stop* iterative methods *early* in a data-driven way

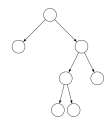
$$\text{stop} = \inf\{\text{iter} \in \{0, \ldots, \text{iteration}_{\text{max}}\} \mid \text{MSE}(\hat{f}_{\text{iter}}) \leq \text{threshold}\},$$

where $\hat{f}_{\text{iter}}$ is an estimator at iteration *iter* for function f and $\text{MSE}(\bullet)$ is the empirical mean squared error on the training set.



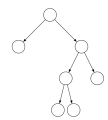
Some related literature

- A. Nobel (1996): Histogram regression estimation using data-dependent partitions, AoS
- S. Gey, E. Nedelec (2005): Model selection for CART regression trees, IEEE IT
- E. Scornet, G. Biau, JP. Vert (2015): Consistency of random forests, AoS
- J. Klusowski, P. Tian (2024): Large-scale prediction with decision trees, JASA
- B. Stankewitz (2024): Early stopping for L2-boosting, AoS



Contribution of this paper

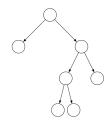
- Introduce the *generalised projection flow* as a unifying framework for smoothing estimators like ridge regression, smoothing splines, CART
- Oracle inequalities for the early stopped CART algorithm
- Ready-to-use approach for practitioners and implementation in a Python library



Nonparametric regression

$$Y_i = f(X_i) + \varepsilon_i, \quad i = 1, \ldots, n,$$
$$\text{with } \mathrm{E}[\varepsilon_i \,|\, X_i] = 0, \ \mathrm{Var}(\varepsilon_i) = \sigma^2$$

- $(X_{i,1}, \ldots, X_{i,d})^\top \in \mathcal{X} \subseteq \mathbb{R}^d$ and response $Y_i \in \mathcal{Y} \subseteq \mathbb{R}$
- We have i.i.d. observations $(X_1, Y_1), \ldots, (X_n, Y_n)$
- Conditional mean is $f(x) = \mathrm{E}(Y|X = x)$
- The squared empirical norm is $\|f\|_n^2 := n^{-1} \sum_{i=1}^n f(X_i)^2$

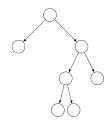


CART splitting

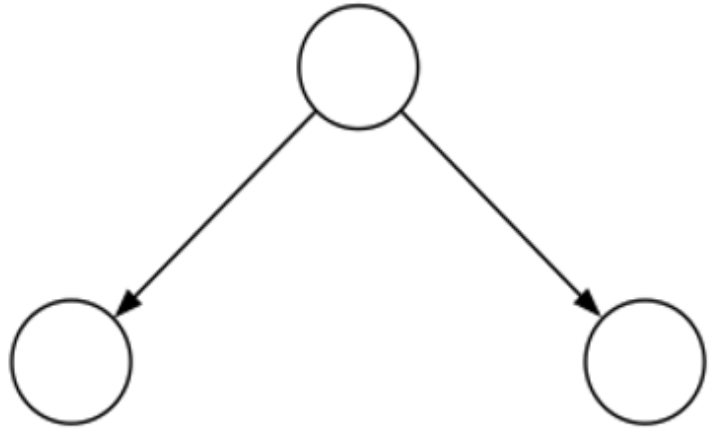
- Let A be a generic (parent) node with child nodes $A_L(j, c) = \{X \in A : X_j < c\}$ and $A_R(j, c) = \{X \in A : X_j \geq c\}$
- Choose coordinate and threshold (j^*, c^*) greedily for A to minimize training error of child nodes

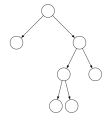
$$(j^*, c^*) \in \underset{(j,c)}{\operatorname{argmin}} \left(\sum_{i:X_i \in A_L} \left(Y_i - \bar{Y}_{A_L}\right)^2 + \sum_{i:X_i \in A_R} \left(Y_i - \bar{Y}_{A_R}\right)^2 \right)$$

with $\bar{Y}_A = \frac{1}{n_A} \sum_{i:X_i \in A} Y_i$ being the node average with local sample size $n_A = |\{i : X_i \in A\}|$.

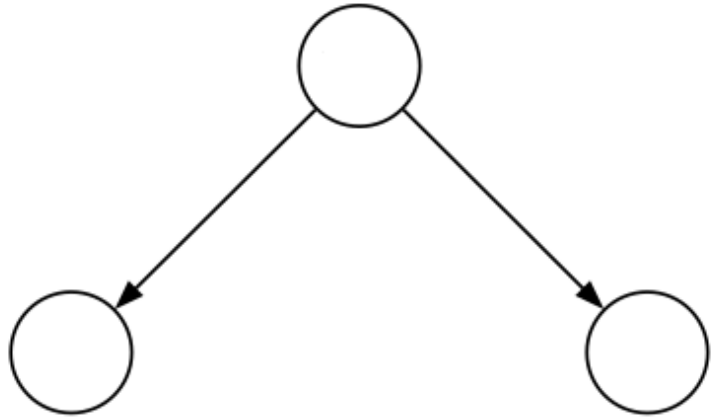


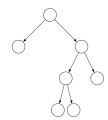
Global stopping rule



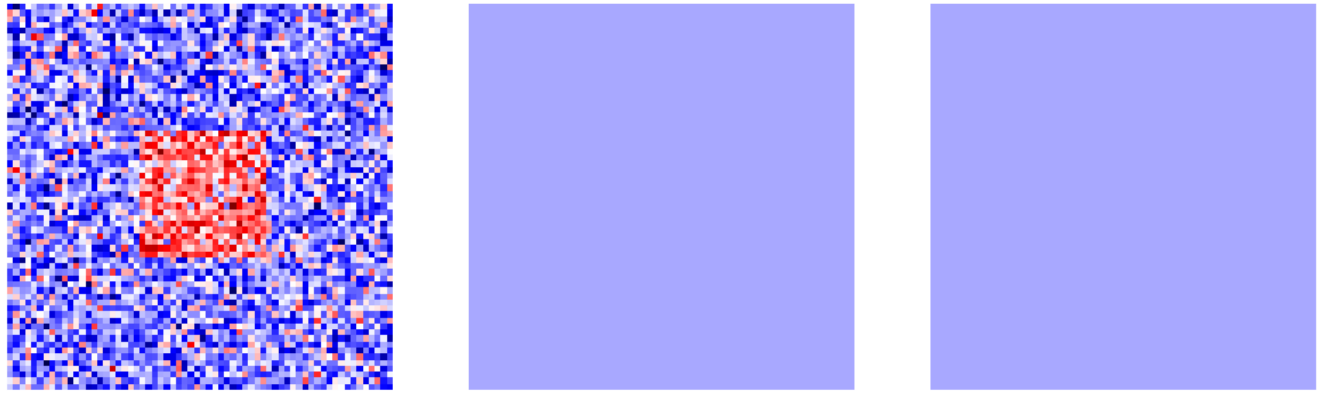


Semi-global stopping rule





Stopping tree growth early

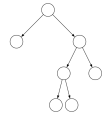


Global early stopping:

breadth-first search & global stopping criterion

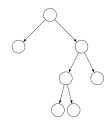
Semi-global early stopping:

best-first search & global stopping criterion



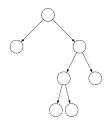
State of the art: Post-pruning

- *Post-pruning* (bottom-up approach):
  - Goal: Find a subtree minimising penalized least squares (penalty $\lambda \times \#P_{\text{subtree}}$).
  - Start with a fully grown tree.
  - Iteratively delete nodes (prune) to minimise the criterion.
  - Use CV to choose among different penalties.
  - Drawback: computationally intensive, not much theory.
- Grow a tree to its maximum depth.
- *Pre-pruning* (top-down approach):
  - limit number of observations in terminal nodes (Breiman et al., 1984).
  - use a pre-specified depth (Klusowski & Tian, 2024).



