Freie Universität Berlin

A boosting algorithm for Generalized Extreme Value distributions

Madlen Peter, Alexander Pasternack, Henning W. Rust

Motivation:

- In weather and climate science statistical modeling is applied for manifold problems.
- It is often meaningful to apply model selection approaches, to avoid overfitting.
- Here, the boosting approach, combines model selection and parameter estimation.
- Boosting has been originally developed for classification problems but has also been extended and used for other applications; i.a. non-homogeneous gaussian regression.

Goal:

- Based on the non-homogeneous boosting (Messner et al. (2016)) we develop a boosting algorithm for a non-stationary Generalized Extreme Value distribution (GEV).
- Most relevant predictor variables for location, scale and shape parameter should be identified.

Method:

- We apply this algorithm to various toy model simulations to assess the effect of this novel approach.

Boosting in general

Freie Universität Berlin

- Boosting iteratively increases model coefficients.
- Most relevant parameters are increased first.
- Best set of coeff. can be found by cross validation (CV).
- Thus, not relevant parameters are zero.

- Origin: boosting for classification.

- Adoption to Non-Homogeneous Gaussian Regression.
- E.g . $f^{Cal}(t,\tau) = \mathcal{N}(\alpha + \beta\mu(t,\tau), (\exp(\gamma + \delta\sigma(t,\tau)))^2))$.
- With: $\alpha(t,\tau) = \sum_{l=0}^{6}(a_{2l} + a_{(2l+1)}t)\tau^l$ rest analog.

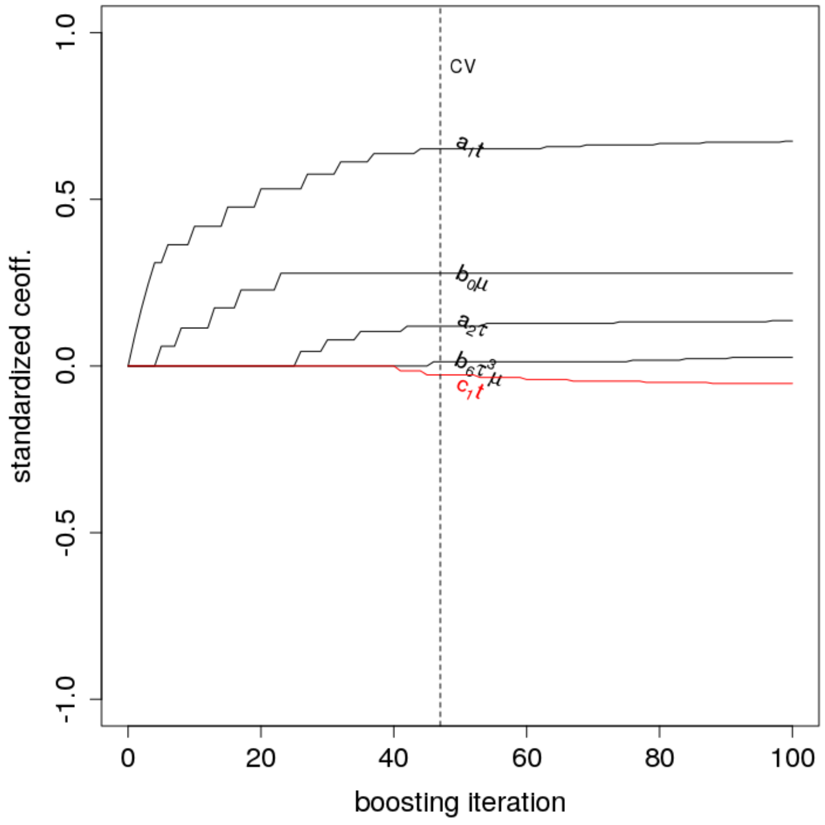

Fig. 1: Exemplary iteration of model coefficients with Non-Homogeneous boosting. This plot was generated with R-package CRCH.

