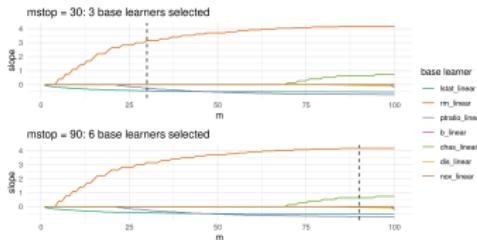


Introduction to Machine Learning

Boosting

Gradient Boosting: Advanced CWB



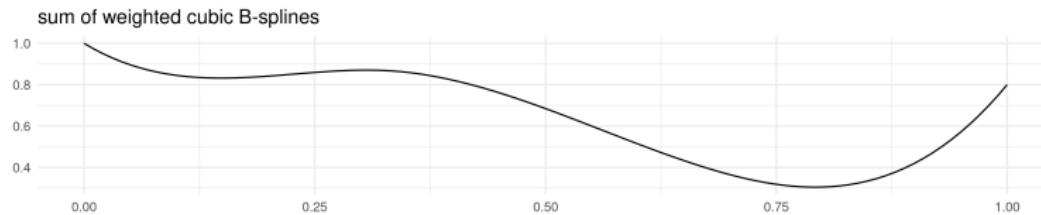
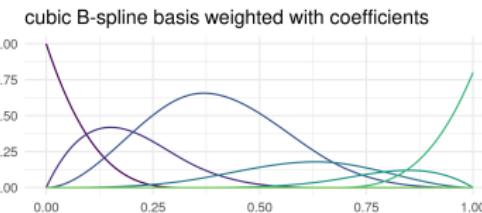
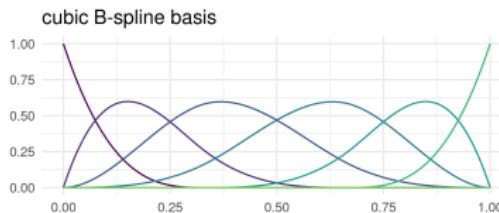
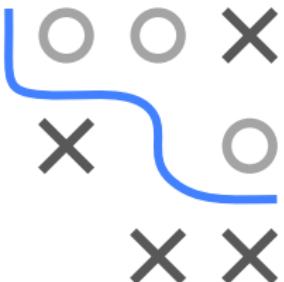
Learning goals