Data-driven sequence of partitions

- $P_1 = \mathbb{R}^d$ and given a partition P_m at iteration m split any $A_m \in P_m$ into two subsets (children) $A_m = A_{m,L} \cup A_{m,R}$ to obtain a partition P_{m+1} with $\#P_{m+1} \leqslant 2\#P_m$.
- The terminal nodes (nodes without children) define a partition of $\mathbb{R}^d$, which is refined as the tree grows.
- Iterate until the partition contains only sets A_m with $n_m = 1$.



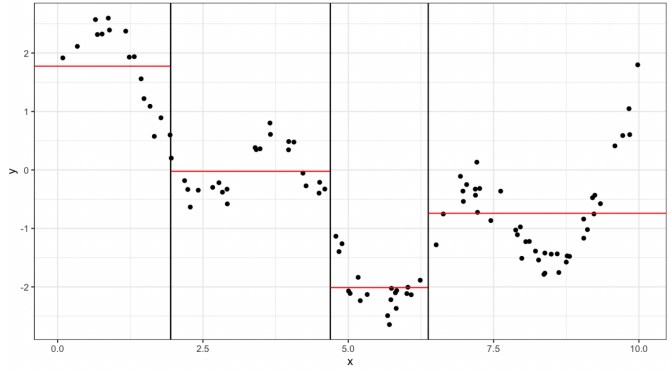
Refining orthogonal projections

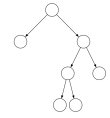
- Partition $P_m = \{A_1^m, \ldots, A_K^m\}$ of $\mathbb{R}^d$ at iteration m generates orthogonal projection $\Pi_{P_m} : L_n^2 \to L_n^2$

$$\Pi_m f(x) = \frac{1}{n_k^m} \sum_{i:X_i \in A_k^m} f(X_i) \quad \text{for } x \in A_k^m.$$

- Formulate the CART-estimator at iteration m as

$$\hat{f}_m(x) := \Pi_m Y := \frac{1}{n_k^m} \sum_{i:X_i \in A_k^m} Y_i \qquad \text{for } x \in A_k^m.$$





Generalized projection flow

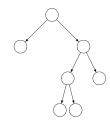
- In practice, the global estimator can overshoot between two generations. Interpolation balances over- and underfitting more granularly.
- Linear interpolation yields projection flow $(\Pi_t)_{t\in[0,n]}$:

$$\Pi_t := (1-\alpha)\Pi_{P_m}+\alpha\Pi_{P_{m+1}}, \quad \text{for } t = m+\alpha \quad \text{and} \quad \alpha \in [0,1].$$

- Estimator and residual norm at $t \in [0,n]$ are given as:

$$\hat{f}_t = \Pi_t Y, \quad R_t^2 = \|Y - \hat{f}_t\|_n^2 = \|(\mathbb{I}-\Pi_t)Y\|_n^2$$

- Computationally, the interpolation comes at no additional cost!



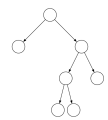
Generalized projection flow

- Further examples of the generalized projection flow are: Gradient descent, ridge regression, L^2-boosting,...

Lemma (error decomposition)

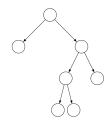
$$\left\| \hat{f}_t - f \right\|_n^2 \leqslant 2 \left\| (\mathrm{Id} - \Pi_t) f \right\|_n^2 + 2 \left\| \Pi_t \varepsilon \right\|_n^2 =: 2A_t + 2S_t$$

- $t \mapsto A_t$ decreases continuously from $\|f\|_n^2$ to 0 .
- $t \mapsto S_t$ increases continuously from 0 to $\|\varepsilon\|_n^2$.



Example: Ridge regression

- Ridge estimator is $\hat{F}_\lambda = S_\lambda Y$ with the smoother matrix $S_\lambda = X(X^\top X + \lambda \mathbb{I})^{-1} X^\top$ and ridge parameter $\lambda \geqslant 0$
- Setting $\Pi_t = S_{\lambda(t)}$ for the inverse function $\lambda(t)$ of $t(\lambda)$ we obtain a regularization flow
- Goal of early stopping: choose the penalisation parameter λ sequentially, starting from large values of λ and decreasing it gradually



Balanced oracle and early stopping

Definition

The (random) balanced oracle τ_b is given by

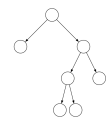
$$\tau_b = \inf\{t \in [0, n] \,|\, A_t \leqslant S_t\}$$

and for a threshold $\kappa > 0$ the data-driven stopping rule τ is given by

$$\tau = \inf\{t \in [0, n] \,|\, R_t^2 \leqslant \kappa\}.$$

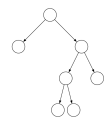
Proposition (adaptation error)

$$\|\hat{f}_\tau - \hat{f}_{\tau_b}\|_n^2 \leqslant \underbrace{|\kappa - \|\varepsilon\|_n^2|}_{\textit{early stop error}} + \underbrace{2\langle(\Pi_{\tau_b} - \Pi_{\tau_b}^2)\varepsilon, \varepsilon\rangle_n}_{\textit{interpolation error}} + \underbrace{2|\langle(\mathbb{I} - \Pi_{\tau_b})^2 f, \varepsilon\rangle_n|}_{\textit{cross term}}.$$

