Model-based Boosting with Variable Selection

Benjamin Hofner

Institut für Medizininformatik, Biometrie und Epidemiologie
Friedrich-Alexander-Universität Erlangen-Nürnberg

gmds 2014

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
MEDIZINISCHE FAKULTÄT

Overview

Part 1 Boosting for GLMs
Variable Selection in Linear Models

Part 2 Boosting for GAMs and STARs
A Journey to Unbiased Variable Selection and Model Choice

Part 3 Biomarker Discovery
Controlling False Discoveries in High-dimensional Situations

Part 1: Boosting for GLMs

Variable Selection in Linear Models

Gene Expression for Diffuse Large-B-Cell Lymphoma

- **Aim:** Predict survival outcome after chemotherapy based on the gene-expression profiles of lymphoma samples.
- **Data:**
  - Source: Rosenwald et al. (2002)
  - 222 patients with diffuse large-B-cell lymphoma
  - 7399 genes per patient

- Find a sparse risk profile using a linear Cox model.

Aims

- Fitting models for (potentially) high-dimensional data sets
- Good prediction performance should be achieved
- Resulting models should be interpretable
- Variable selection, i.e., only relevant covariates should be included

- One solution to all of this: Component-wise Boosting

Model Fitting

- Model fitting, in general, aims at **minimizing the expected loss** with appropriate loss function ρ, e.g.,

squared error loss	for Gaussian models $\rho(y, \boldsymbol{x}^\top \boldsymbol{\beta}) = (y - \boldsymbol{x}^\top \boldsymbol{\beta})^2$
negative log-likelihood	for GLMs
negative partial log-likelihood	for Cox models

- In practice: Minimization of the **empirical risk**

$$\frac{1}{n} \sum_{i=1}^{n} \rho(y_i, \boldsymbol{x}_i^\top \boldsymbol{\beta})$$

Model Fitting via Boosting

Boosting

- **minimizes the empirical risk** (e.g., negative log likelihood)
- via functional **gradient descent** (FGD)[1]
  (▶ steepest descent)
- in a **stagewise** fashion.

[1] Note: There are other flavors of boosting. Here we focus on FGD boosting.

Details on

Component-wise FGD Boosting

1. Set $m := 0$ and initialize estimate $\hat{\eta}^{[m]}$, where $\eta = \boldsymbol{x}^\top \boldsymbol{\beta}$ ($=$ linear predictor).
2. Increase iteration $m := m + 1$

...

4. Compute **negative gradient** of the loss function:

$$u_i^{[m]} = - \left. \frac{\partial \rho(y_i, \eta)}{\partial \eta} \right|_{\eta = \hat{\eta}^{[m-1]}(\boldsymbol{x}_i)}, \; i = 1, \ldots, n$$

5. **Estimate** the negative gradient vector $\boldsymbol{u}^{[m]} = (u_1^{[m]}, \ldots, u_n^{[m]})$ w.r.t. $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_J$ **separately**[2] by real-valued base-learners $g_j(\cdot)$:

$$\mathbf{u}^{[m]} = \hat{g}_j^{[m]}(\boldsymbol{x}_j) + \varepsilon_j, \qquad j = 1, \ldots, J$$

6. **Selection:** Chose best fitting base-learner $g_{j^*}(\cdot)$ with respect to some criterion (typically minimal RSS):

$$j^* = \underset{1 \leq j \leq p}{\operatorname{argmin}} \sum_{i=1}^{n} (u_i^{[m]} - \hat{g}_j^{[m]}(x_{ij}))^2.$$

[2] e.g. by separate simple linear regression models, one per base-learner:
`lm(u ~ x_j)`
Here, $g_j(\boldsymbol{x}_j) = \beta_j \boldsymbol{x}_j$

...

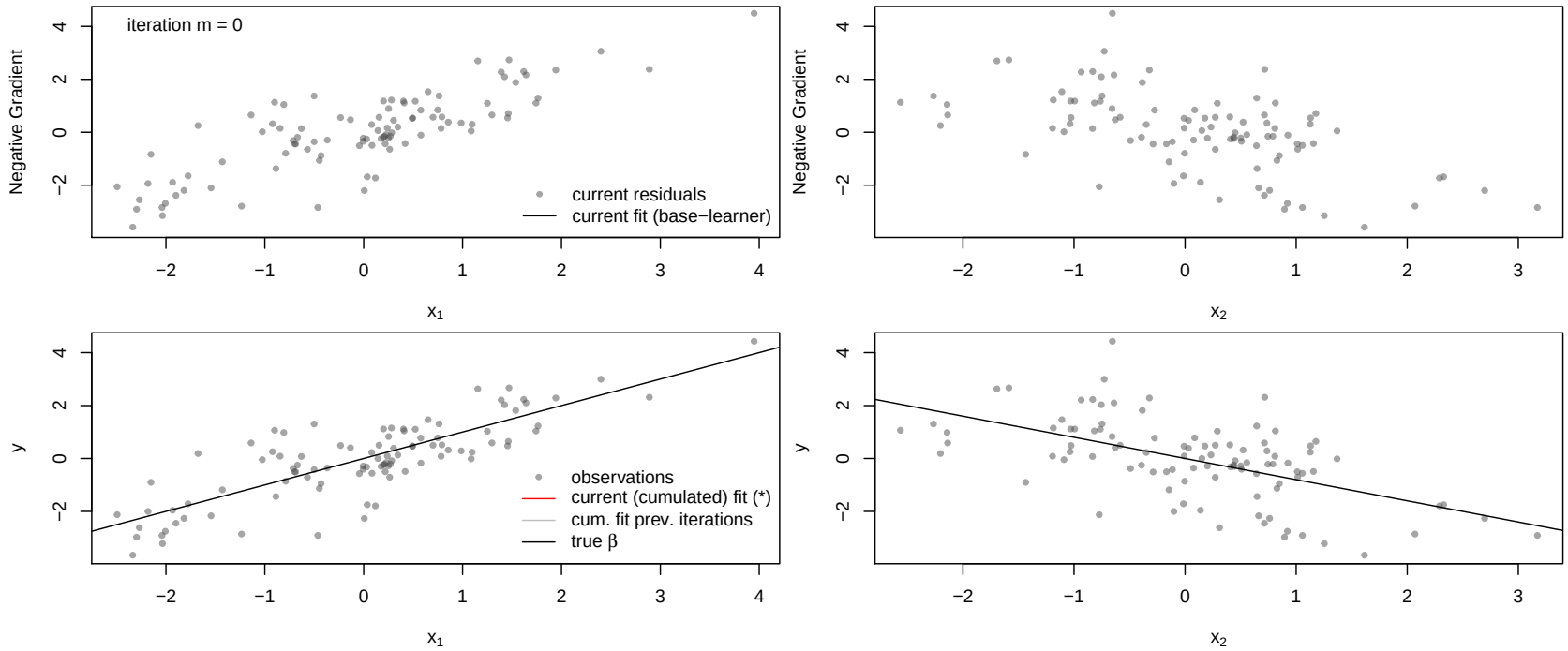
6. Compute the **update** for the additive predictor