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

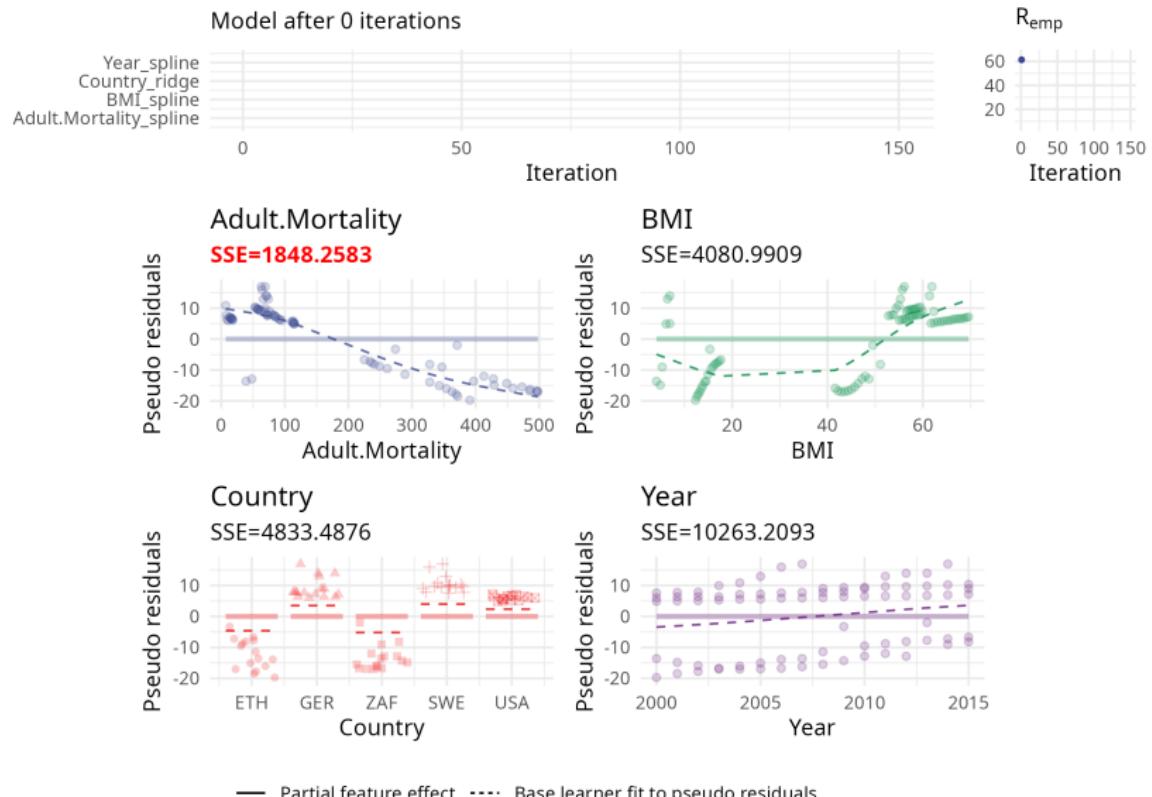


NONLINEAR BASE LEARNERS

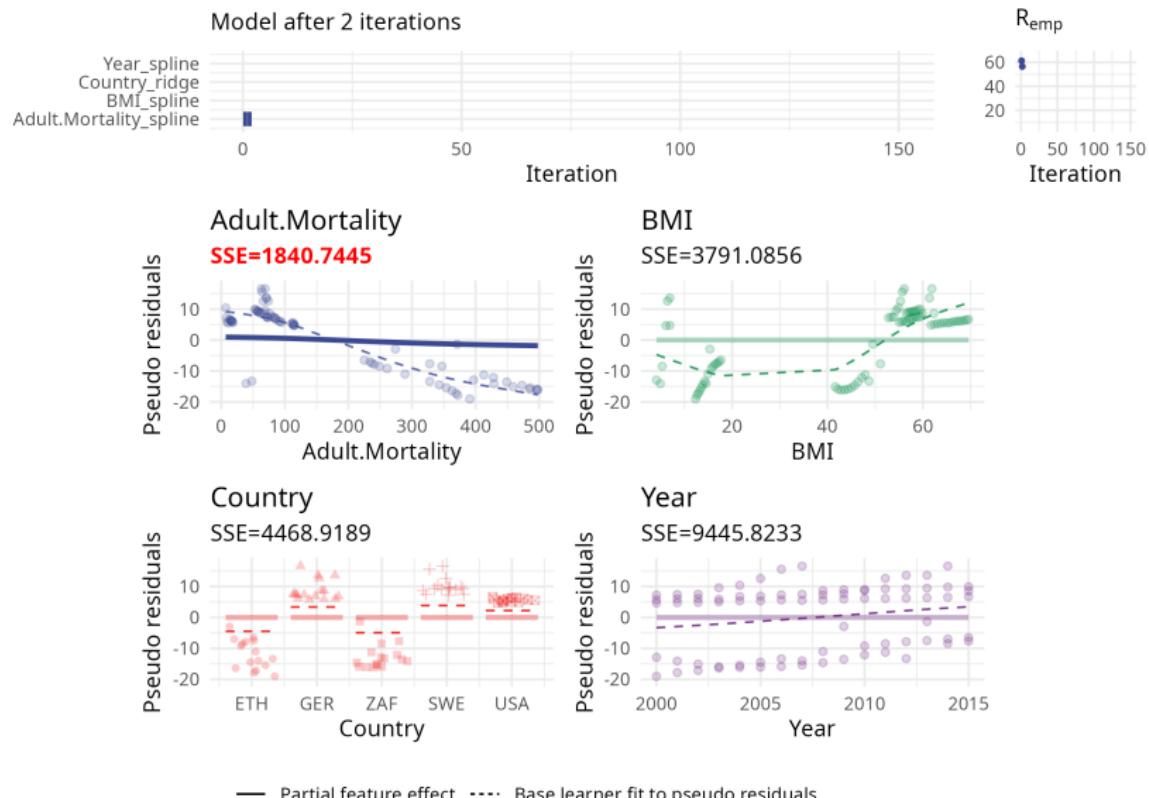
As an alternative we can use nonlinear base learners, such as P - or B -splines, which make the model equivalent to a **generalized additive model (GAM)** (as long as the base learners keep their additive structure, which is the case for splines).



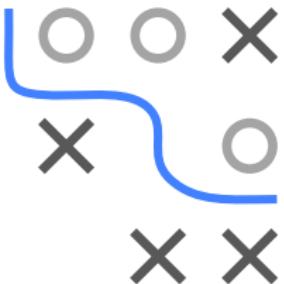
EXAMPLE: LIFE EXPECTANCY (NONLINEAR)



EXAMPLE: LIFE EXPECTANCY (NONLINEAR)



EXAMPLE: LIFE EXPECTANCY (NONLINEAR)



Model after 5 iterations

Year_spline
Country_ridge
BMI_spline
Adult.Mortality_spline

0

50

100

150

Iteration

R_{emp}

60
40
20

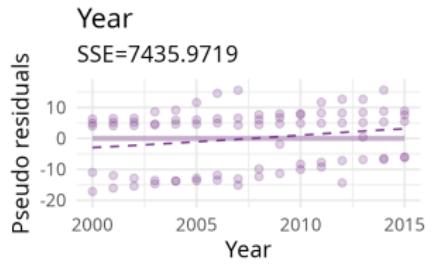
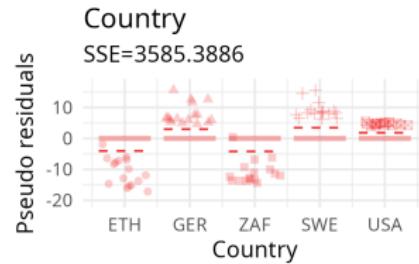
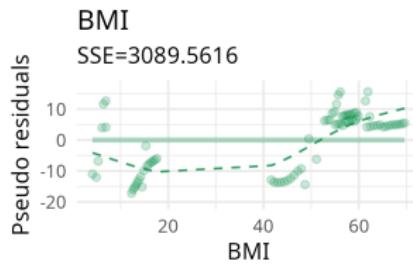
0

50

100

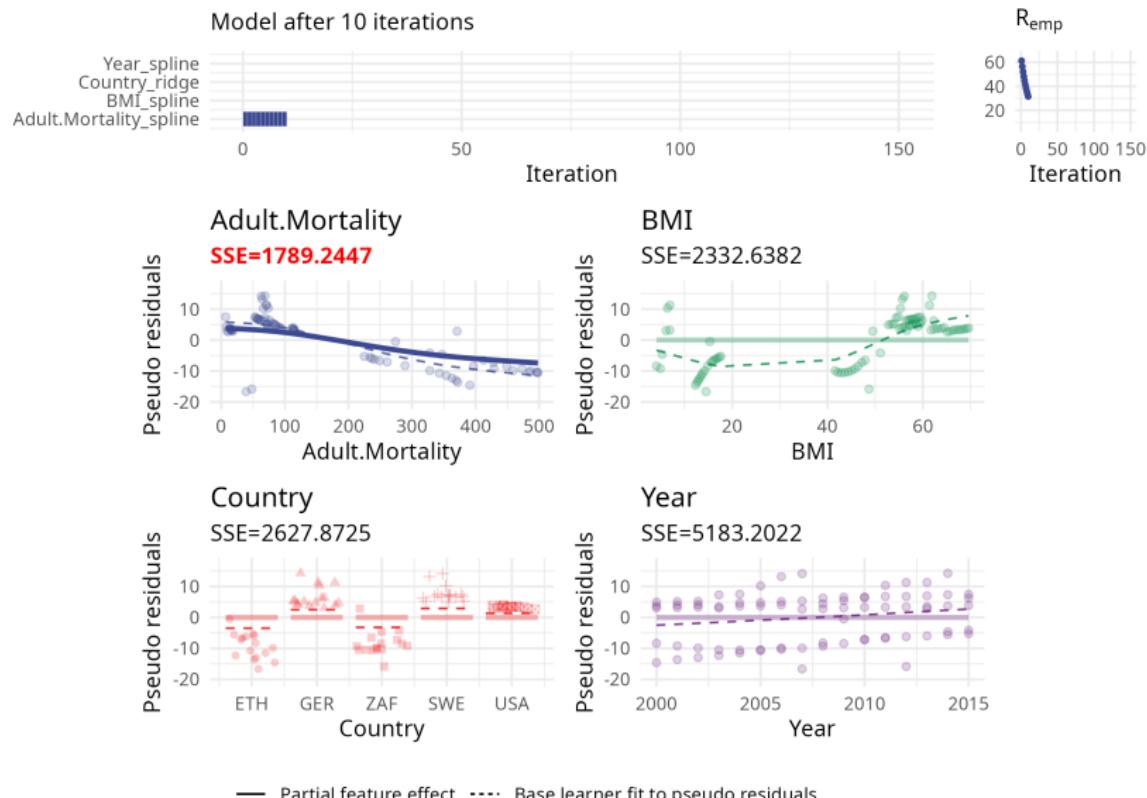
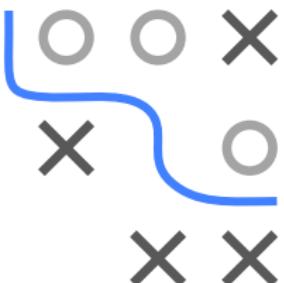
150

