universität innsbruck

Boosting Distributional Soft Regression Trees

Hard skills alone sometimes won’t cut it!

Nikolaus Umlauf

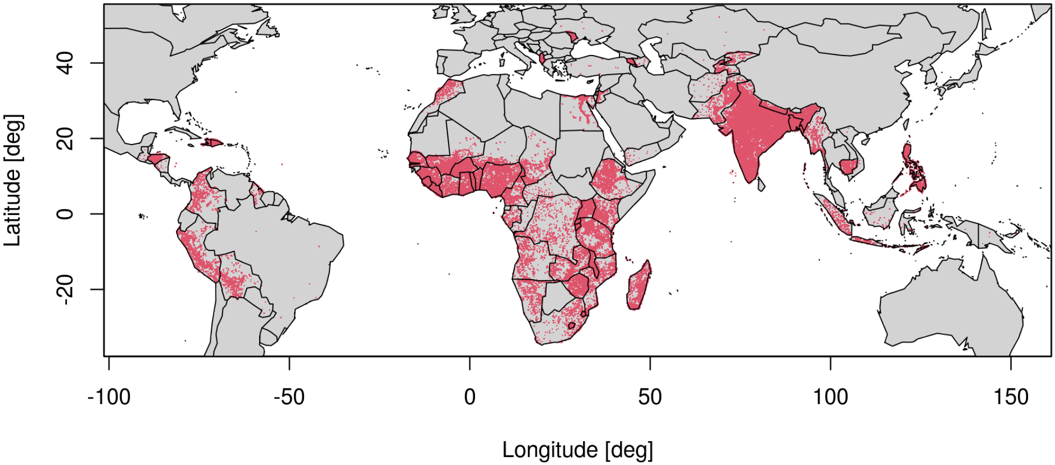
http://nikum.org

Child Anaemia Risk

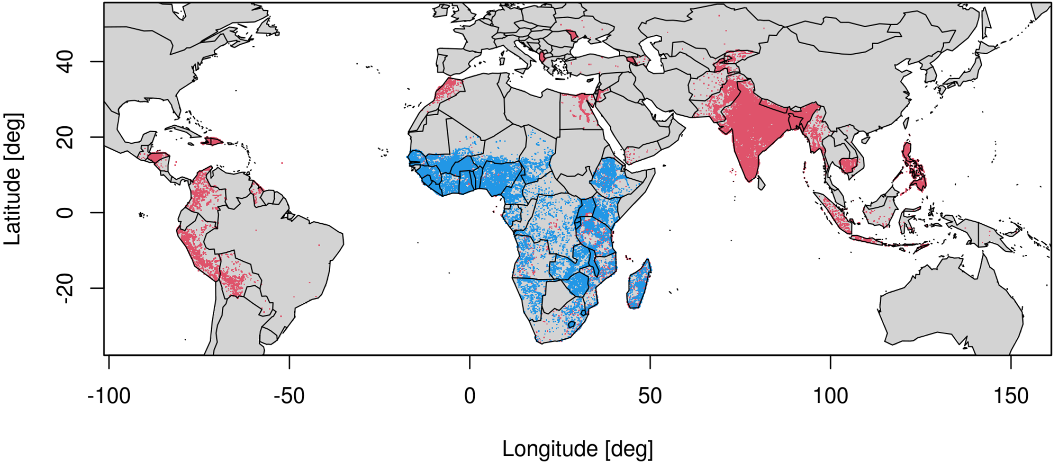
Joint work with:
Johannes Seiler, Mattias Wetscher and Nadja Klein.

- Project aiming to better explain childhood problems in low- and middle-income countries.
- Contribute to monitoring of the Sustainable Development Goals (SDG).
- We compiled a brand new data set using DHS data.
- Data on global conflicts, topography and environmental data from satellite earth observations (NDVI), temperature and precipitation data from ERA5.
- Data from 1990–2019 with $n > 3M$ observations.