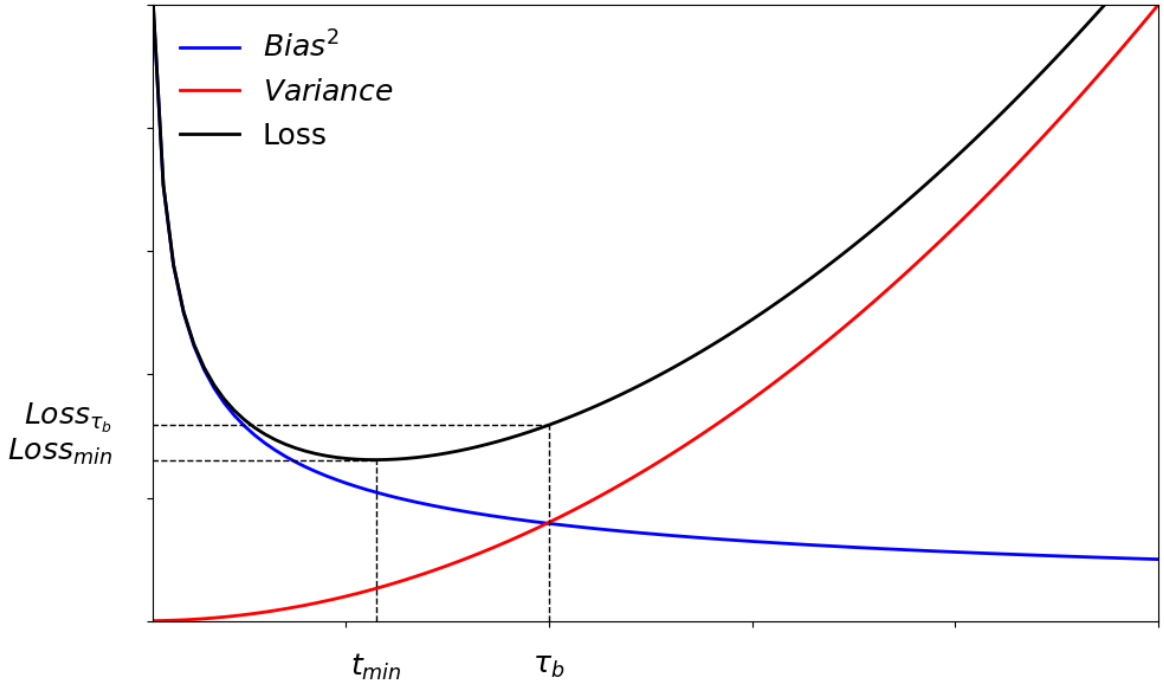


Intuition on the bound

- Early stop error: κ has to be chosen around $\|\varepsilon\|_n^2$
- Interpolation error: Price we pay for interpolation since the projection flow is not orthogonal
- Cross-term: Comes from the dependence of Π_t, τ_b and ε

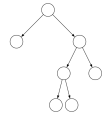


Generalized projection flow



- By monotonicity of bias and variance, we have that

$$B_t^2 \geq B_{\tau_b}^2 \text{ for } t \leq \tau_b, \text{ and } S_t^2 \geq S_{\tau_b}^2 \text{ for } t \geq \tau_b$$

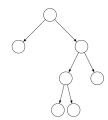


Stopping threshold

- In practice, the stopping threshold κ is unknown and needs to be estimated (e.g., by scaled lasso in linear case, see Stankewitz, 2024).
- Nearest-neighbour estimator by Devroye et al. (2018)

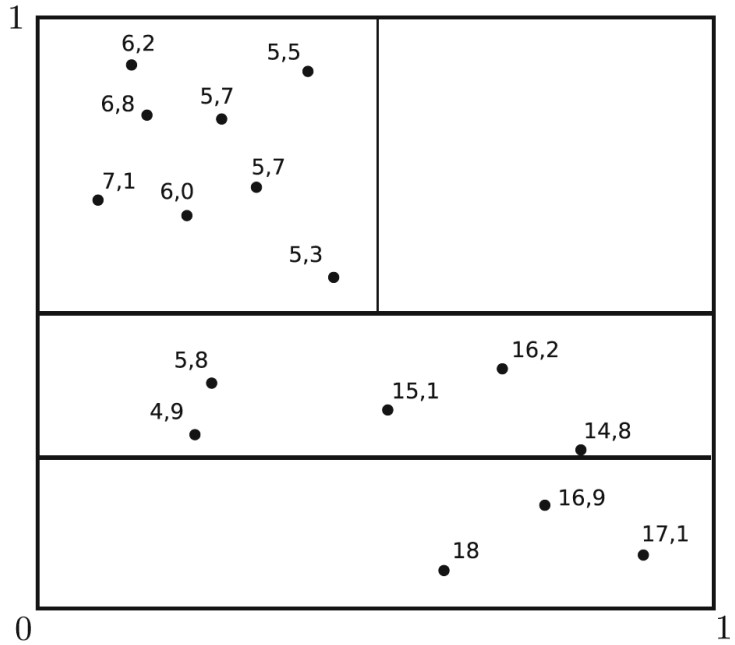
$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} Y_i^2 - \frac{1}{n}\sum_{i=1}^{n} Y_i Y_{nn(i)},$$

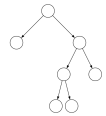
where $X_{nn(i)}$ is the nearest neighbour of X_i with respect to the Euclidean distance in $\mathbb{R}^d$ and $Y_{nn(i)}$ is the corresponding response.



Assume Π_t data-independent

- Assumption of data-independent Π_t is similar to centered forest or uniform random forest (Biau, 2012; Arlot & Genuer, 2014; and Klusowksi (2021))
- A coordinate is uniformly chosen across dimensions and split at the center along the selected coordinate





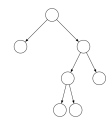
Independent splitting

Theorem

If the projection flow (Π_t) *for semi-global early stopping is independent of* $(X_i, Y_i)_{i=1,\dots,n}$ *and* $\varepsilon_i \sim N(0, \sigma^2)$*, then we have the oracle inequality*

$$\mathsf{E}[\|\hat{f}_\tau - f\|_n^2] \lesssim \inf_{t \in [0,n]} \mathsf{E}[\|\hat{f}_t - f\|_n^2] + \mathsf{E}[|\kappa - \|\varepsilon\|_n^2|] + \frac{\sigma^2 \log(2n)}{n}.$$

- For f Lipschitz, the minimax rate is $n^{-2/(2+d)}$. A 1-NN residual estimator $\hat{\sigma}^2$ estimates $\|\varepsilon\|_n^2$ with rate $n^{-2/d} \vee n^{-1/2}$.
- Conclusion: Early stopping with $\kappa = \hat{\sigma}^2$ adapts to the semi-global oracle for f Lipschitz, $d \geqslant 2$.



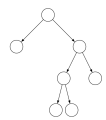
Dependent splitting

Theorem

Consider the projection flow (Π_t) *for global or semi-global early stopping and assume* $\varepsilon_i\bar{\sigma}$*-subgaussian. Then we have the oracle-type inequality*

$$\mathsf{E}[\|\hat{f}_\tau - f\|_n^2] \lesssim \mathsf{E}\Big[\inf_{t\in[0,n]}\Big(\|(\mathbb{I}-\Pi_t)f\|_n^2 + \|\Pi_t\varepsilon\|_n^2\Big)\Big] + \mathsf{E}[|\kappa - \|\varepsilon\|_n^2|] + \frac{\bar{\sigma}^2\log((d\vee 2)n)}{n}\mathsf{E}[\tau_b+1].$$

- The cross term dominates the interpolation error, since for t iterations there are $\mathcal{O}\left((dn)^t\right)$ splitting possibilities.
- Best known bound for stochastic error for CART splitting at the balanced oracle is $\bar{\sigma}^2\mathsf{E}[\tau_b]\log^2(n)\log(dn)/n$. (Klusowski & Tian, 2024)