Iteration

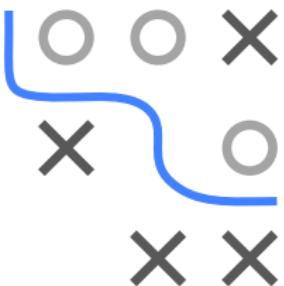


— Partial feature effect - - - Base learner fit to pseudo residuals

EXAMPLE: LIFE EXPECTANCY (NONLINEAR)



EXAMPLE: LIFE EXPECTANCY (NONLINEAR)

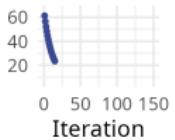


Model after 15 iterations

Year_spline
Country_ridge
BMI_spline
Adult.Mortality_spline

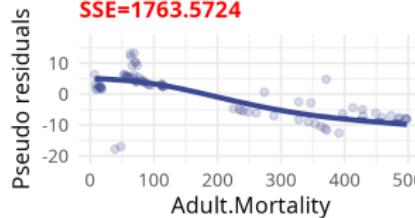


R_{emp}



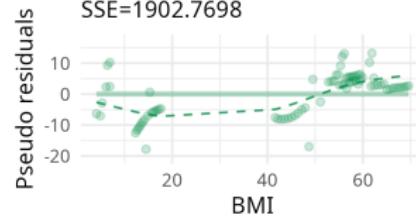
Adult.Mortality

SSE=1763.5724



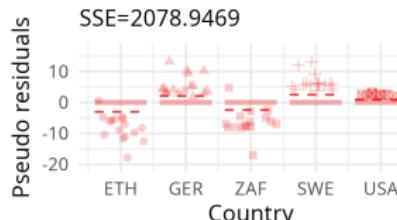
BMI

SSE=1902.7698



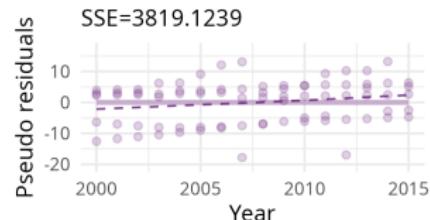
Country

SSE=2078.9469



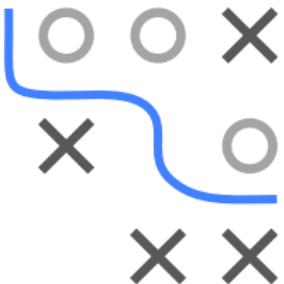
Year

SSE=3819.1239



— Partial feature effect - - - Base learner fit to pseudo residuals

EXAMPLE: LIFE EXPECTANCY (NONLINEAR)



Model after 17 iterations

Year_spline
Country_ridge
BMI_spline
Adult.Mortality_spline

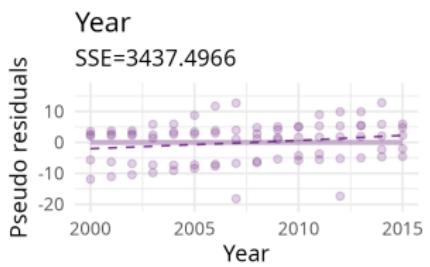
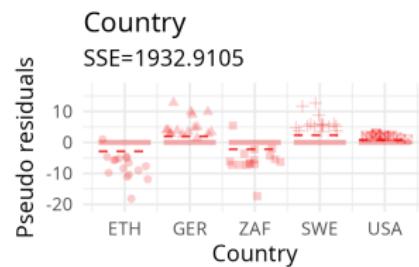
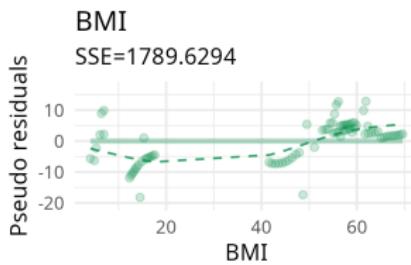
0 50 100 150

Iteration

R_{emp}

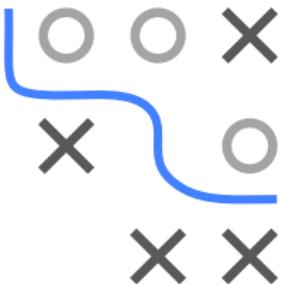
60
40
20
0 50 100 150

Iteration



— Partial feature effect - - - Base learner fit to pseudo residuals

EXAMPLE: LIFE EXPECTANCY (NONLINEAR)



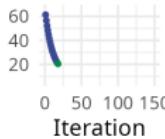
Model after 18 iterations

Year_spline
Country_ridge
BMI_spline
Adult.Mortality_spline



Iteration

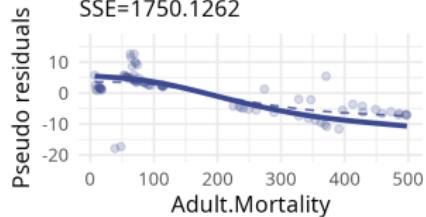
R_{emp}



Iteration

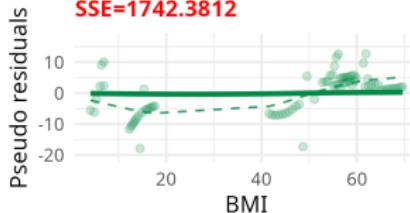
Adult.Mortality

SSE=1750.1262



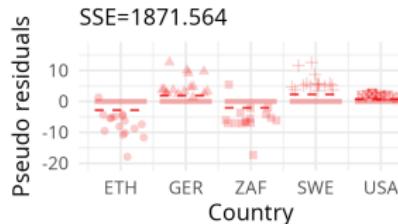
BMI

SSE=1742.3812



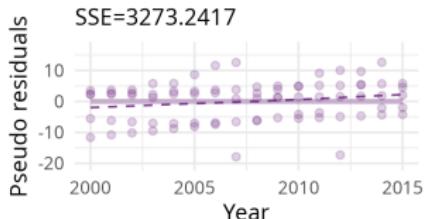
Country

SSE=1871.564



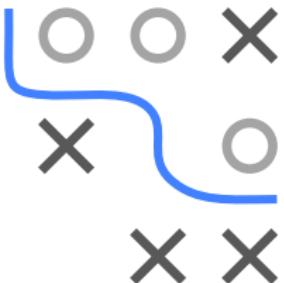
Year

SSE=3273.2417



— Partial feature effect - - - Base learner fit to pseudo residuals

EXAMPLE: LIFE EXPECTANCY (NONLINEAR)