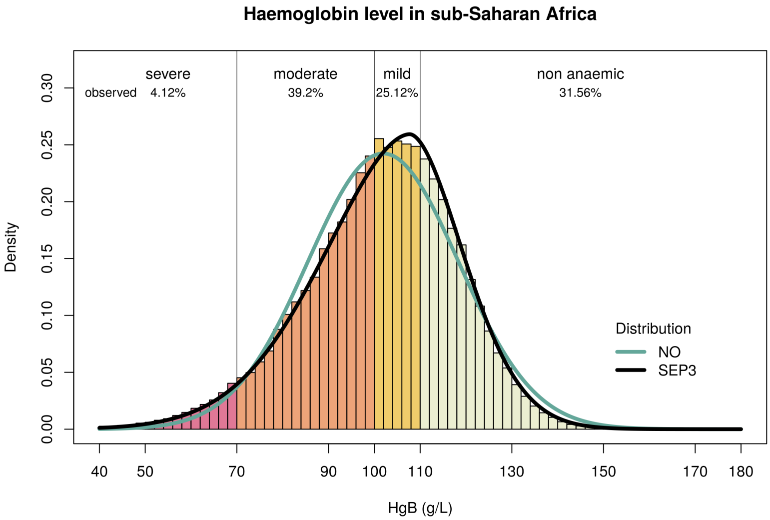
Child Anaemia Risk



Child Anaemia Risk



Child Anaemia Risk



Modeling Challenges

- Distributional regression using large data sets?
- Variable selection?
- Capturing complex interactions, space-time, etc.?

Model Specification

Any parameter of a population distribution $\mathcal{D}_y$ may be modeled by

$$y \sim \mathcal{D}_y \left(h_1(\theta_1) = \eta_1, \; \ldots \;, \; h_K(\theta_K) = \eta_K\right), \quad k = 1, \ldots, K,$$

and

- $h_k(\cdot)$: Link functions for each distribution parameter.
- η_k: Predictors modeled by covariates.

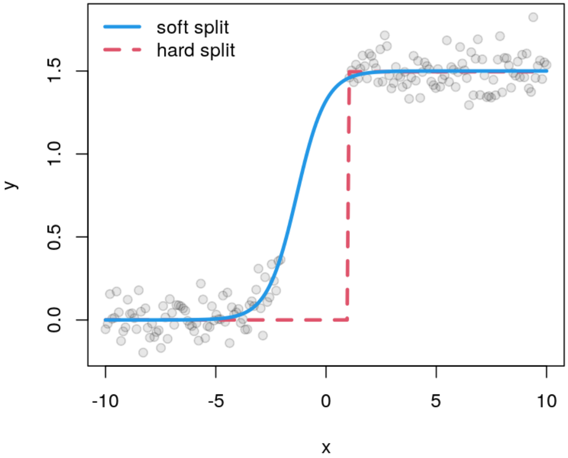
Now, instead of using a traditional structured additive predictor, we introduce a more flexible approach by employing adaptive *Soft Regression Trees* with

$$\boldsymbol{\eta}_k \equiv f_k(\mathbf{X}) = \beta_{k,0} + \sum_{j=1}^{J_k} P_{k,j}(\mathbf{X}, \boldsymbol{\Omega}_{(k,j)})\beta_{k,j}.$$

Soft Regression Trees

- Hard binary split yields only two possible predictions:
- $\hat{y} = 0$ for $x < 1$ and $\hat{y} = 1.5$ for $x \geq 1$.

- Soft split allows a smooth transition.
- Rather than assigning observations to single nodes, soft split uses a better balanced weighting.



Soft Regression Trees

Growing a *Soft Tree*:

- Root node is "split softly" by

$$N_l(\boldsymbol{x}_i) = N_l^L(\boldsymbol{x}_i) \cdot p_l(\boldsymbol{x}_i) + N_l^R(\boldsymbol{x}_i) \cdot (1 - p_l(\boldsymbol{x}_i)),$$

- with weighting function $p_l(\cdot) : \mathbb{R} \mapsto [0, 1]$, e.g.,

$$p_l(\boldsymbol{x}_i) = \frac{1}{1 + \exp(-(\boldsymbol{x}_i^\top \omega_l))},$$

  where ω_l are weights that need to be estimated.

