# Introduction to Machine Learning

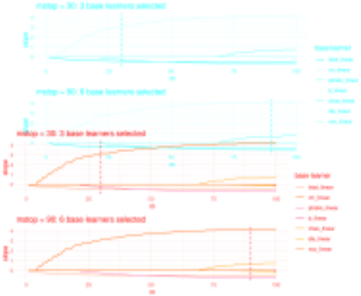
Boosting

Gradient Boosting: Advanced CWB

# Gradient Boosting: Advanced CWB

Learning goals

- Details of nonlinear BLs and splines
- Decomposition for splines
- Fair base learner selection
- Feature importance and PDPs



## Learning goals

- Details of nonlinear BLs and splines
- Decomposition for splines
- Fair base learner selection
- Feature importance and PDPs

# NONLINEAR EFFECT DECOMPOSITION

Kneib, Hothorn, and Tutz 2009 proposed a decomposition of each base learner into a constant, a linear and a nonlinear part. The boosting algorithm will automatically decide which feature to include – linear, nonlinear, or none at all:

$$
\begin{aligned}
b_j(x_j, \boldsymbol{\theta}^{[m]}) &= b_{j,\text{const}}(x_j, \boldsymbol{\theta}^{[m]}) + b_{j,\text{lin}}(x_j, \boldsymbol{\theta}^{[m]}) + b_{j,\text{nonlin}}(x_j, \boldsymbol{\theta}^{[m]}) \\
&= \theta^{[m]}_{\text{j,const}} + x_j \cdot \theta^{[m]}_{j,\text{lin}} + s_j(x_j, \boldsymbol{\theta}^{[m]}_{j,\text{nonlin}}),
\end{aligned}
$$

where

- $\theta_{\text{j,const}}$ is the intercept of feature $j$,
- $x_j \cdot \theta^{[m]}_{j,\text{lin}}$ is a feature-specific linear base learner, and
- $s_j(x_j, \boldsymbol{\theta}^{[m]}_{j,\text{nonlin}})$ is a (centered) nonlinear base learner capturing deviation from the linear effect

Careful: We usually also apply an orthogonalization procedure on top of this but skip technical details here.