Model after 20 iterations

Year_spline
Country_ridge
BMI_spline
Adult.Mortality_spline

0 50 100 150

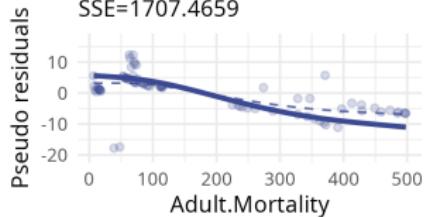
Iteration

R_{emp}

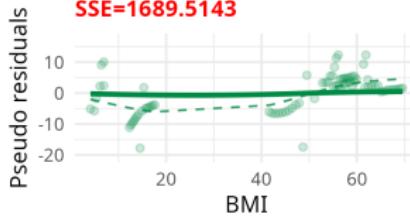
60
40
20
0 50 100 150

Iteration

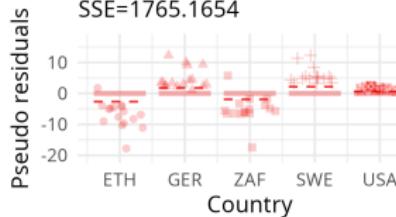
Adult.Mortality
 $SSE=1707.4659$



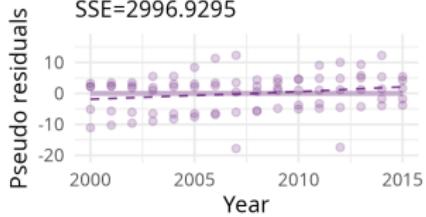
BMI
 $SSE=1689.5143$



Country
 $SSE=1765.1654$

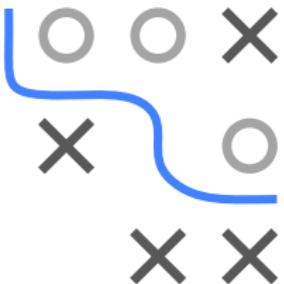


Year
 $SSE=2996.9295$



— Partial feature effect - - - Base learner fit to pseudo residuals

EXAMPLE: LIFE EXPECTANCY (NONLINEAR)

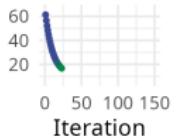


Model after 23 iterations

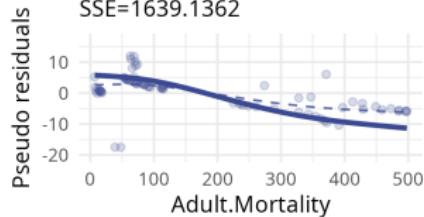
Year_spline
Country_ridge
BMI_spline
Adult.Mortality_spline

Iteration

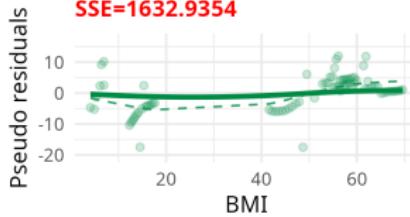
R_{emp}



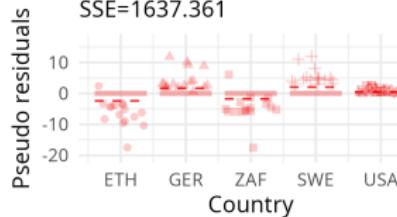
Adult.Mortality
 $SSE=1639.1362$



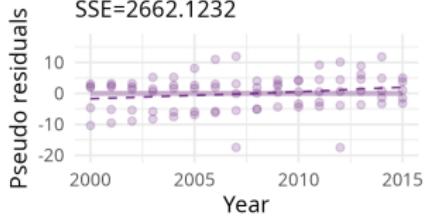
BMI
 $SSE=1632.9354$



Country
 $SSE=1637.361$

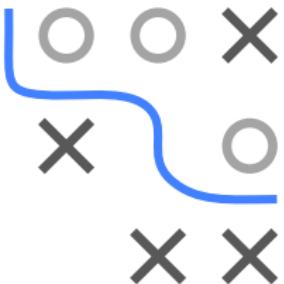


Year
 $SSE=2662.1232$



— Partial feature effect - - - Base learner fit to pseudo residuals

EXAMPLE: LIFE EXPECTANCY (NONLINEAR)



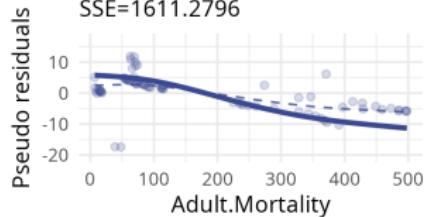
Model after 24 iterations

Year_spline
Country_ridge
BMI_spline
Adult.Mortality_spline



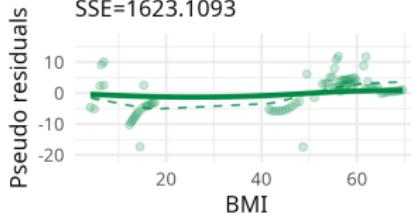
Adult.Mortality

SSE=1611.2796



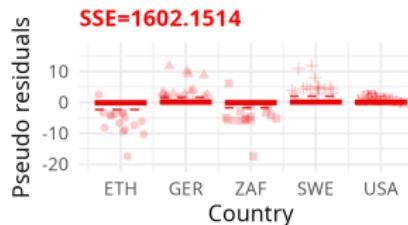
BMI

SSE=1623.1093



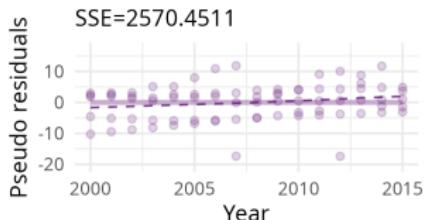
Country

SSE=1602.1514



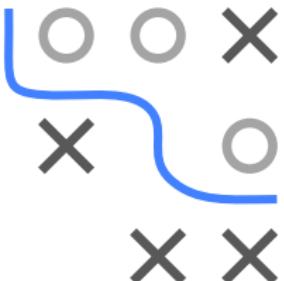
Year

SSE=2570.4511



— Partial feature effect - - - Base learner fit to pseudo residuals

EXAMPLE: LIFE EXPECTANCY (NONLINEAR)



Model after 30 iterations

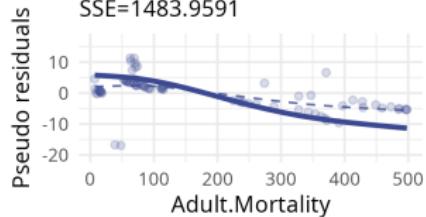
Year_spline
Country_ridge
BMI_spline
Adult.Mortality_spline