- For terminal nodes $N_l^L(\boldsymbol{x}_i)$ and $N_l^R(\boldsymbol{x}_i)$ we have

$$N_l(\boldsymbol{x}_i) = \beta_l^L \cdot p_l(\boldsymbol{x}_i) + \beta_l^R \cdot (1 - p_l(\boldsymbol{x}_i)).$$

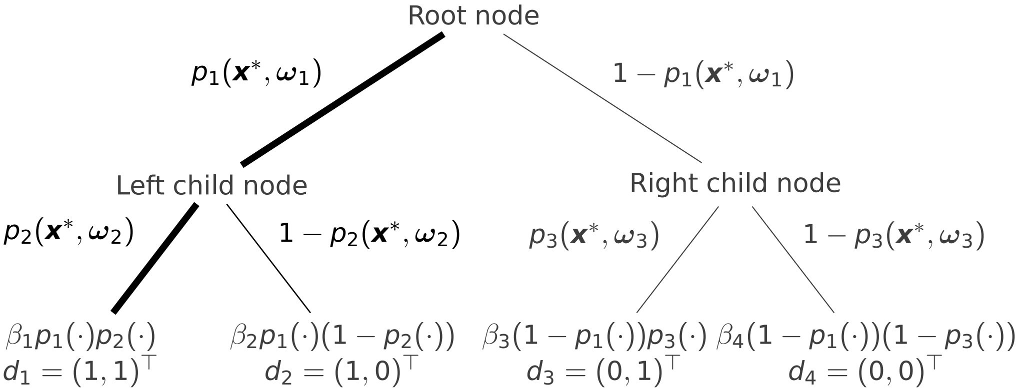
Soft Regression Trees

- Given set of weights $\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_T$ for T terminal nodes.
- Predictions can be computed by linear combination $N(\boldsymbol{x}^*, \boldsymbol{\Omega})^\top \boldsymbol{\beta}$,
- with $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_T)^\top$ and $N(\boldsymbol{x}^*, \boldsymbol{\Omega})^\top = (P_1(\boldsymbol{x}^*, \boldsymbol{\Omega}_1), \ldots, P_T(\boldsymbol{x}^*, \boldsymbol{\Omega}_T))^\top$
- Path probabilities are computed by

$$P_l(\boldsymbol{x}^*, \boldsymbol{\Omega}_l) = \prod_{r \in \mathcal{D}_l} p_r(\boldsymbol{x}^*)^{d_r} (1 - p_r(\boldsymbol{x}^*))^{1-d_r},$$

with $d_r \in \{0, 1\}$ indicating the binary directions (left/right) and $\mathcal{D}_l$ is the set of nodes involved in one path D_l.

Soft Regression Trees



Each path D_l, $l = 1, \ldots, 4$ from the top root node to one of the four terminal nodes represents one column of the design matrix $N(\mathbf{X}, \boldsymbol{\Omega}) \in \mathbb{R}^{n \times T}$.

Soft Regression Trees

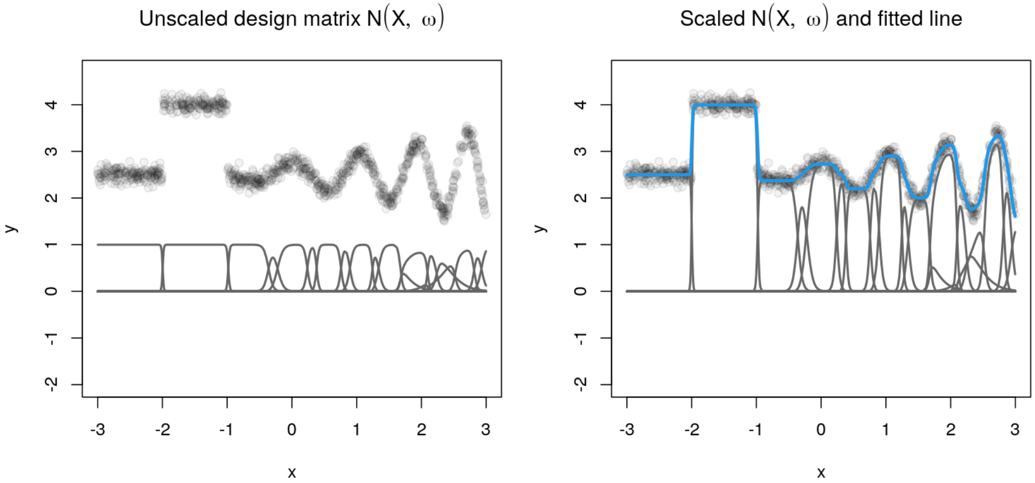
Adaptive *Soft Tree*:

- Let D_l be any path from the root node to any node $N_l(\boldsymbol{x}_i)$.
- Let $\mathcal{D}_l$ be the set of nodes involved in forming in path D_l with path probability P_l and a set of weights $\boldsymbol{\Omega}_l$.
- The adaptive *Soft Tree* is

$$f(\boldsymbol{x}_i) = \beta_0 + \sum_{j=1}^{J} P_j(\boldsymbol{x}_i, \boldsymbol{\Omega}_j)\beta_j.$$

- Tree structure allows to decompose into coarse and fine elements,
- with finer structures controlled by the rearmost elements of the sum.
- Algorithm follows path with highest improvement, similar to classical trees.
- Estimation of parameters by ML, due to adaptive structure considerably fast.

Soft Regression Trees

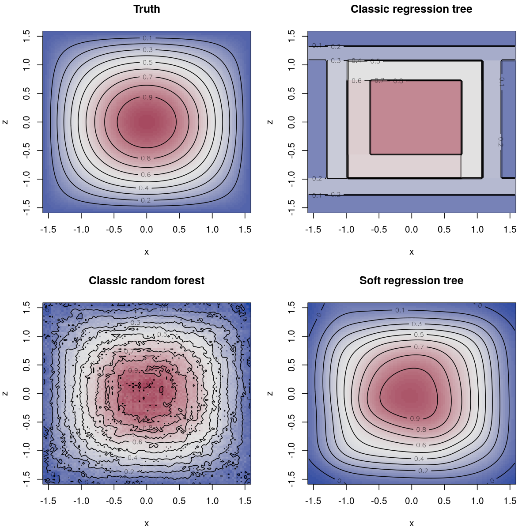


Soft Regression Trees

- True function $f(x, z) = \sin(x) \cdot \sin(z)$,
- $y_i = f(x_i, z_i) + \varepsilon_i$ with $\varepsilon_i \sim N(0, 0.1^2)$,
- $n = 10000$.

- Rough approx. of *Classic Tree*.
- *Random Forest* with 2000 trees is not able to reproduce the smooth surface.

- *Soft Tree* with 24 terminal nodes able to reproduce the true function quite well.



Boosting Distributional Soft Regression Trees

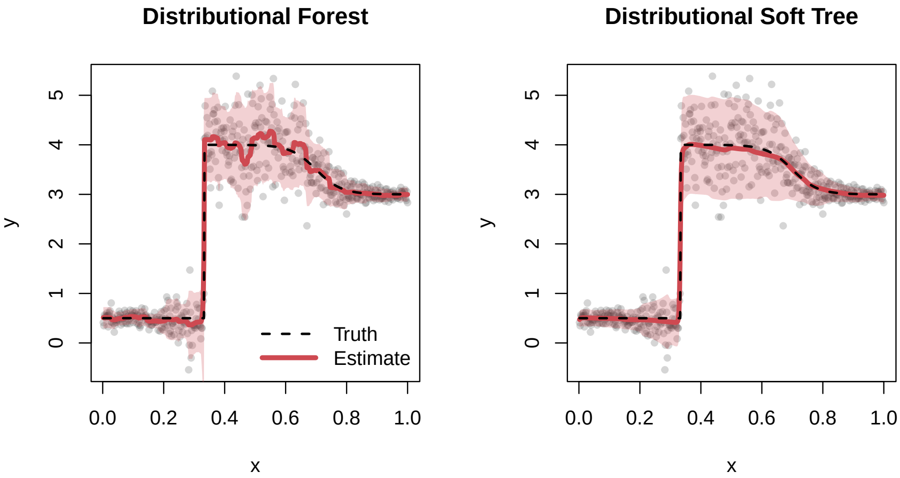
Algoritm Sketch:

- Use univariate soft split for $P_{k,j}(\cdot)$, i.e., incorporate variable selection.
- When growing the tree, select best performing node and soft split covariate according to log-likelihood contribution.
- At each iteration t and for each distributional predictor update

$$\eta_k^{[t+1]} = \eta_k^{[t]} + \nu \cdot f_k^{[t]}(\mathbf{X}),$$

  with step length parameter ν (e.g., $\nu = 0.1$).
- Use *Soft Trees* with small depth, enforces slow improvement.