$$\hat{\eta}^{[m]} = \hat{\eta}^{[m-1]} + \nu \cdot \hat{g}_{\mathbf{j}^*}^{[m]}$$

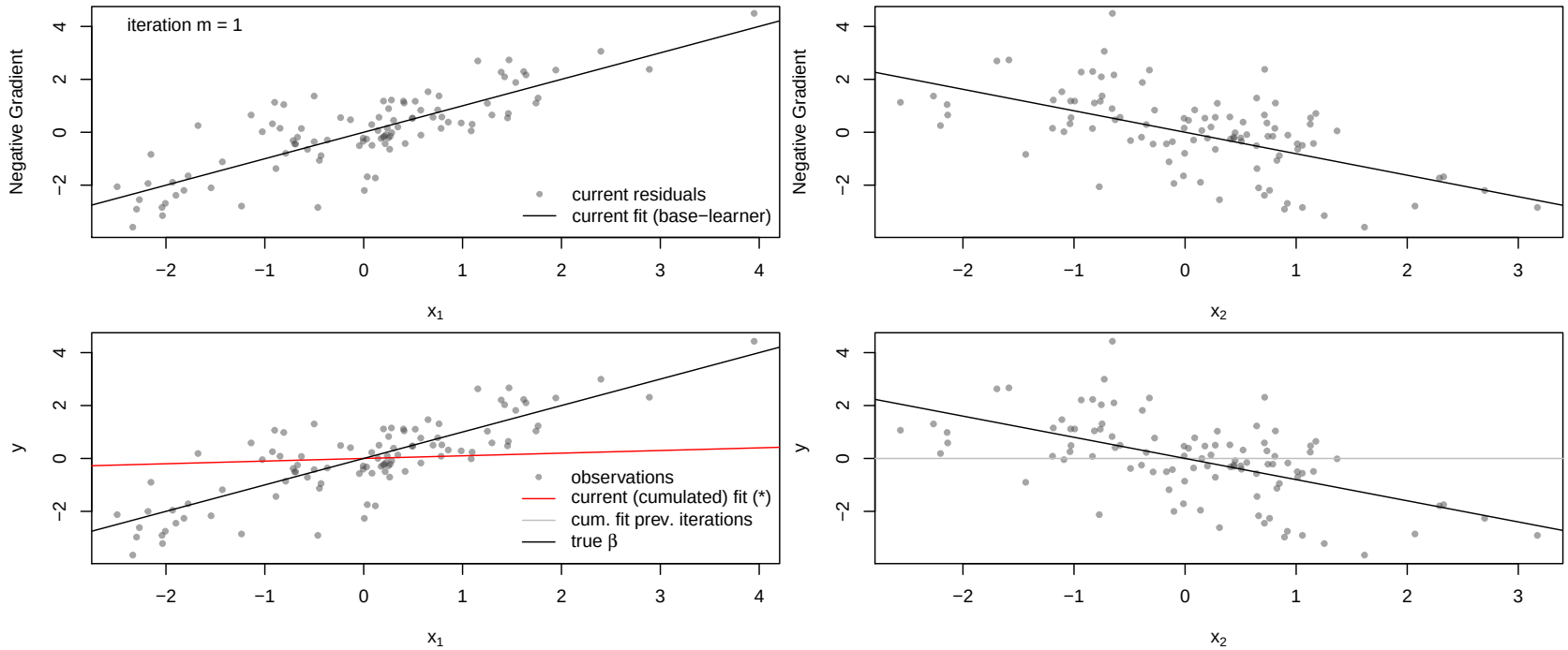
   with step-length factor $0 < \nu \leq 1$ (typically $\nu = 0.1$).
7. Continue iterating steps (2) to (6) until $m = m_{\text{stop}}$ for a given stopping iteration m_{stop}.