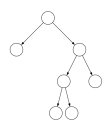


Application

Data set	d, n	Pruning	Global	Global Int	Two-Step	Semi
Boston	14, 506	3.89 (23/17s)	4.87 (8/0.3s)	5.12 (8/0.3s)	3.97 (8/1s)	5.35 (5/0.6s)
Comm.	100, 1994	0.15 (8/158s)	0.15 (8/3s)	0.16 (8/3s)	0.15 (8/14s)	0.17 (26/7s)
Abalone	8, 4177	2.34 (20/112s)	2.38 (16/2s)	2.41 (16/2s)	2.33 (22/8s)	2.58 (51/2s)
Ozone	9, 330	4.75 (8/8s)	4.72 (8/0.1s)	4.68 (8/0.1s)	4.74 (10/1s)	5.05 (7/0.2s)
Forest	11, 517	38.79 (6/10s)	32.94 (4/0.08s)	31.98 (4/0.08s)	31.51 (3/0.4s)	32.81 (3/0.1s)

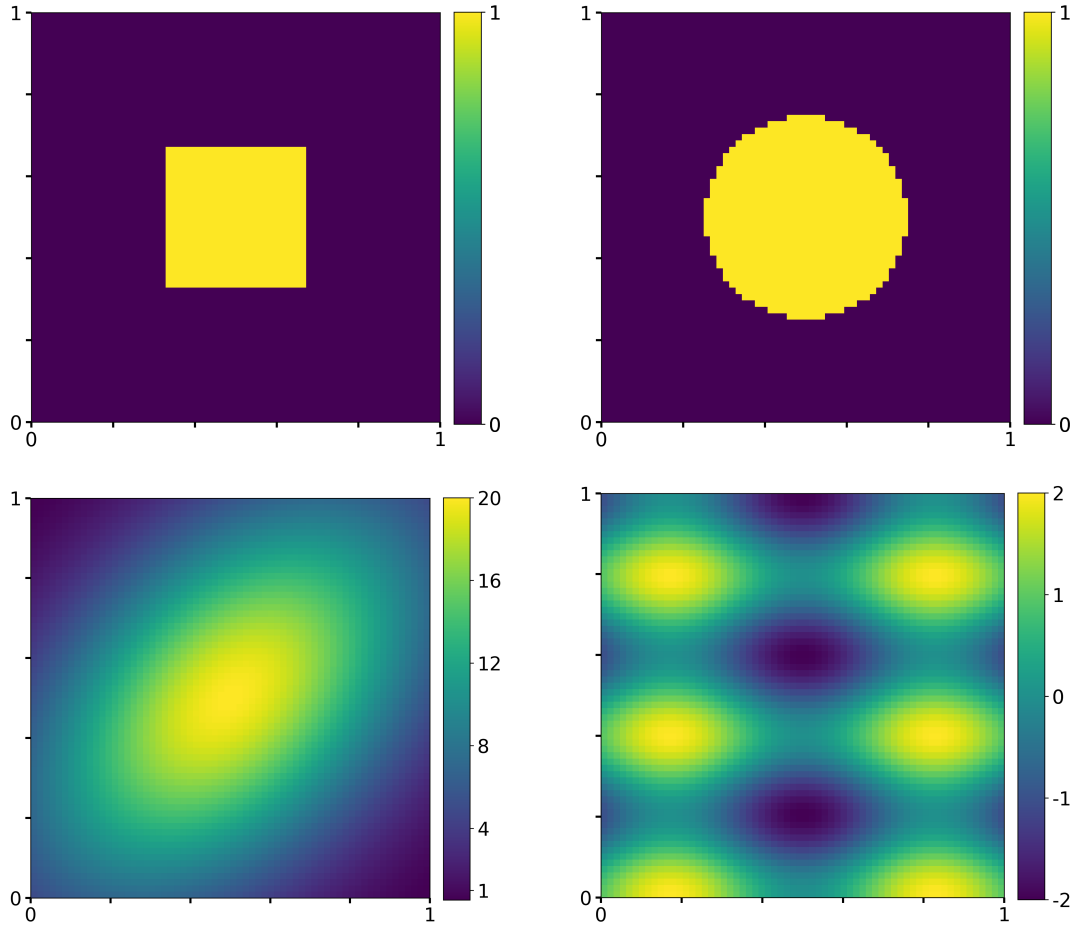
Table: Median RMSE (median terminal nodes/run times) for different datasets and methods.

Major improvement: Computational time (factor of ≈ 50)!



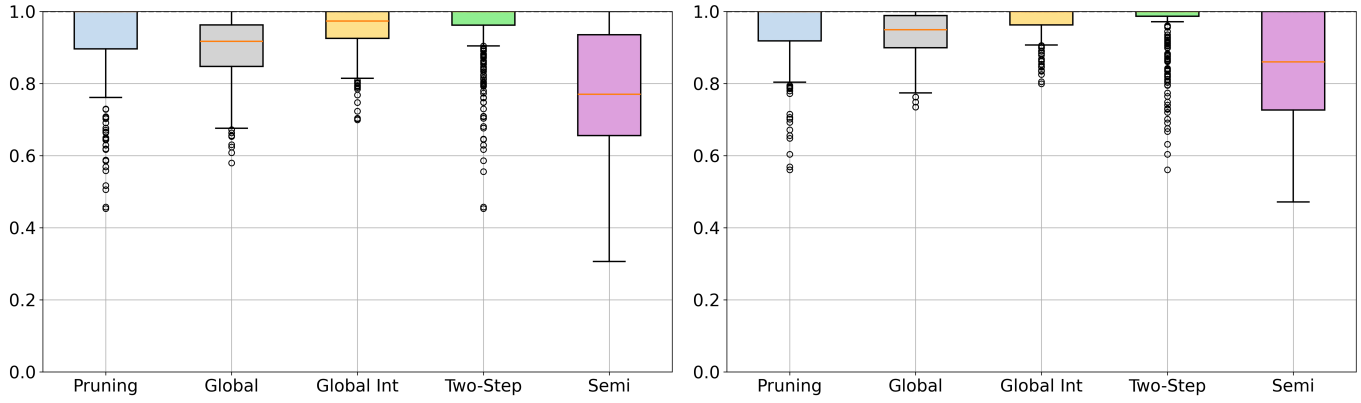
Simulation

- $X_i \overset{\text{iid}}{\sim} U(0,1)^5$, $n = 1000$, $\varepsilon \sim N(0,1)$, $d = 5$, $M = 300$ MC runs



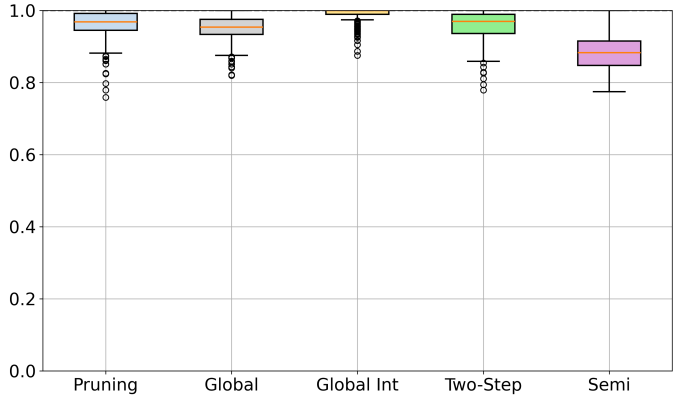
Relative efficiency

- $\min_t \|\hat{f}_t - f\|_{n'} / \|\hat{f}_\tau - f\|_{n'}$ on test set (larger is better)