Boosting Distributional Soft Regression Trees



Boosting Distributional Soft Regression Trees

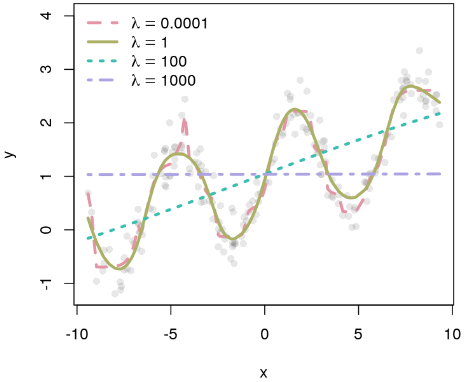
Extensions:

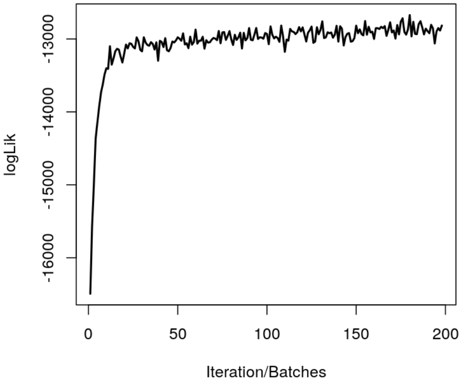
- Optimize weights using shrinkage parameter

$$\text{pen ML}(\boldsymbol{\omega}_{rk} | \mathbf{y}, \mathbf{X}) = \underset{\boldsymbol{\omega}_{rk}}{\text{arg max}}\ \ell(\boldsymbol{\omega}_{rk}; \mathbf{y}, \mathbf{X}) - \lambda_k J(\boldsymbol{\omega}_{rk}),$$

  improves stability, additionally avoids overfitting.
- For big data, use randomly selected subset $\boldsymbol{s}^{[t]} \subset \{1, \dots, n\}$, i.e., $\mathbf{X}_{\boldsymbol{s}^{[t]}}$.
- Batch updates only accepted, if log-likelihood is increased after all parameters θ_k are updated.
- In practice, “convergence” is achieved when the log-likelihood improvements become small and appear to fluctuate around a certain level.
- Two-step approach, drop all features with small contributions, refit.

Boosting Distributional Soft Regression Trees





Simulation

We consider models using the following predictors:

$$
\begin{aligned}
\eta_\mu &= \left[\left(10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5\right) - 1.5\right]\frac{2}{26.48} + 1 \\
\eta_\sigma &= \left(\left(z_1^2 + \left(z_2 z_3 - \frac{1}{z_2 z_4}\right)^2\right)^{0.5} - 7.96\right)\frac{2}{1736.85} - 2.5 \\
\eta_\nu &= \sin(2x_1)\cos(0.5x_3) + 1 \\
\eta_\tau &= 0.5x_2^2 - 1.
\end{aligned}
$$

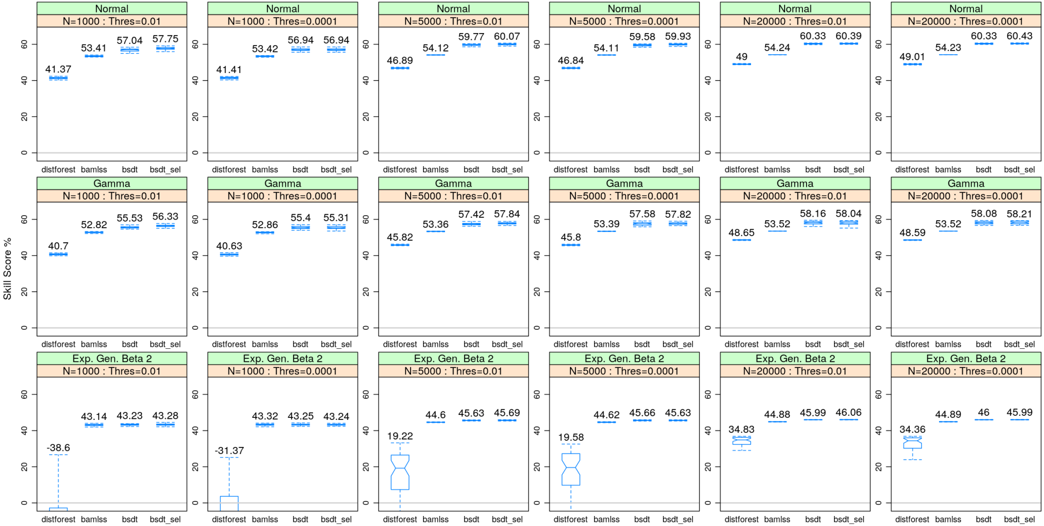
Predictor η_μ and η_σ are scaled versions of the Friedman 1 and 2 functions.