In Short

Boosting in Graphs



Boosting in Graphs

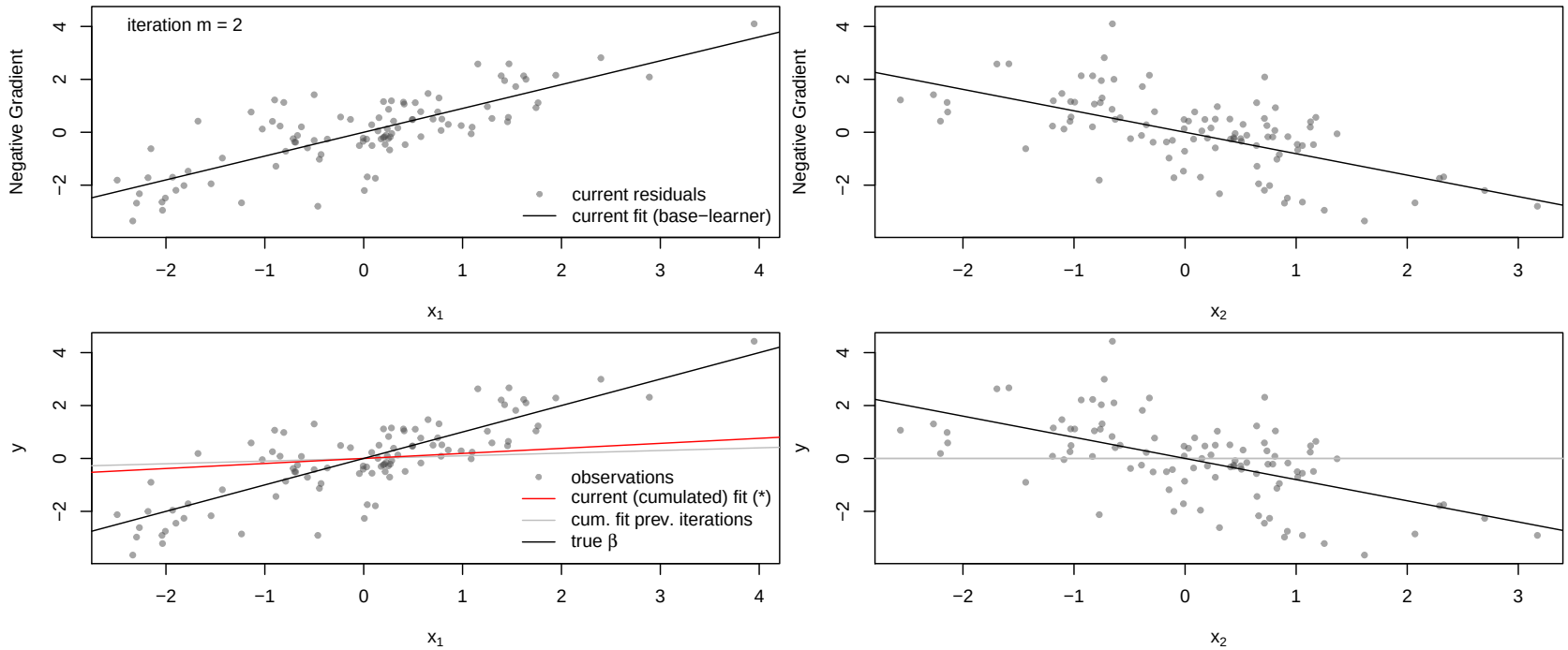


(*) current cumulated fit for β_j:

$$\hat{\beta}_j = \sum_m \underbrace{\nu}_{\text{step-length}} \cdot \underbrace{\hat{\beta}_j^{[m]}}_{\text{estimate in m-th step}} \cdot \underbrace{\mathbb{1}(j_m^* = j_m)}_{\text{selected in step m}}$$