Iteration

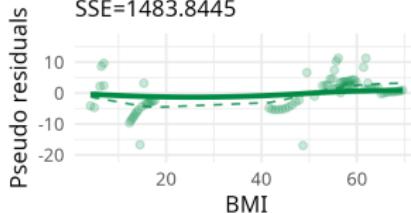
Adult.Mortality

SSE=1483.9591



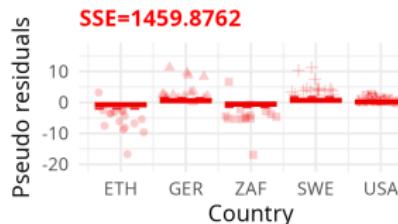
BMI

SSE=1483.8445



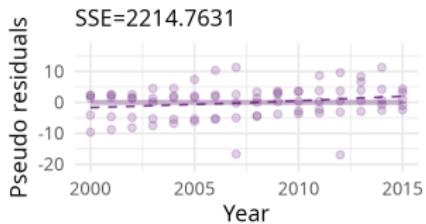
Country

SSE=1459.8762



Year

SSE=2214.7631



— Partial feature effect - - - Base learner fit to pseudo residuals

EXAMPLE: LIFE EXPECTANCY (NONLINEAR)



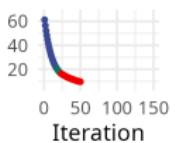
Model after 50 iterations

Year_spline
Country_ridge
BMI_spline
Adult.Mortality_spline



Iteration

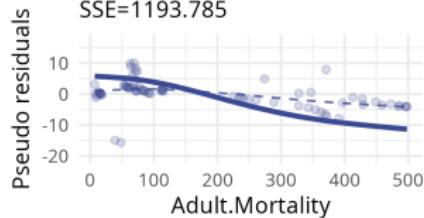
R_{emp}



Iteration

Adult.Mortality

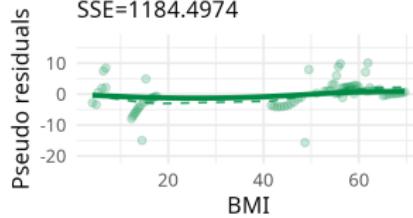
SSE=1193.785



Adult.Mortality

BMI

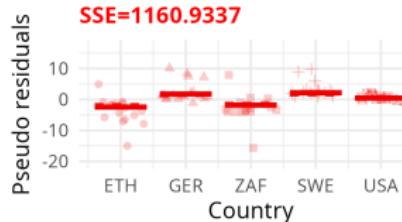
SSE=1184.4974



BMI

Country

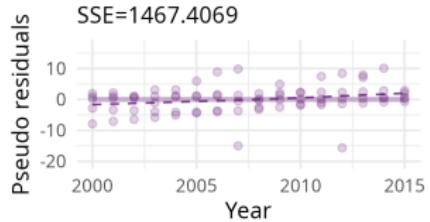
SSE=1160.9337



Country

Year

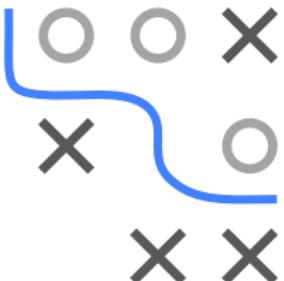
SSE=1467.4069



Year

— Partial feature effect - - - Base learner fit to pseudo residuals

EXAMPLE: LIFE EXPECTANCY (NONLINEAR)

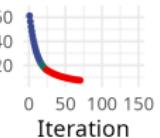


Model after 70 iterations

Year_spline
Country_ridge
BMI_spline
Adult.Mortality_spline

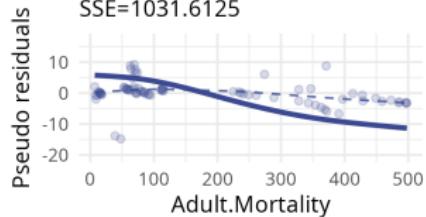


R_{emp}



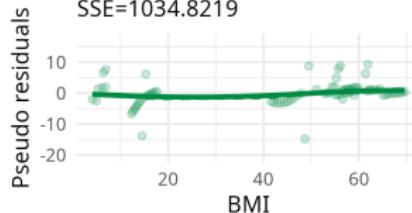
Adult.Mortality

SSE=1031.6125



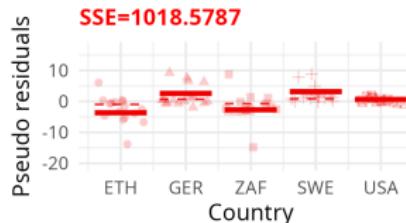
BMI

SSE=1034.8219



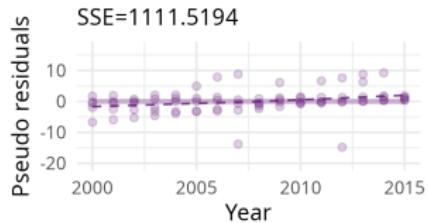
Country

SSE=1018.5787



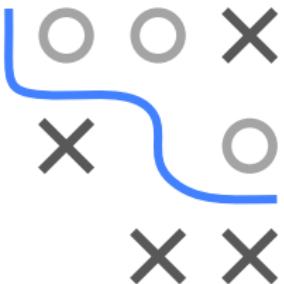
Year

SSE=1111.5194



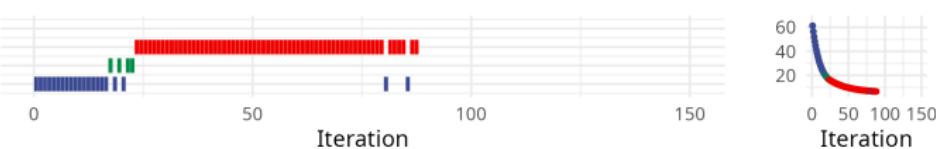
— Partial feature effect - - - Base learner fit to pseudo residuals

EXAMPLE: LIFE EXPECTANCY (NONLINEAR)