Non-homogenous Boosting

Freie Universität Berlin

Schematic overview of the iteration algorithm of Non-homogenous Boosting:

1) Initial values for $\mu_1, \mu_2, ..\quad \sigma_1, \sigma_2, ..$

2) Derivative of the loss function $\frac{\partial \mathrm{nllh}}{\partial \mu} \frac{\partial \mathrm{nllh}}{\partial \sigma}$

For μ

3) best = argmax cor($\frac{\partial \mathrm{nllh}}{\partial \mu}$, $\boldsymbol{X}$)

4) stepsize s = $\mathrm{cor}_{\mathrm{best}} \cdot \nu$

5) $\mu_{\mathrm{best,current}} = \mu_{\mathrm{best},old} + \mathrm{s}$

6) log-Likelihood $L_\mu(y, \theta)$

For σ

3) best = argmax cor($\frac{\partial \mathrm{nllh}}{\partial \sigma}$, $\boldsymbol{X}$)

4) stepsize s = $\mathrm{cor}_{\mathrm{best}} \cdot \nu$

5) $\sigma_{\mathrm{best,current}} = \sigma_{\mathrm{best},old} + \mathrm{s}$

6) log-Likelihood $L_\sigma(y, \theta)$

$$L_\mu(y,\theta) > L_\sigma(y,\theta):\ \mu_{\mathrm{best,new}} = \mu_{\mathrm{best,current}}$$
$$L_\mu(y,\theta) < L_\sigma(y,\theta):\ \sigma_{\mathrm{best,new}} = \sigma_{\mathrm{best,current}}$$

→updating μ or σ per iteration !

- nllh: negative log-likelihood for every time step.
- $\mu_1, \sigma_1, \ldots$: predictor coefficients.
- $\boldsymbol{X}$: model matrix
- ν: Learning rate
- cor: correlation coeff.

GEV-Boosting - Preparations

Freie Universität Berlin

Log-Likelihood

$$L(\mu,\sigma,\xi) = \sum_{i=1}^{N} \log(f(y_i|\mu,\sigma,\xi)) \quad \longrightarrow \quad l(\mu,\sigma,\xi) = f(y_i|\mu,\sigma,\xi)$$

with:

$$\mu(t) = \boldsymbol{X}^T m$$
$$\sigma(t) = \boldsymbol{Y}^T s$$
$$\xi(t) = \boldsymbol{Z}^T u$$

Calculate first derivative

$$\frac{\partial l}{\partial \mu}, \frac{\partial l}{\partial \sigma}, \frac{\partial l}{\partial \xi}$$

-Set first iteration $i = 1$

-Predefine step size ν

Set initial coefficient values for iteration $i = 1$

$$m_{i=1,j} = m_{1,1}, \dots, m_{1,J} \quad s_{i=1,k} = s_1, \dots, s_K \quad u_{i=1,b} = u_1, \dots, u_B$$

GEV-Boosting - Iteration

Freie Universität Berlin

for μ:

Regression for each pred. j:
$$\frac{\partial l}{\partial \mu}(t) \sim \alpha_j X_j(t)$$

Find best performing predictor j for $\frac{\partial l}{\partial \mu}(t)$, e.g. with R^2, rename it j^*

Update:
$$m^* = m_i$$
$$m^*{}_{j^*} = m_{i-1,j^*} + \nu\alpha_{j^*}$$

for σ:

Regression for each pred. k:
$$\frac{\partial l}{\partial \sigma}(t) \sim \beta_k Y_k(t)$$

Find best performing predictor k for $\frac{\partial l}{\partial \sigma}(t)$, rename it k^*

Update:
$$s^* = s_i$$
$$s^*{}_{k^*} = s_{i-1,k^*} + \nu\beta_{k^*}$$

for ξ:

Regression for each pred. b:
$$\frac{\partial l}{\partial \xi}(t) \sim \gamma_b Z_b(t)$$

Find best performing predictor b for $\frac{\partial l}{\partial \xi}(t)$, rename it b^*

Update:
$$u^* = u_i$$
$$u^*{}_{b^*} = u_{i-1,b^*} + \nu\gamma_{b^*}$$

If $L(\boldsymbol{X}^T m^*, \boldsymbol{Y}^T s, \boldsymbol{Z}^T u) > L(\boldsymbol{X}^T m, \boldsymbol{Y}^T s^*, \boldsymbol{Z}^T u)$ & $> L(\boldsymbol{X}^T m, \boldsymbol{Y}^T s, \boldsymbol{Z}^T u^*)$
Update $m_i = m^*$ or analog s_i or u_i

$$i = i + 1$$

Apply GEV-Boosting to a toy model

Freie Universität Berlin

1. Generate some toy model Observations

$$O(t) \sim GEV(\mu_o(t), \sigma_o(t), \xi_o)$$

with

$$\mu_o(t) = 10.2 + 50t$$
$$\sigma_o(t) = 5.5 + 10t$$
$$\xi_o = 0.1$$

2. Model assumption

$$O(t) \sim GEV(\mu_m(t), \sigma_m(t), \xi_m)$$

with

$$\mu_m(t) = m_1 + m_1 t + m_2 t^2$$
$$\sigma_m(t) = s_1 + s_1 t + s_2 t^2$$
$$\xi_m(t) = u_1 + u_1 t + u_2 t^2$$