In Short

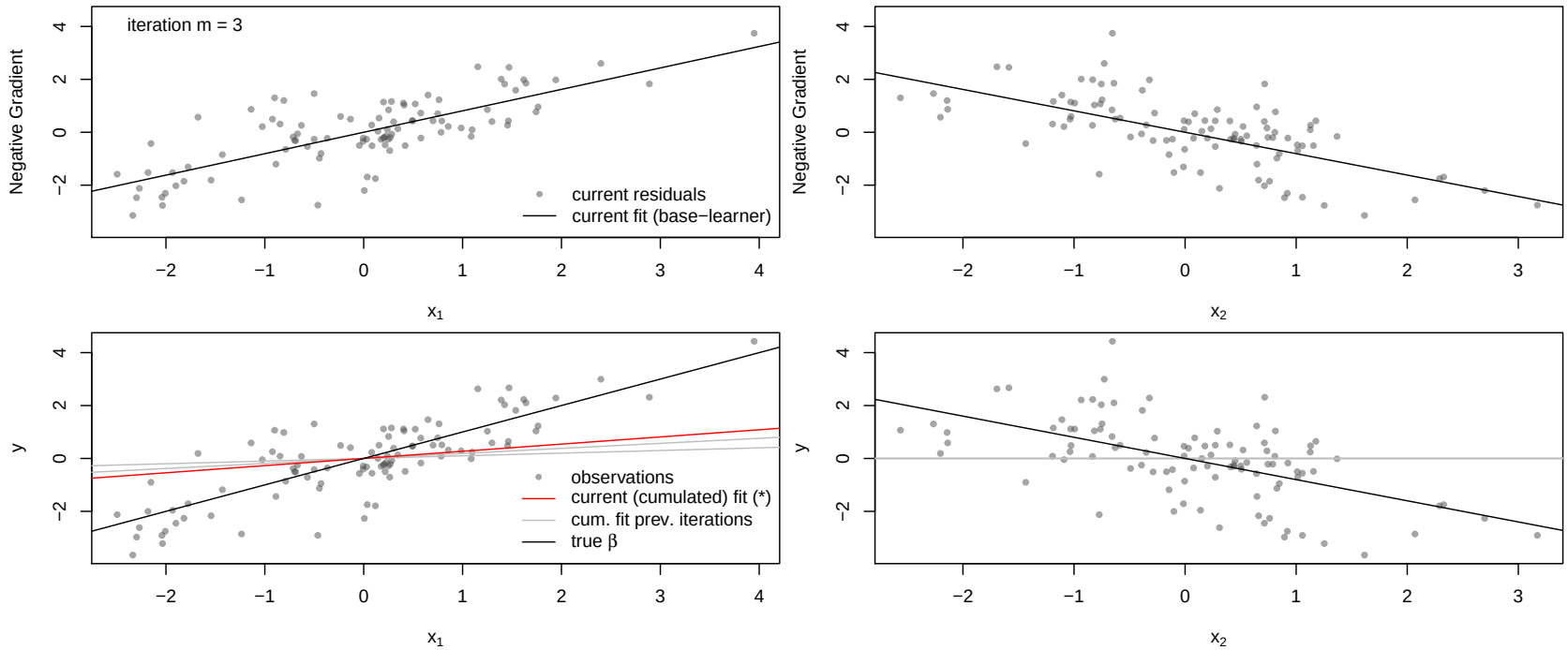
Boosting in Graphs



(*) current cumulated fit for β_j:

$$\hat{\beta}_j = \sum_m \underbrace{\nu}_{\text{step-length}} \cdot \underbrace{\hat{\beta}_j^{[m]}}_{\text{estimate in m-th step}} \cdot \underbrace{\mathbb{1}(j_m^* = j_m)}_{\text{selected in step m}}$$

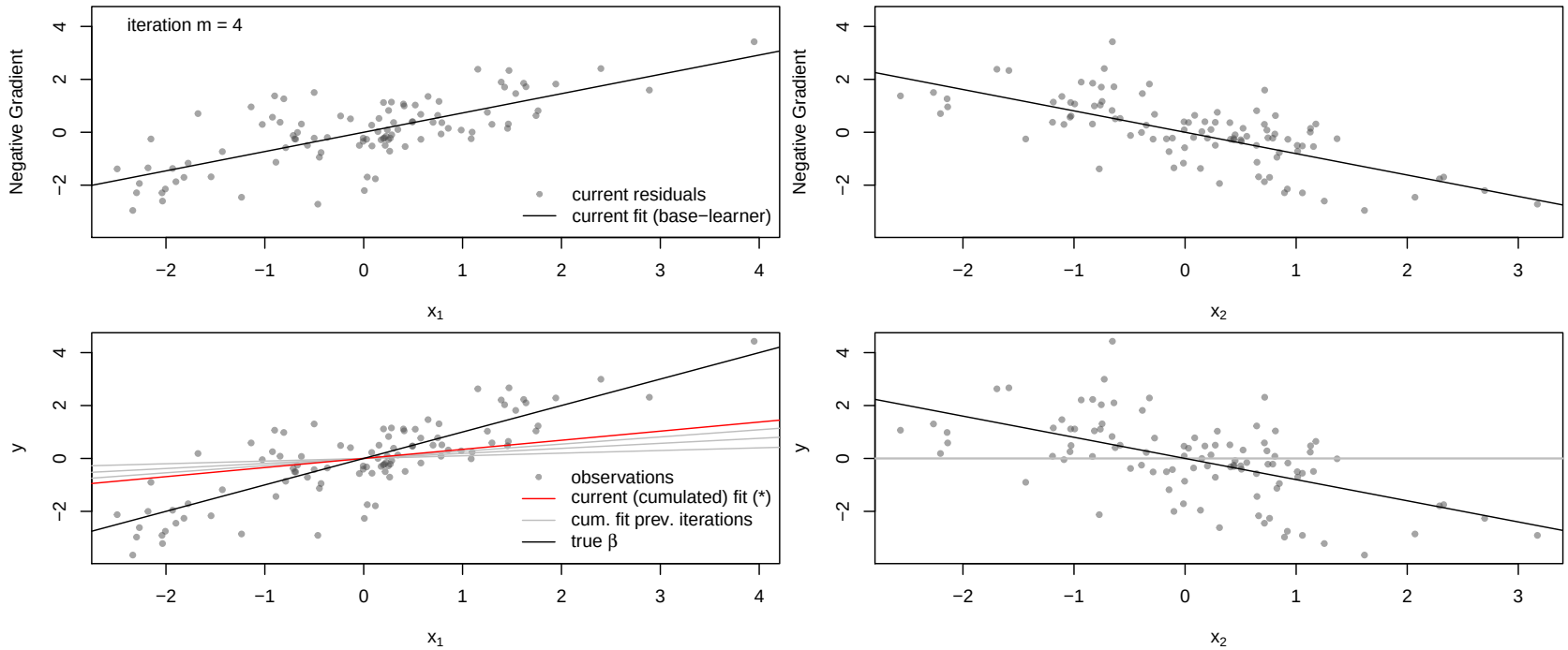
Boosting in Graphs



(*) current cumulated fit for β_j:

$$\hat{\beta}_j = \sum_m \underbrace{\nu}_{\text{step-length}} \cdot \underbrace{\hat{\beta}_j^{[m]}}_{\text{estimate in m-th step}} \cdot \underbrace{\mathbb{1}(j_m^* = j_m)}_{\text{selected in step m}}$$