Simulation

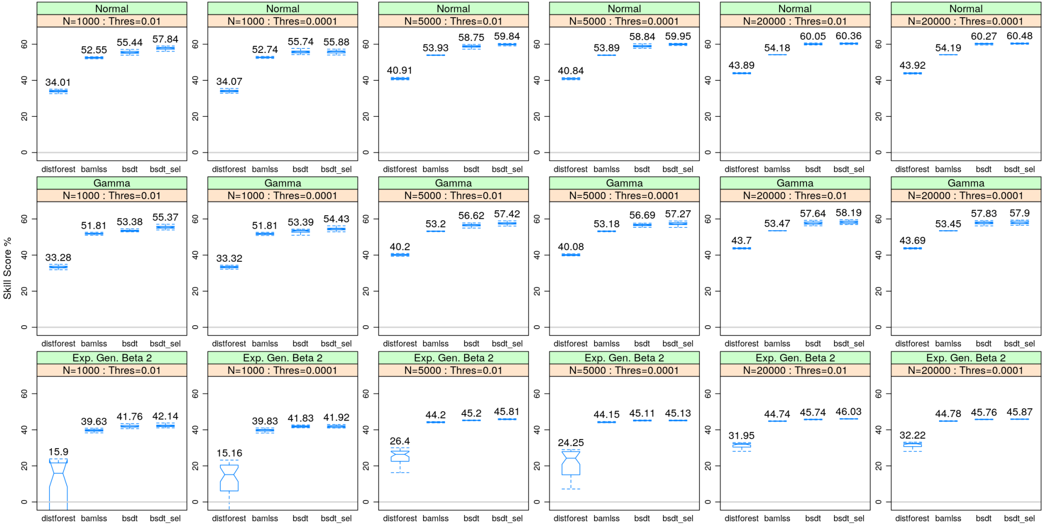
Scenarios:

- We investigate performance for `NO()`, `GA()` and `EGB2()` distribution,
- using $n = 1000, 5000, 20000$ observations.
- We study performance using $\texttt{nnoise} = 0, 20$ variables.
- Two settings for covariate data:
  uncorrelated $\rho = 0$ and correlation of $\rho = 0.7$.
- 50 replications each, evaluated using skill score computed by comparison to naive model CRPS.
- Algorithms: *bamlss*, *distforest*, *bsrt*, *bsrt_sel*.

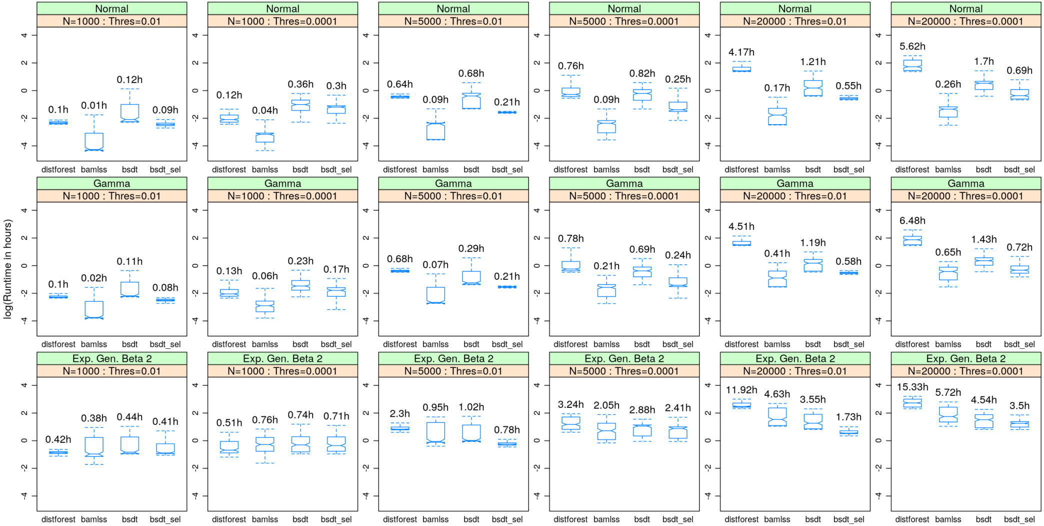
Simulation ($\rho = 0$, $\texttt{nnoise} = 0$)