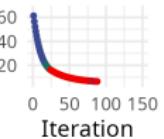


Model after 88 iterations

Year_spline
Country_ridge
BMI_spline
Adult.Mortality_spline

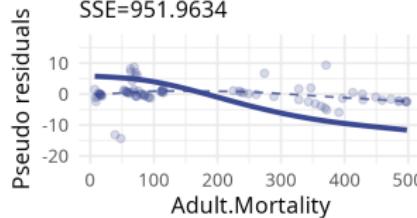


R_{emp}



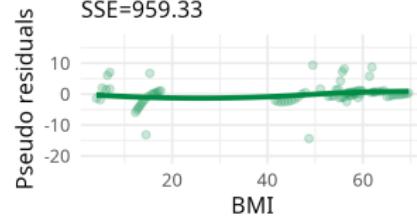
Adult.Mortality

SSE=951.9634



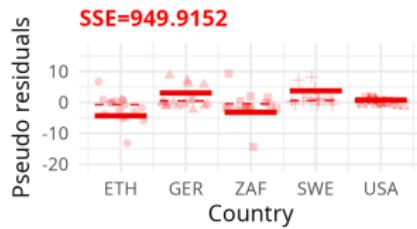
BMI

SSE=959.33



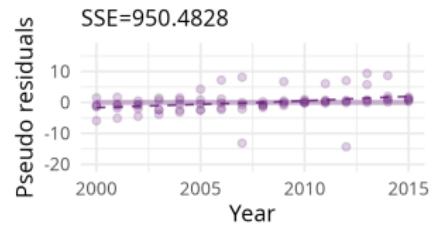
Country

SSE=949.9152



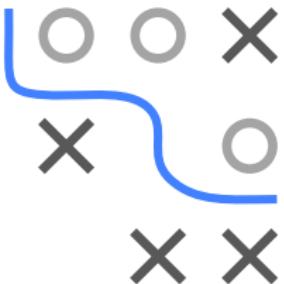
Year

SSE=950.4828



— Partial feature effect - - - Base learner fit to pseudo residuals

EXAMPLE: LIFE EXPECTANCY (NONLINEAR)

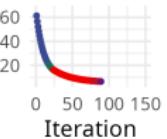


Model after 89 iterations

Year_spline
Country_ridge
BMI_spline
Adult.Mortality_spline

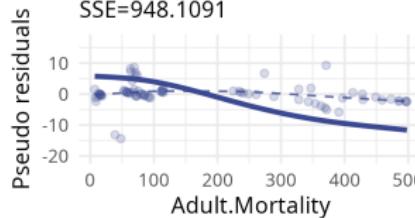


R_{emp}



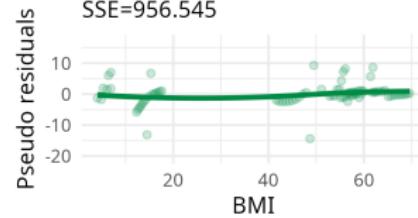
Adult.Mortality

SSE=948.1091



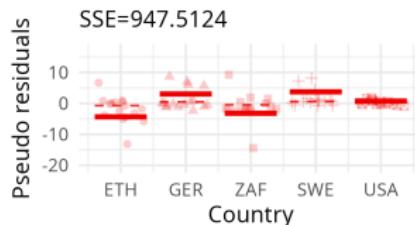
BMI

SSE=956.545



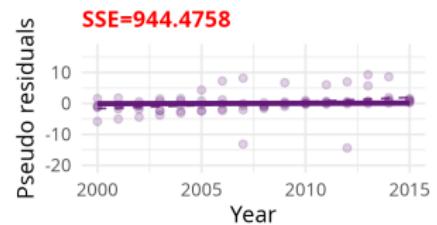
Country

SSE=947.5124



Year

SSE=944.4758



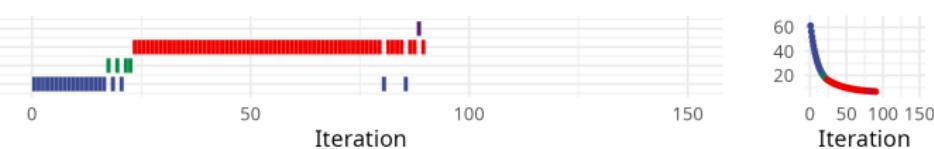
— Partial feature effect - - - Base learner fit to pseudo residuals

EXAMPLE: LIFE EXPECTANCY (NONLINEAR)

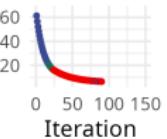


Model after 90 iterations

Year_spline
Country_ridge
BMI_spline
Adult.Mortality_spline

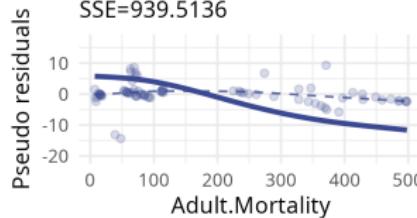


R_{emp}



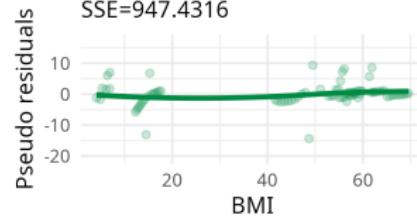
Adult.Mortality

SSE=939.5136



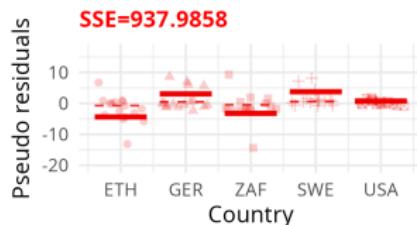
BMI

SSE=947.4316



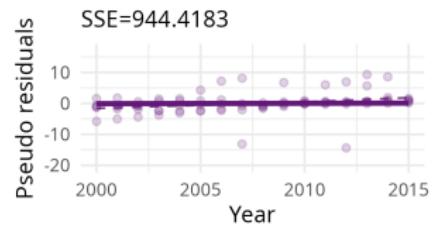
Country

SSE=937.9858



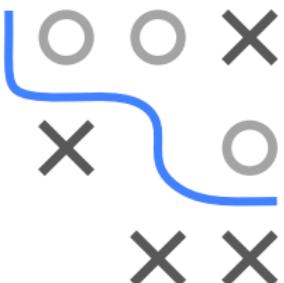
Year

SSE=944.4183



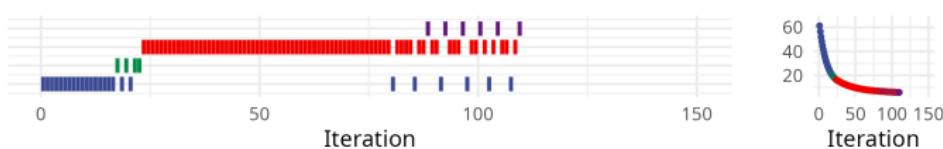
— Partial feature effect - - - Base learner fit to pseudo residuals

EXAMPLE: LIFE EXPECTANCY (NONLINEAR)