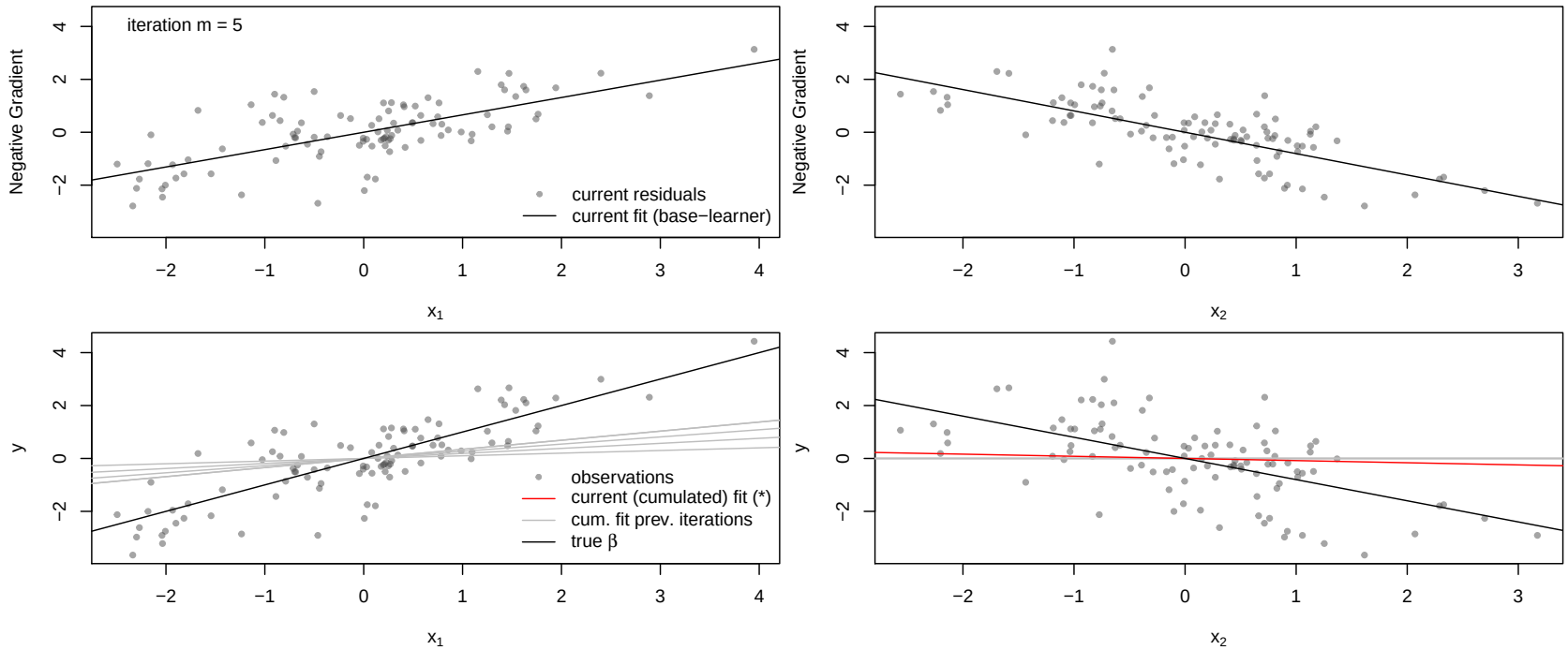
Boosting in Graphs



(*) current cumulated fit for β_j:

$$\hat{\beta}_j = \sum_m \underbrace{\nu}_{\text{step-length}} \cdot \underbrace{\hat{\beta}_j^{[m]}}_{\text{estimate in m-th step}} \cdot \underbrace{\mathbb{1}(j_m^* = j_m)}_{\text{selected in step m}}$$