Simulation ($\rho = 0.7$, $\texttt{nnoise} = 20$)



Simulation (Runtime)



Application (Estimation)

Required packages.

```
R> library("softtrees")
R> library("gamlss.dist")
```

Model formula.

```
R> f <- hgb ~ cage + gender + mbmi + magebirth + bord + hhs + ai + lgdp +
+     distance + nl12 + ndvi12 + pre12 + t2m12 + soil + ... + x + y
R> f <- rep(list(f), 4)
```

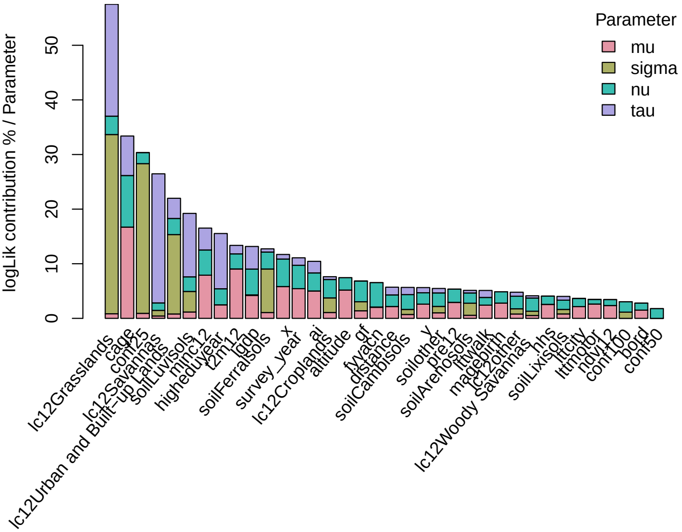
Generate batches.

```
R> set.seed(123)
R> batch_ids <- lapply(1:200, function(i) { sample(nrow(df), size = 10000) })
```

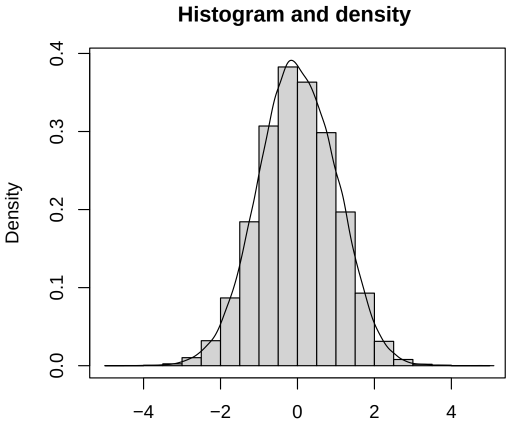
Estimate model.

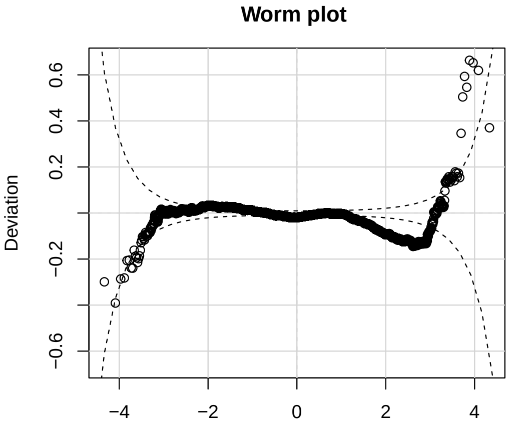
```
R> b <- bsdt(f, data = df, family = EGB2,
+     batch_ids = batch_ids, k = 2, nu = 0.1, lambda = 0.01)
```