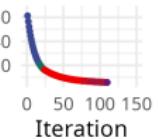


Model after 110 iterations

Year_spline
Country_ridge
BMI_spline
Adult.Mortality_spline

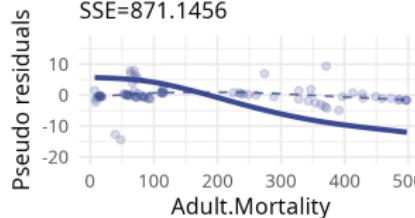


R_{emp}



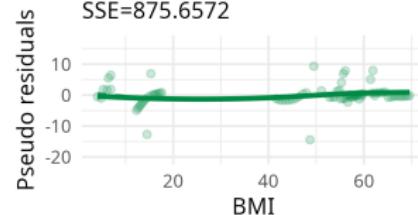
Adult.Mortality

SSE=871.1456



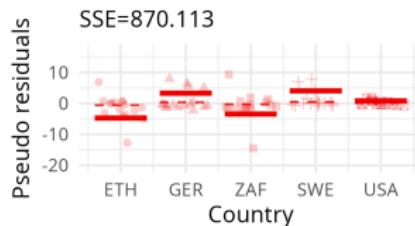
BMI

SSE=875.6572



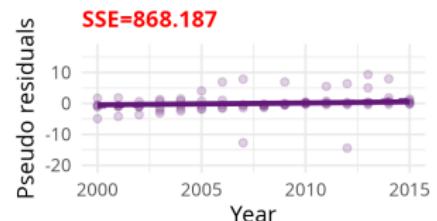
Country

SSE=870.113



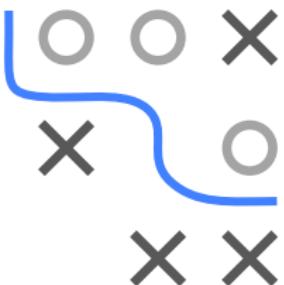
Year

SSE=868.187



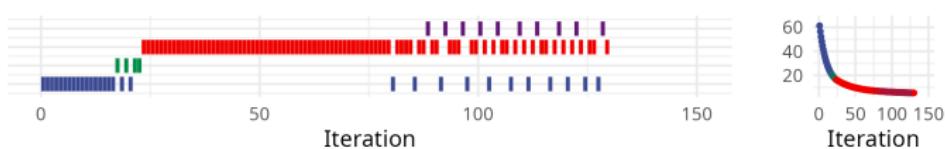
— Partial feature effect - - - Base learner fit to pseudo residuals

EXAMPLE: LIFE EXPECTANCY (NONLINEAR)

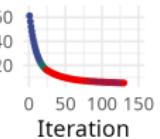


Model after 130 iterations

Year_spline
Country_ridge
BMI_spline
Adult.Mortality_spline

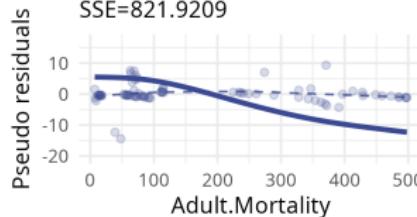


R_{emp}



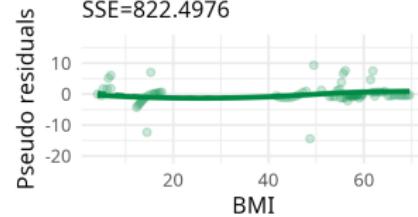
Adult.Mortality

SSE=821.9209



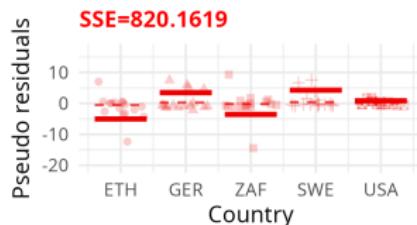
BMI

SSE=822.4976



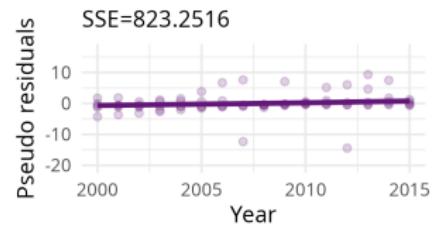
Country

SSE=820.1619



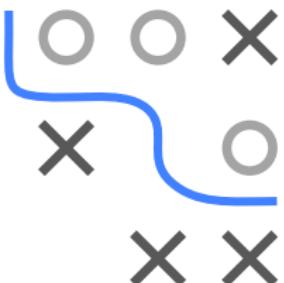
Year

SSE=823.2516



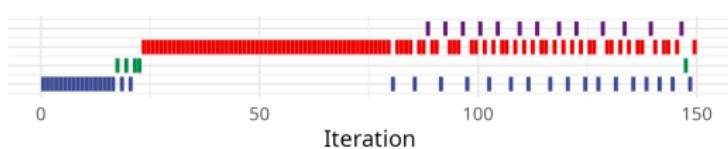
— Partial feature effect - - - Base learner fit to pseudo residuals

EXAMPLE: LIFE EXPECTANCY (NONLINEAR)

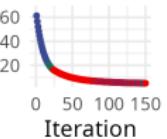


Model after 150 iterations

Year_spline
Country_ridge
BMI_spline
Adult.Mortality_spline

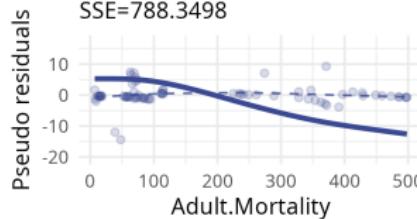


R_{emp}



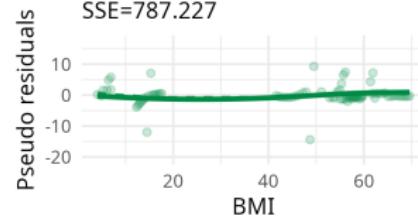
Adult.Mortality

SSE=788.3498



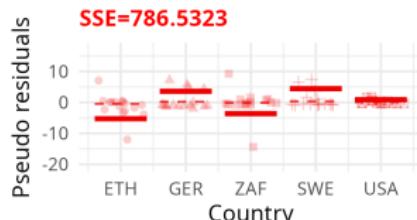
BMI

SSE=787.227



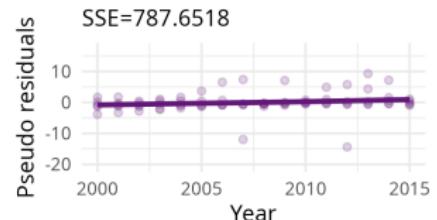
Country

SSE=786.5323



Year

SSE=787.6518



— Partial feature effect - - - Base learner fit to pseudo residuals

NONLINEAR EFFECT DECOMPOSITION

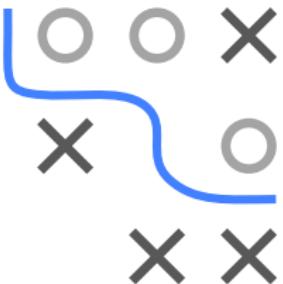
► Kneib, Hothorn, and Tutz 2009 proposed a decomposition of each base learner into a constant, a linear and a nonlinear part. The boosting algorithm will automatically decide which feature to include – linear, nonlinear, or none at all:

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

where

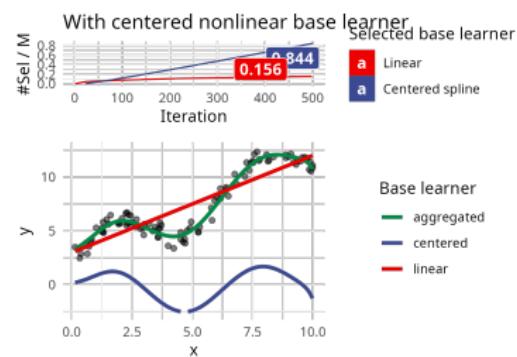
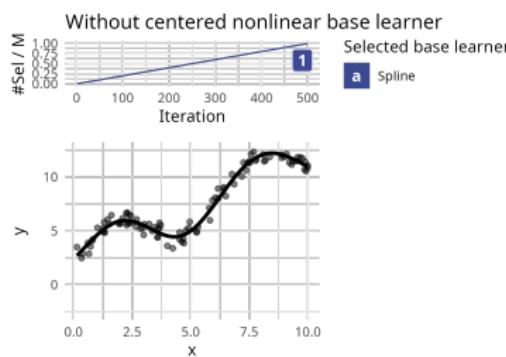
- $\theta_{j,\text{const}}$ is the intercept of feature j ,
- $x_j \cdot \theta_{j,\text{lin}}^{[m]}$ is a feature-specific linear base learner, and
- $s_j(x_j, \theta_{j,\text{nonlin}}^{[m]})$ is a (centered) nonlinear base learner capturing deviation from the linear effect

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



NONLINEAR EFFECT DECOMPOSITION

- Suppose $n = 100$ uniformly distributed x values between 0 and 10.
- The response $y = 2 \sin(x) + x + 2 + \varepsilon$ has a nonlinear and linear component ($\varepsilon \sim \mathcal{N}(0, \frac{1}{2})$).
- We apply CWB with $M = 500$ to $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$ with:
 - One model with $\mathcal{B} = \{b_{j,\text{lin}}, b_{j,\text{nonlin}}\}$
 - One model with $\mathcal{B} = \{b_{j,\text{lin}}, b_{j,\text{nonlin}}^c\}$



FAIR BASE LEARNER SELECTION

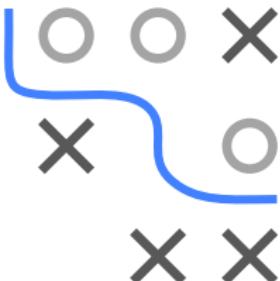
- Using splines and linear base learners in CWB will favor the more complex spline BLs over the linear BLs
- This makes it harder to achieve the desired behavior of the base learner decomposition as explained previously
- To conduct a fair base learner selection, we set the degrees of freedom of all base learners equal
- The idea is to set a single learner's regularization/penalty term so that their complexity is treated equally
- We also skip some technical details here



AVAILABLE BASE LEARNERS

There is a large number of possible base learners, e.g.:

- Linear effects and interactions (with or without intercept)
- Uni- or multivariate splines and tensor product splines
- Trees
- Random effects and Markov random fields
- Effects of functional covariates
- ...



In combination with the flexible choice of loss functions, boosting can be applied to fit a huge class of models.

Recent extensions include distributional regression (GAMLSS), where multiple additive predictors are boosted to model all distributional parameters (e.g., cond. mean and variance for a Gaussian model).

PARTIAL DEPENDENCE PLOTS

If we use single features in base learners, we consider each BL as a wrapper around a feature representing the feature's effect on the target. BLs can be selected more than once (with varying parameter estimates), signaling that this feature is more important.

E.g. let $j \in \{1, 2, 3\}$, the first three iterations might look as follows

$$m = 1 : \hat{f}^{[1]}(\mathbf{x}) = \hat{f}^{[0]} + \alpha \hat{b}_2(x_2, \hat{\theta}^{[1]})$$

$$m = 2 : \hat{f}^{[2]}(\mathbf{x}) = \hat{f}^{[1]} + \alpha \hat{b}_3(x_3, \hat{\theta}^{[2]})$$

$$m = 3 : \hat{f}^{[3]}(\mathbf{x}) = \hat{f}^{[2]} + \alpha \hat{b}_2(x_2, \hat{\theta}^{[3]})$$

Due to linearity, \hat{b}_2 base learners can be aggregated:

$$\hat{f}^{[3]}(\mathbf{x}) = \hat{f}^{[0]} + \alpha(\hat{b}_2(x_2, \hat{\theta}^{[1]}) + \hat{b}_3(x_3, \hat{\theta}^{[2]}))$$

Which is equivalent to: $\hat{f}^{[3]}(\mathbf{x}) = \hat{f}_0 + \hat{f}_2(x_2) + \hat{f}_3(x_3)$.

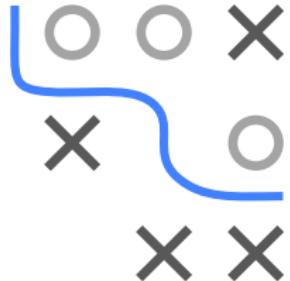
Hence, \hat{f} can be decomposed into the marginal feature effects (PDPs).



FEATURE IMPORTANCE

- We can further exploit the additive structure of the boosted ensemble to compute measures of **variable importance**.
- To this end, we simply sum for each feature x_j the improvements in empirical risk achieved over all iterations until $1 < m_{\text{stop}} \leq M$:

$$VI_j = \sum_{m=1}^{m_{\text{stop}}} \left(\mathcal{R}_{\text{emp}} \left(f^{[m-1]}(\mathbf{x}) \right) - \mathcal{R}_{\text{emp}} \left(f^{[m]}(\mathbf{x}) \right) \right) \cdot \mathbb{I}_{[j \in j^{[m]}]},$$



TAKE-HOME MESSAGE

- Componentwise gradient boosting is the statistical re-interpretation of gradient boosting
- We can fit a large number of statistical models, even in high dimensions ($p \gg n$)
- A drawback compared to statistical models is that we do not get valid inference for coefficients → post-selection inference
- In most cases, gradient boosting with trees will dominate componentwise boosting in terms of performance due to its inherent ability to include higher-order interaction terms