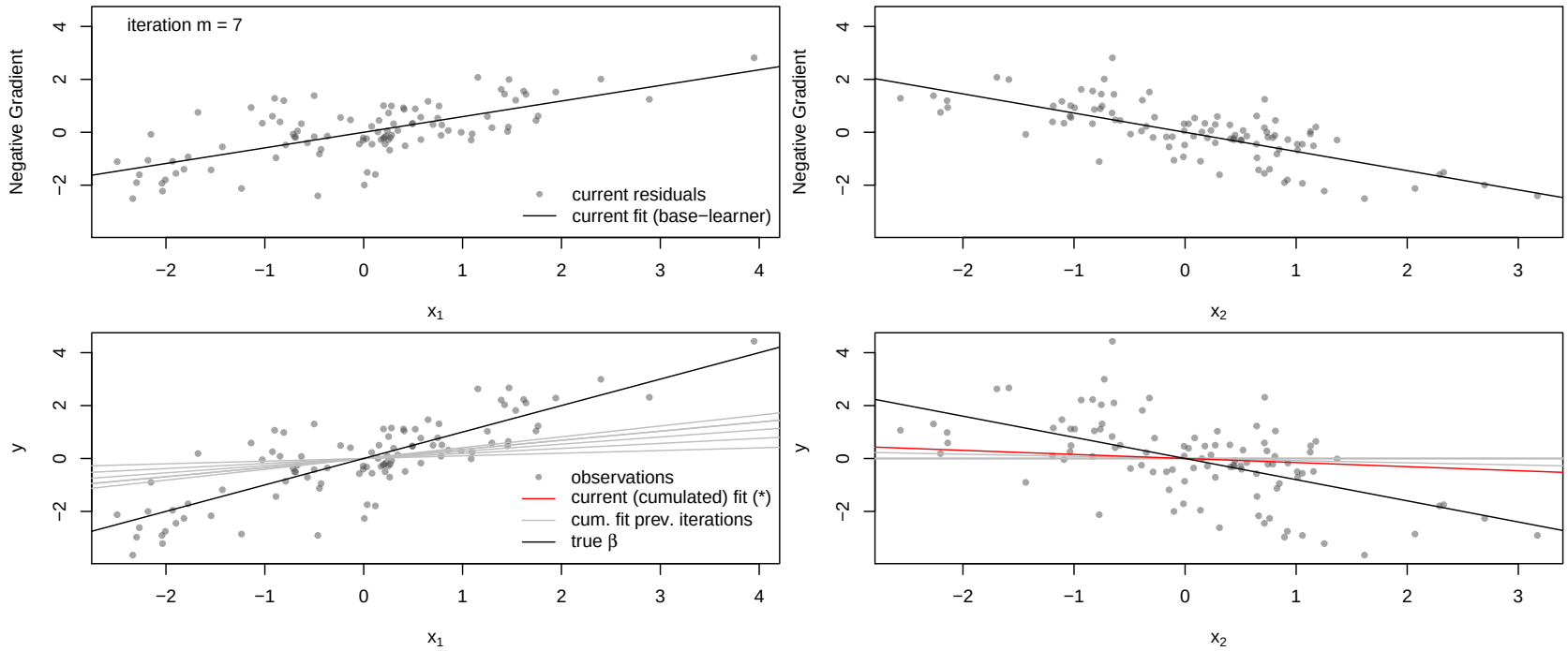
Boosting in Graphs



(*) current cumulated fit for β_j:

$$\hat{\beta}_j = \sum_m \underbrace{\nu}_{\text{step-length}} \cdot \underbrace{\hat{\beta}_j^{[m]}}_{\text{estimate in m-th step}} \cdot \underbrace{\mathbb{1}(j_m^* = j_m)}_{\text{selected in step m}}$$

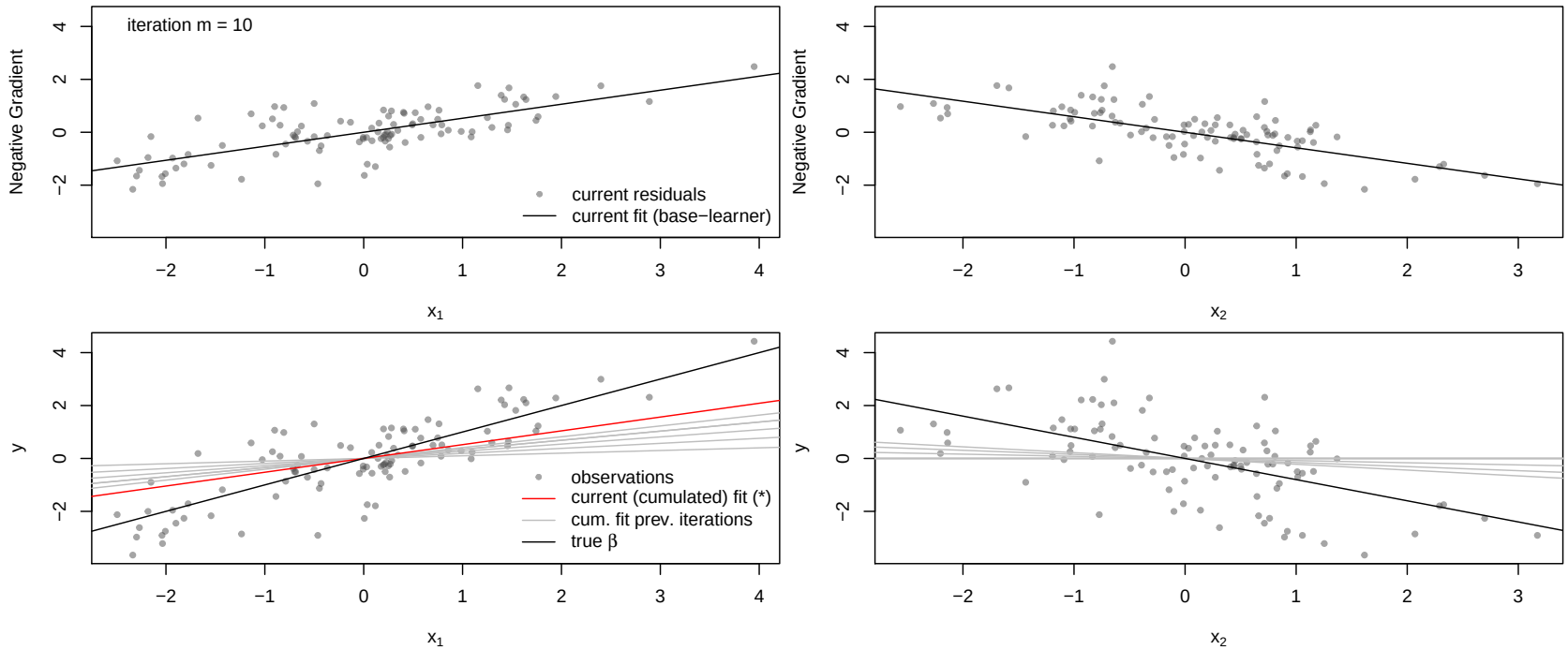
Boosting in Graphs



(*) current cumulated fit for β_j:

$$\hat{\beta}_j = \sum_m \underbrace{\nu}_{\text{step-length}} \cdot \underbrace{\hat{\beta}_j^{[m]}}_{\text{estimate in m-th step}} \cdot \underbrace{\mathbb{1}(j_m^* = j_m)}_{\text{selected in step m}}$$