Application (Variable Selection)

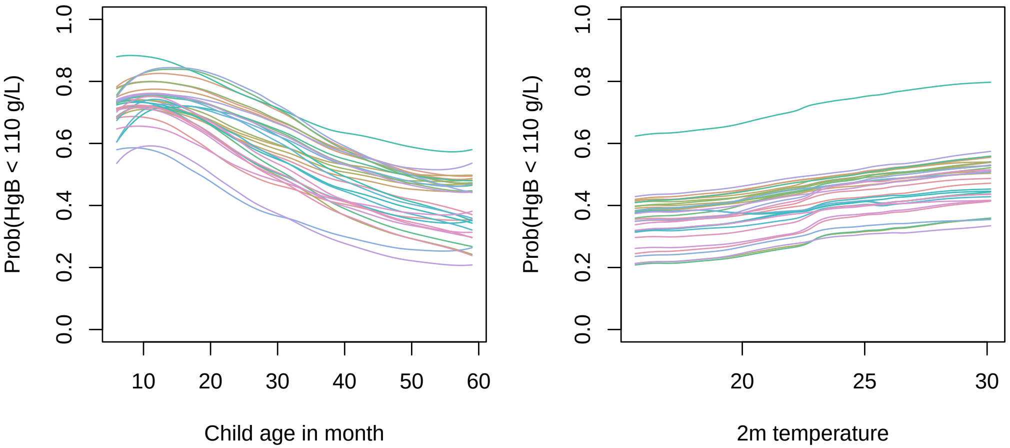


Application (Quantile Residuals)

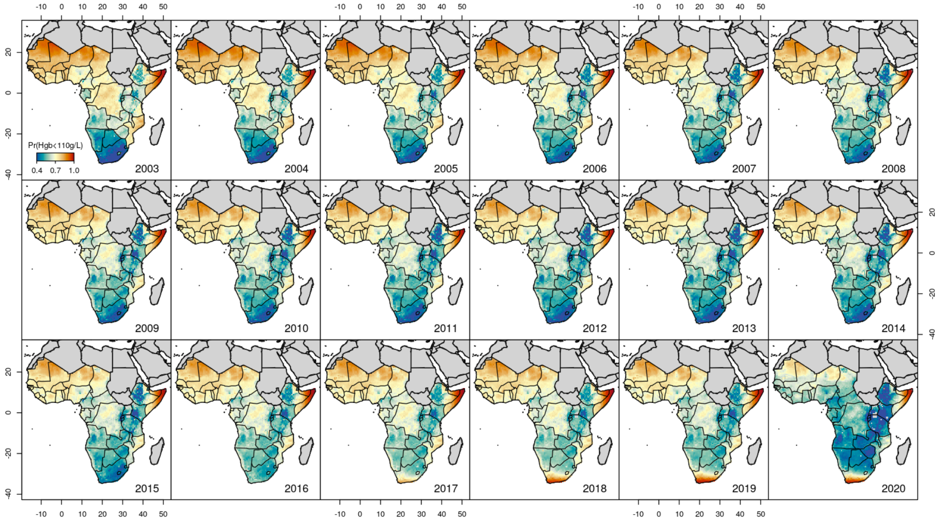




Application (Marginal Effects)



Application (Risk Map)



References

- Umlauf (2023). *softtrees: Soft Distributional Regression Trees and Forests*. https://github.com/freezenik/softtrees.
- Umlauf and Klein (2022). *Distributional Adaptive Soft Regression Trees*. Working paper, doi:10.48550/ARXIV.2210.10389.
- Umlauf, Klein, Simon, Zeileis (2021). *bamlss: A Lego Toolbox for Flexible Bayesian Regression (and Beyond)* . Journal of Statistical Software, doi:10.18637/jss.v100.i04.
- Schlosser, Hothorn, Stauffer and Zeileis (2019). *Distributional Regression Forests for Probabilistic Precipitation Forecasting in Complex Terrain*. The Annals of Applied Statistics, doi:10.1214/19-AOAS1247.
- Umlauf, Klein, and Zeileis (2018). *BAMLSS: Bayesian Additive Models for Location, Scale and Shape (and Beyond)*. Journal of Computational and Graphical Statistics, doi:10.1080/10618600.2017.1407325.