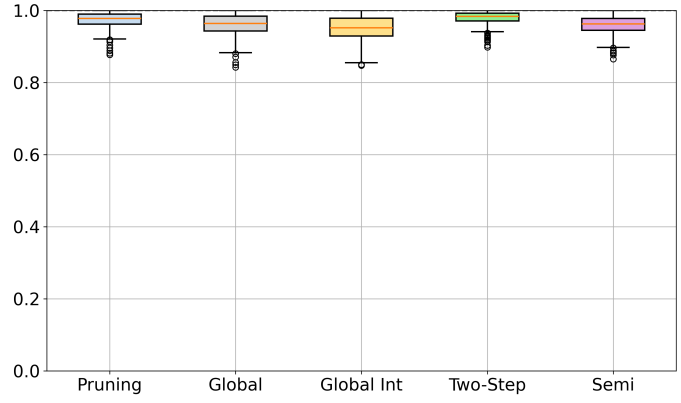


(a) Rectangular function

(b) Circular function

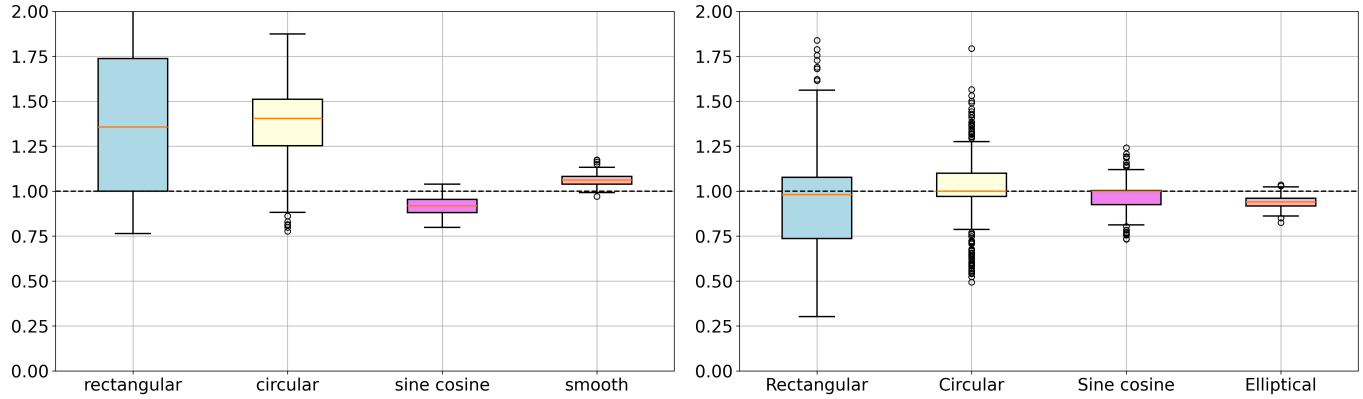


(c) Sine cosine function



(d) Elliptical function

Oracle ratios

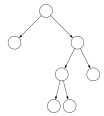


(a) $\rho_{glob,semi}$(smaller in favour of global)

(b) $\rho_{prun,semi}$(smaller in favour of pruning)

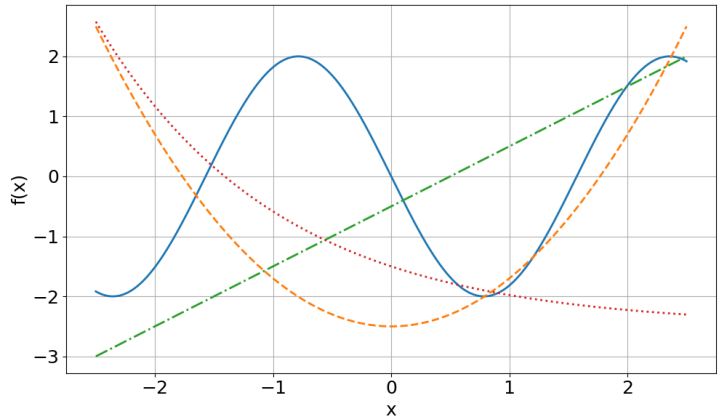
$$\rho_{glob,semi} = \min_t \|\hat{f}_t^{glob} - f\|_{n'} / \min_t \|\hat{f}_t^{semi} - f\|_{n'},$$

$$\rho_{prun,semi} = \min_t \|\hat{f}_t^{prun} - f\|_{n'} / \min_t \|\hat{f}_t^{semi} - f\|_{n'}.$$

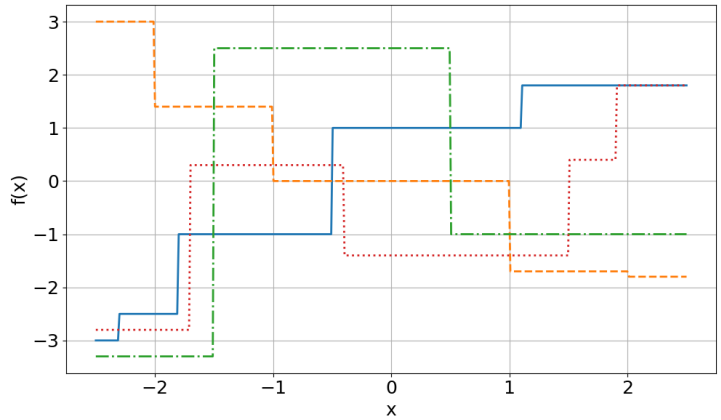


Simulation: sparse additive model

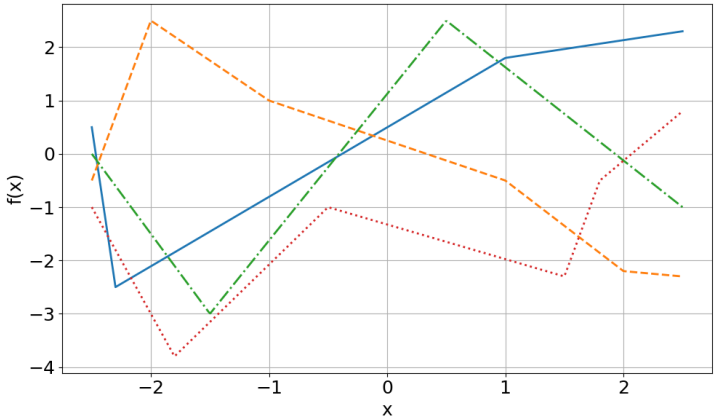
- $X_i \overset{\text{iid}}{\sim} U(-2.5, 2.5)^{30}$, $n = 1000$, $\varepsilon \sim N(0, 1)$, $M = 300$ MC runs