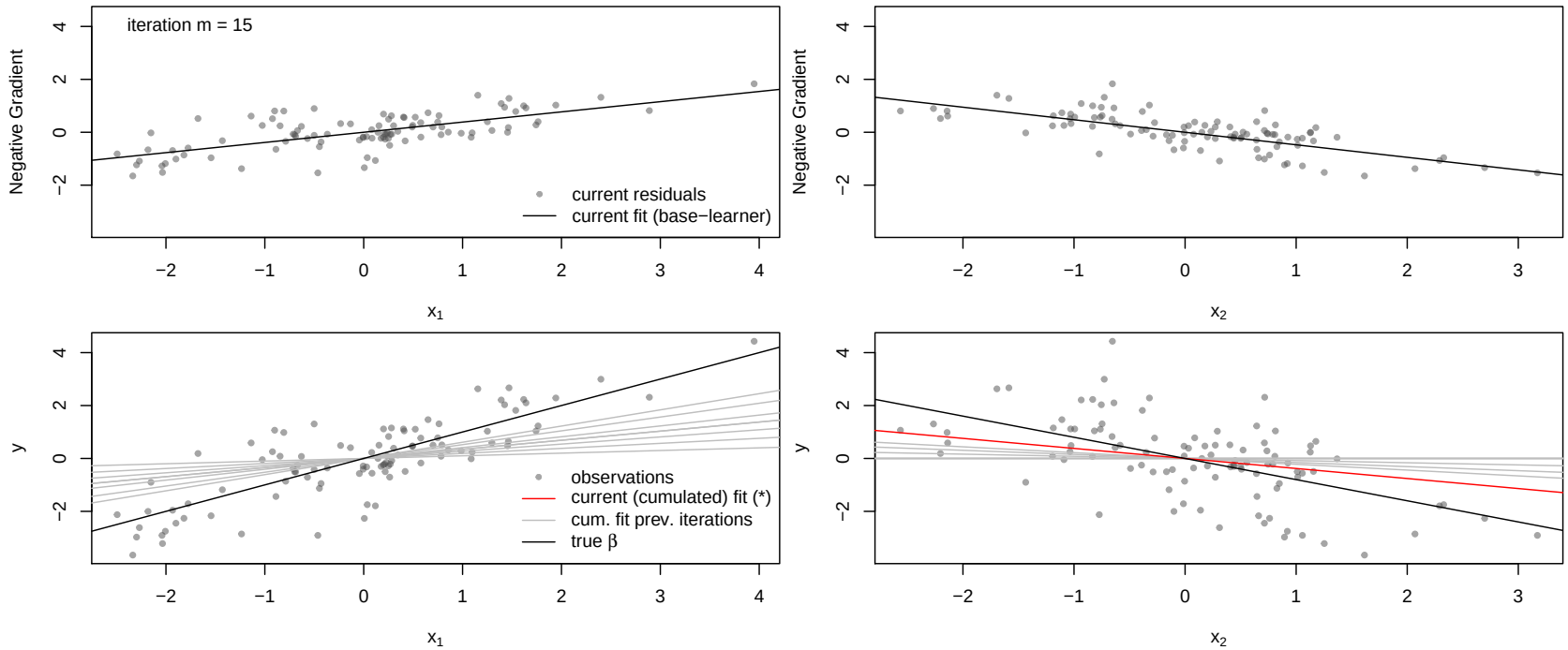
Boosting in Graphs



(*) current cumulated fit for β_j:

$$\hat{\beta}_j = \sum_m \underbrace{\nu}_{\text{step-length}} \cdot \underbrace{\hat{\beta}_j^{[m]}}_{\text{estimate in m-th step}} \cdot \underbrace{\mathbb{1}(j_m^* = j_m)}_{\text{selected in step m}}$$