Toy model observation:

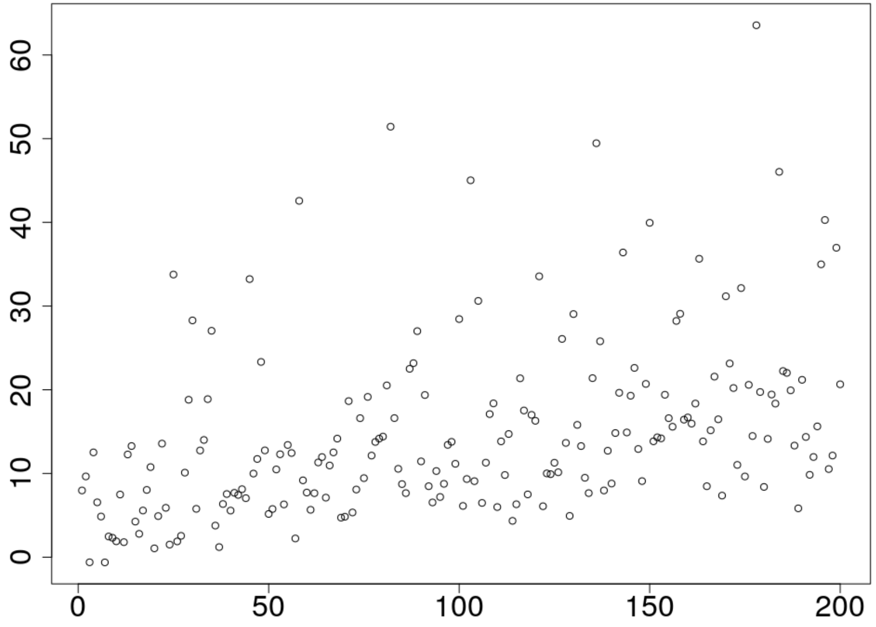

Fig. 2: Pseudo-observation over 200 time steps generated by the toy model.

Result

Freie Universität Berlin

Example 1:

Toy model:

$$\mu_o(t) = 10.2 + 50t$$

$$\sigma_o(t) = 5.5 + 10t$$

$$\xi_o = 0.1$$

After 1000 Iteration-Steps:

$$\mu_m(t) = 11 + 42.1t - 2t^2$$

$$\sigma_m(t) = 6.7 + 0t + 5.5t^2$$

$$\xi_m = 0.12 + 0t + 0t^2$$

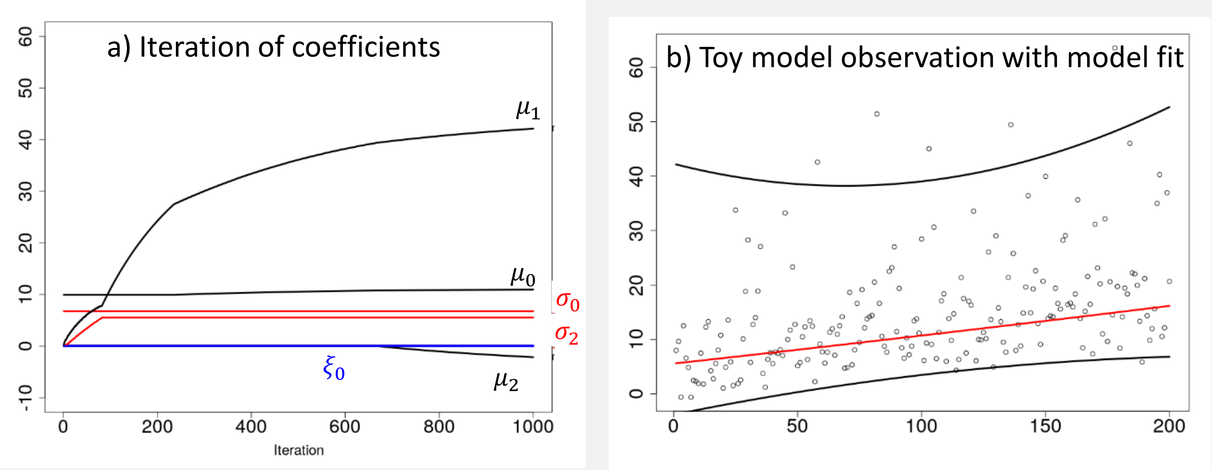

Fig. 3: a) Iteration of the toy model coefficients of example 1 with boosting (black line refers to location, red lines to scale and blue lines to shape coeff.). Fig. b) shows the corresponding prediction after 1000 iteration steps with the expected value (red line), the 95% prediction interval (black lines) and the corresponding pseudo-observations.

Example 2:

Toy model:

$$\mu_o(t) = 10.2 + 50t + 50t^2$$

$$\sigma_o(t) = 5.5 + 10t$$

$$\xi_o = 0.1$$

After 2000 Iteration-Steps:

$$\mu_m(t) = 10.5 + 50.9t + 49.6t^2$$

$$\sigma_m(t) = 6.7 + 11t + 0t^2$$

$$\xi_m = 0.89 - 0.4t + 0t^2$$

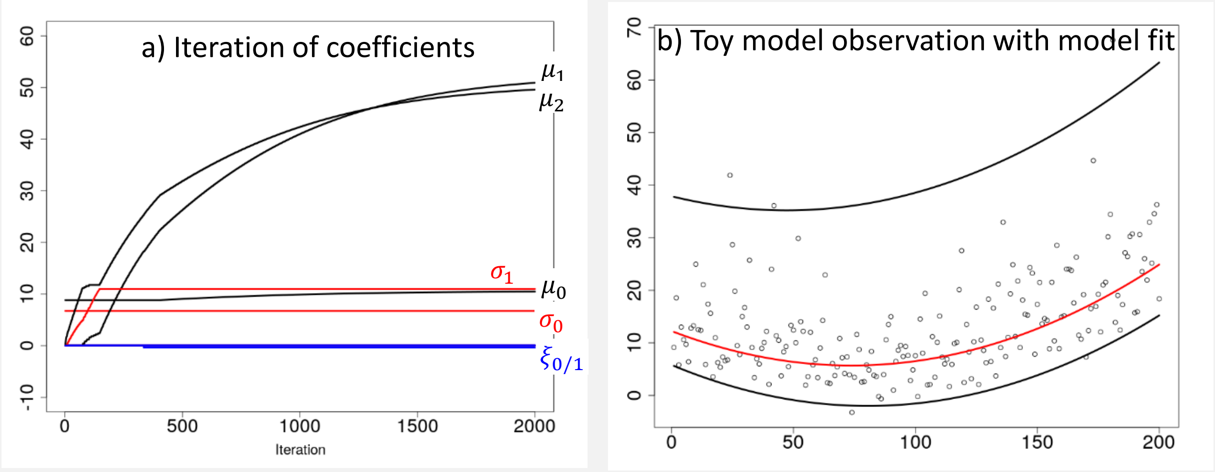

Fig. 4: a) Iteration of the toy model coefficients of example 2 with boosting (black line refers to location, red lines to scale and blue lines to shape coeff.). Fig. b) shows the corresponding prediction after 1000 iteration steps with the expected value (red line), the 95% prediction interval (black lines) and the corresponding pseudo-observations.

Summary

- **The work is still in progress, but …**

- Algorithm for GEV-Boosting works – but not very satisfying.
- GEV-Boosting increases the most relevant predictors first, but sometimes increases minor relevant ones.
- Much more iterations are needed, compared to Non-Homogeneous Boosting.
- Maybe, the presented toy model example is not suitable.

References:

- Messner JW, Zeileis A, Broecker J, Mayr GJ (2014). "Probabilistic Wind Power Forecasts with an Inverse Power Curve Transformation and Censored Regression." Wind Energy, 17(11), 1753–1766. doi: 10.1002/we.1666.
- Messner JW, Mayr GJ, Zeileis A (2016). "Heteroscedastic Censored and Truncated Regression with crch." The R Journal, 8(1), 173–181.
- Messner JW, Mayr GJ, Zeileis A (2017). "Non-Homogeneous Boosting for Predictor Selection in Ensemble Post-Processing." Monthly Weather Review, 145(1), 137–147. doi: 10.1175/MWR-D-16-0088.1.

madlen.peter@met.fu-berlin.de