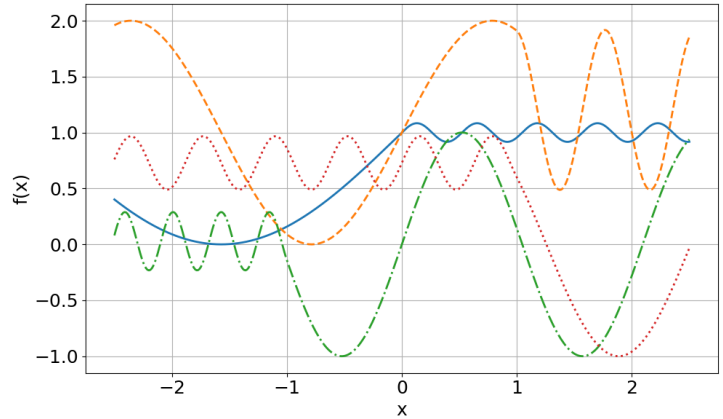
(a) Smooth function



(b) Step function



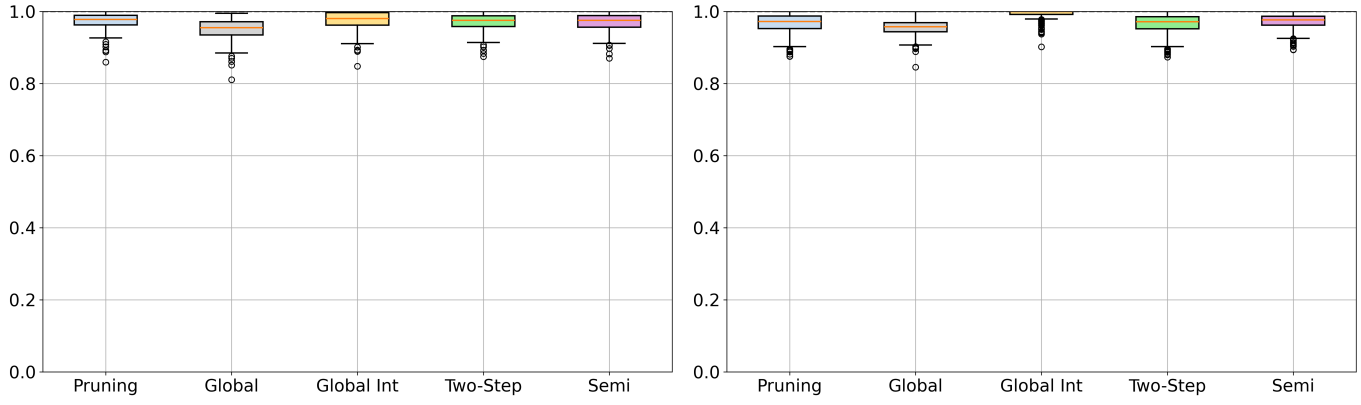
(c) Piecewise linear function



(d) Hills-type function

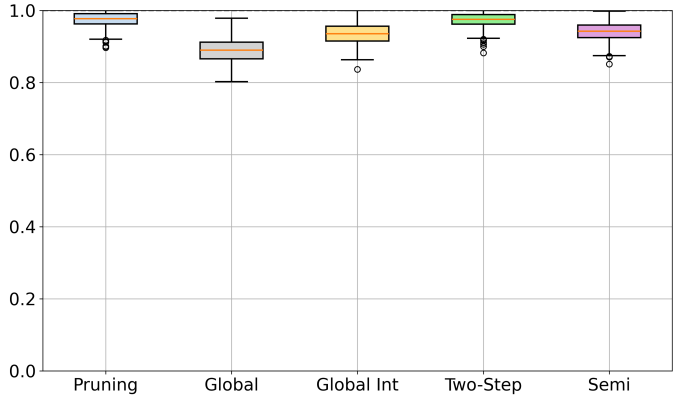
Relative efficiency, sparse additive model

- $\min_t \|\hat{f}_t - f\|_{n'} / \|\hat{f}_\tau - f\|_{n'}$ on test set (larger is better)

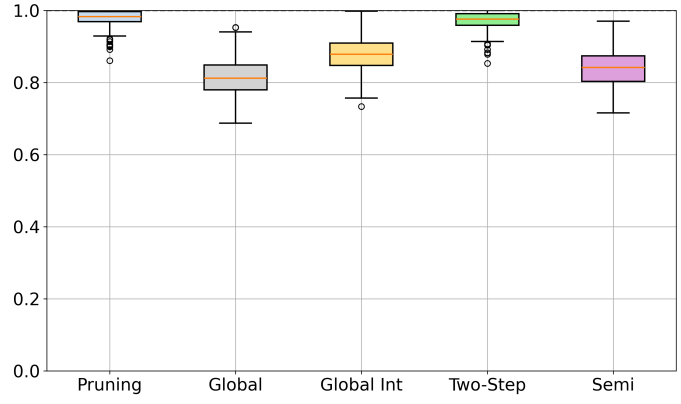


(a) Smooth function

(b) Step function



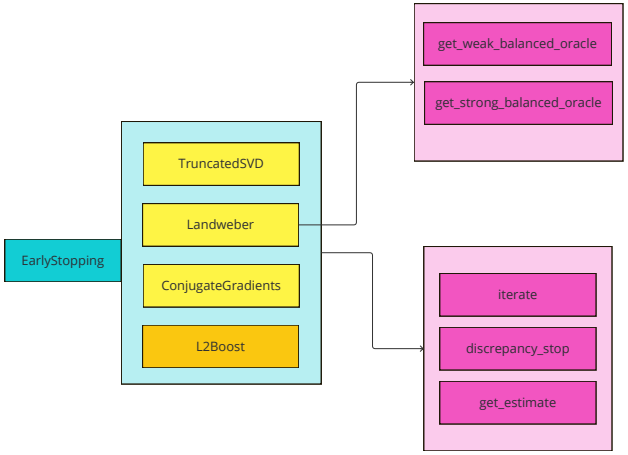
(c) Piecewise linear function

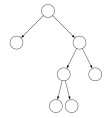


(d) Hills-type function

Python library 'EarlyStoppingPy'

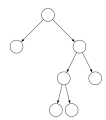
- Unifying library of Early Stopping methods for
  - Landweber algorithm (Blanchard, Hoffmann & Reiß, 2018b)
  - L2 Boosting (Stankewitz, 2024)
  - truncated SVD (Blanchard, Hoffmann & Reiß, 2018a)
  - Conjugate Gradient (Hucker & Reiß, 2024)
  - Regression tree (Miftachov & Reiß, 2025)
- pip install EarlyStoppingPy





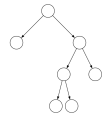
Conclusion

- Generalised projection flow as a unifying framework for early stopped smoothing estimators
- Oracle inequalities for the early stopped CART estimator in global and semi-global order
- Ready-to-use algorithm for practitioners by using nearest neighbour estimator $\hat{\sigma}^2$ as a stopping threshold



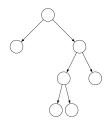
Questions?

Thank you for this wonderful seminar!



An extension: random forest classifier

- Consensus in the applied community for the random forest: grow each tree to its full depth in a forest (Breiman, 2001)
- The Google research team (Zürich) grows the random forest with a pre-specified default depth for each tree
- Zhou and Mentch (2023) argue that the full depth is not required for the advantage of averaging the trees
- Extend the regression tree to the random forest (or at least an ensemble of trees)



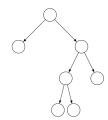
Nonparametric classification

- Let the covariates be $(X_{i,1}, \ldots, X_{i,d})^\top \in \mathcal{X} \subseteq \mathbb{R}^d$ and response $Y_i \in \mathcal{Y} \subseteq \mathbb{R}$ and the binary response is $Y_i \subseteq \{0, 1\}$

$$Y_i \sim \text{Ber}\left(f(X_i)\right), \text{ with } \varepsilon_i = Y_i - f(X_i) \text{ for } i = 1, \ldots, n,$$

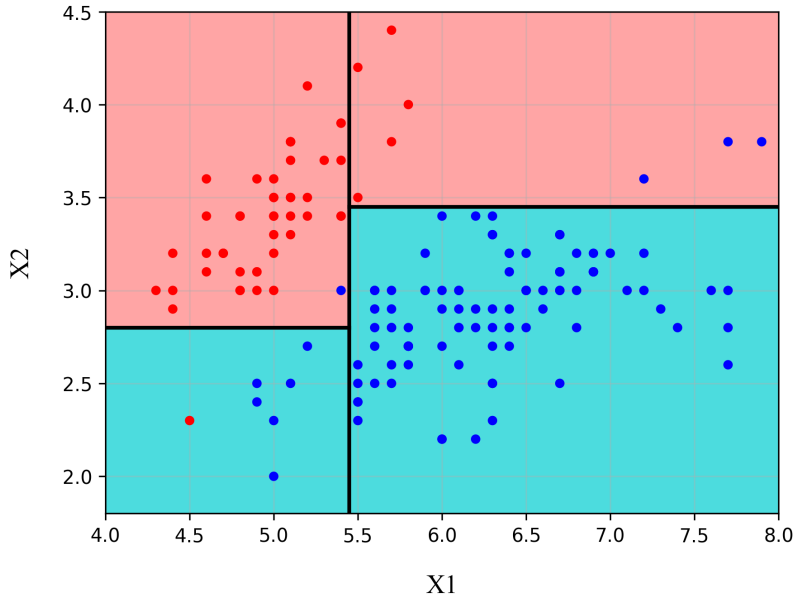
  where the noise ε_i is independent but heteroskedastic
- Goal is to estimate the unknown conditional probability $f(x) = \mathbb{P}(Y = 1 \mid X = x)$
- Plug-in classifier gives the label at query point x:

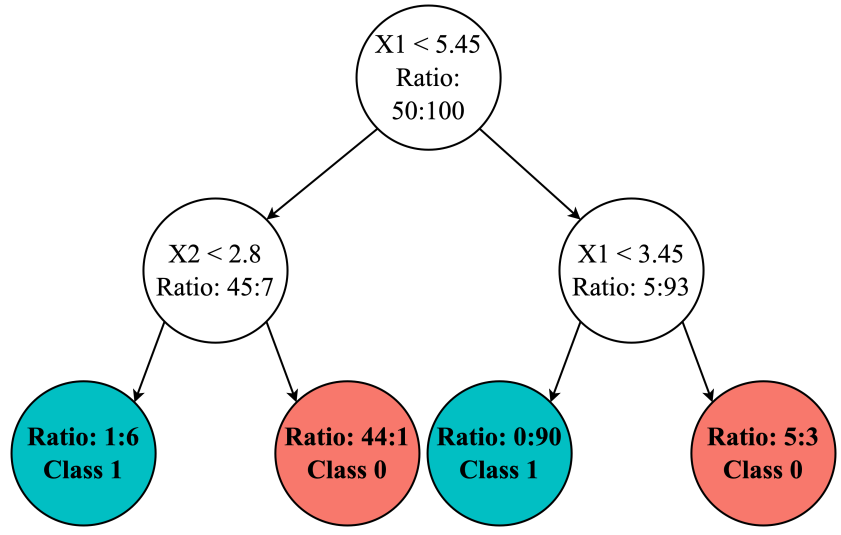
$$c(x) = \begin{cases} 1, & \text{if } f(x) \geq 0.5, \\ 0, & \text{otherwise.} \end{cases}$$

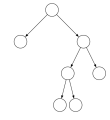


Splitting criterion stays the same!

- Gini impurity for classification trees is equivalent to squared error impurity, thus splitting criterion is the same as in regression tree, see Louppe (2014)



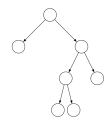




Early stopping for *random* tree

- Random forest is characterized by bootstrap sampling and random sampling of variables at each split
- Denote both sources of randomness with the i.i.d. realizations $\Theta_1, \ldots, \Theta_B$ of the random variable Θ.
- Classification tree estimator at iteration m is given by

$$\begin{aligned}\hat{f}_m(x; \Theta_b) &:= \Pi_m(\Theta_b) Y \\ &:= \frac{1}{n_k^m(\Theta_b)} \sum_{i: X_i \in A_k^m(\Theta_b)} Y_i \quad \text{for } x \in A_k^m(\Theta_b).\end{aligned}$$



Early stopping for *random* tree

- Projection flow for iteration $t = m + \alpha$ and $\alpha \in [0, 1]$

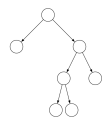
$$\Pi_t(\Theta_b) := (1 - \alpha)\Pi_m(\Theta_b) + \alpha\Pi_{m+1}(\Theta_b)$$

- Interpolated estimator and residual norm follow as

$$\hat{f}_t(x; \Theta_b) = \Pi_t(\Theta_b)Y, \quad R_t^2(\Theta_b) = \|Y - \hat{f}_t(\Theta_b)\|_n^2 = \|(\mathbb{I} - \Pi_t(\Theta_b))Y\|_n^2$$

- Stopping rule for each randomized tree is given by

$$\tau_b := \min\left\{t \in [0, n] | R_t^2(\Theta_b) \leqslant \kappa\right\}$$



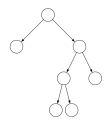
Early stopping for tree ensemble

- Early stopped random forest classifier follows as

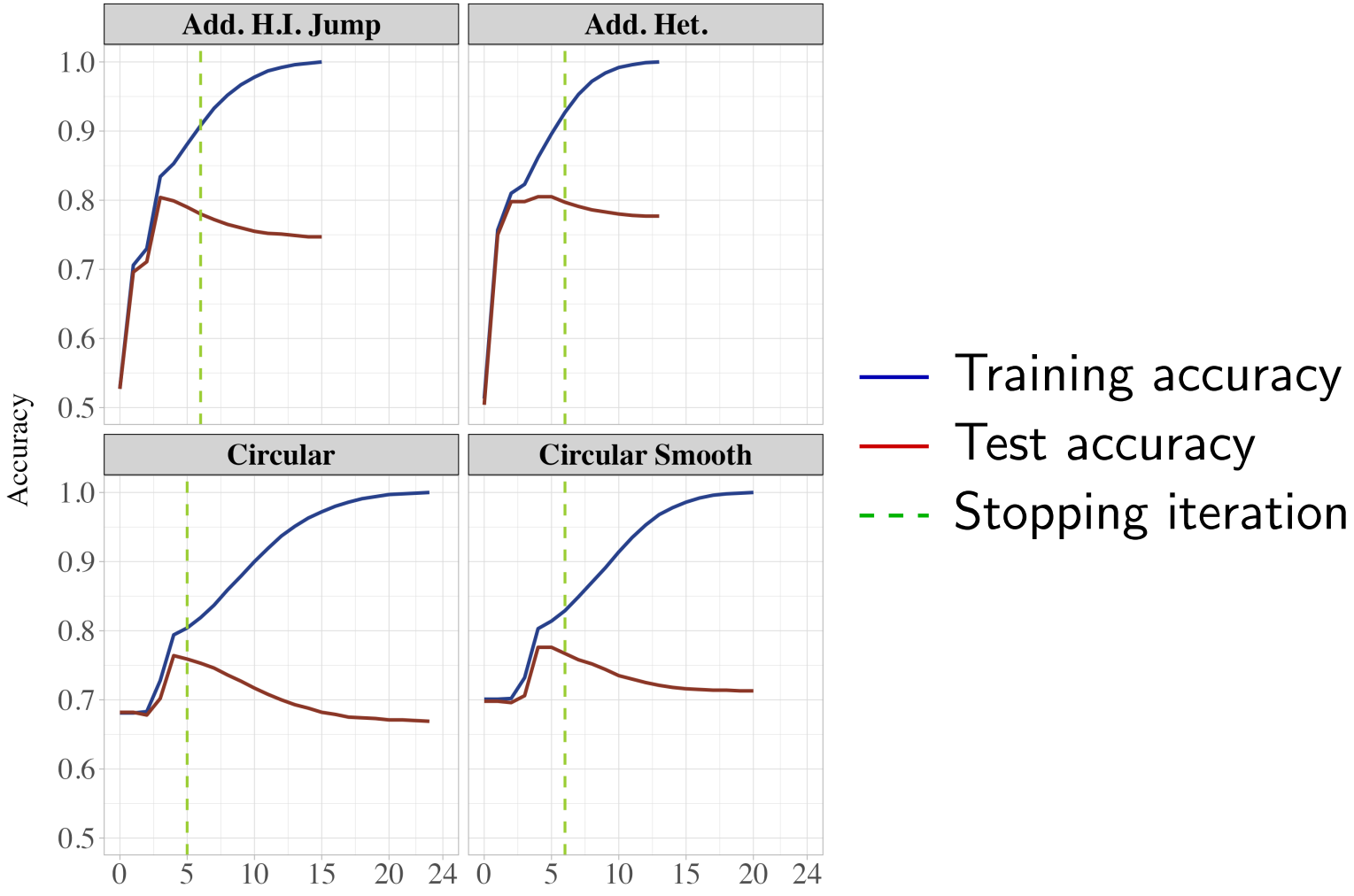
$$\hat{f}_{RF}(x) := \hat{f}_{RF}(x; \Theta_{1:B}, \tau_{1:B}) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_{\tau_b}(x; \Theta_b)$$

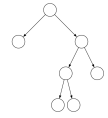
- Plug-in classifier is given by

$$\hat{c}(x) = \begin{cases} 1, & \text{if } \hat{f}_{RF}(x) \geq 0.5, \\ 0, & \text{otherwise.} \end{cases}$$

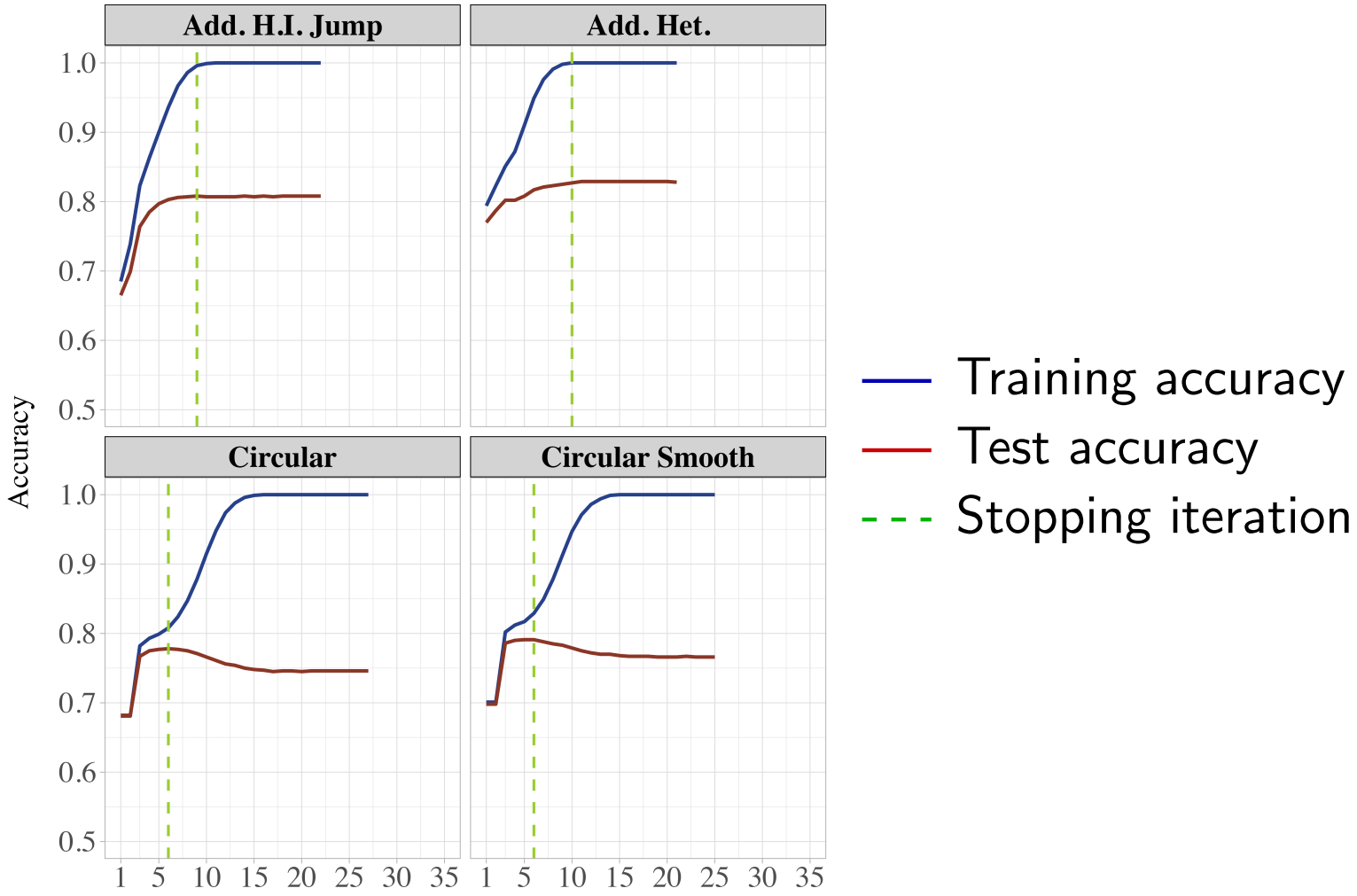


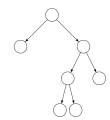
Simulation results: single tree classifier





Simulation results: random forest classifier





Questions?

Thank you for this wonderful seminar!