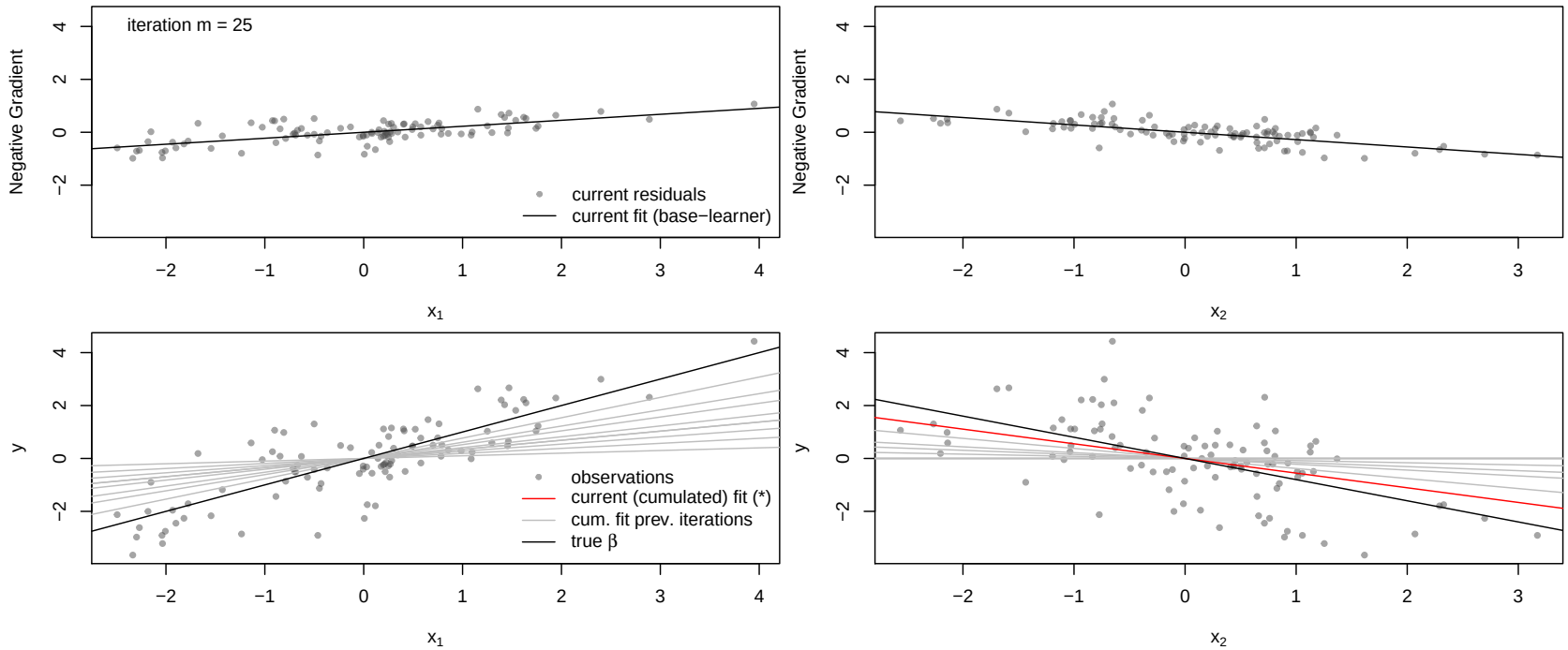
Boosting in Graphs



(*) current cumulated fit for β_j:

$$\hat{\beta}_j = \sum_m \underbrace{\nu}_{\text{step-length}} \cdot \underbrace{\hat{\beta}_j^{[m]}}_{\text{estimate in m-th step}} \cdot \underbrace{\mathbb{1}(j_m^* = j_m)}_{\text{selected in step m}}$$

Boosting in Graphs

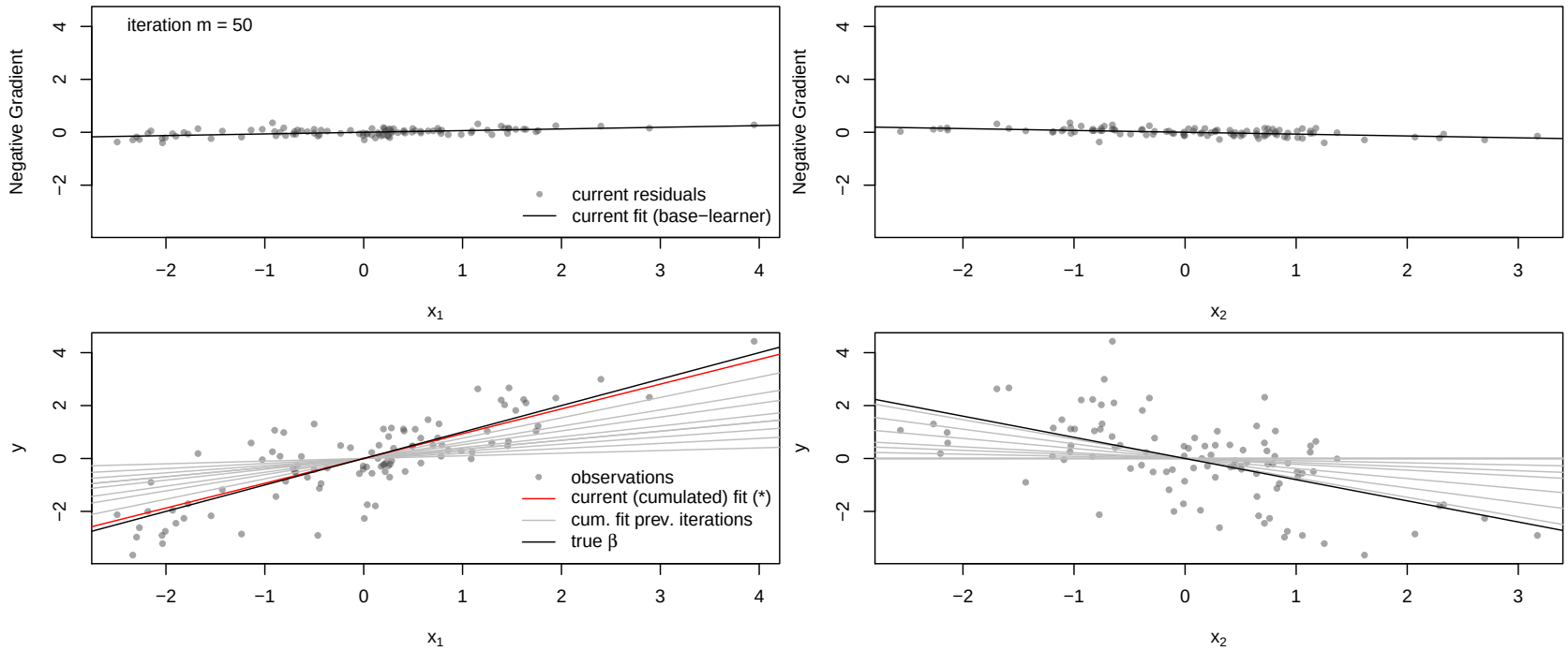


(*) current cumulated fit for β_j:

$$\hat{\beta}_j = \sum_m \underbrace{\nu}_{\text{step-length}} \cdot \underbrace{\hat{\beta}_j^{[m]}}_{\text{estimate in m-th step}} \cdot \underbrace{\mathbb{1}(j_m^* = j_m)}_{\text{selected in step m}}$$

In Short

Boosting in Graphs

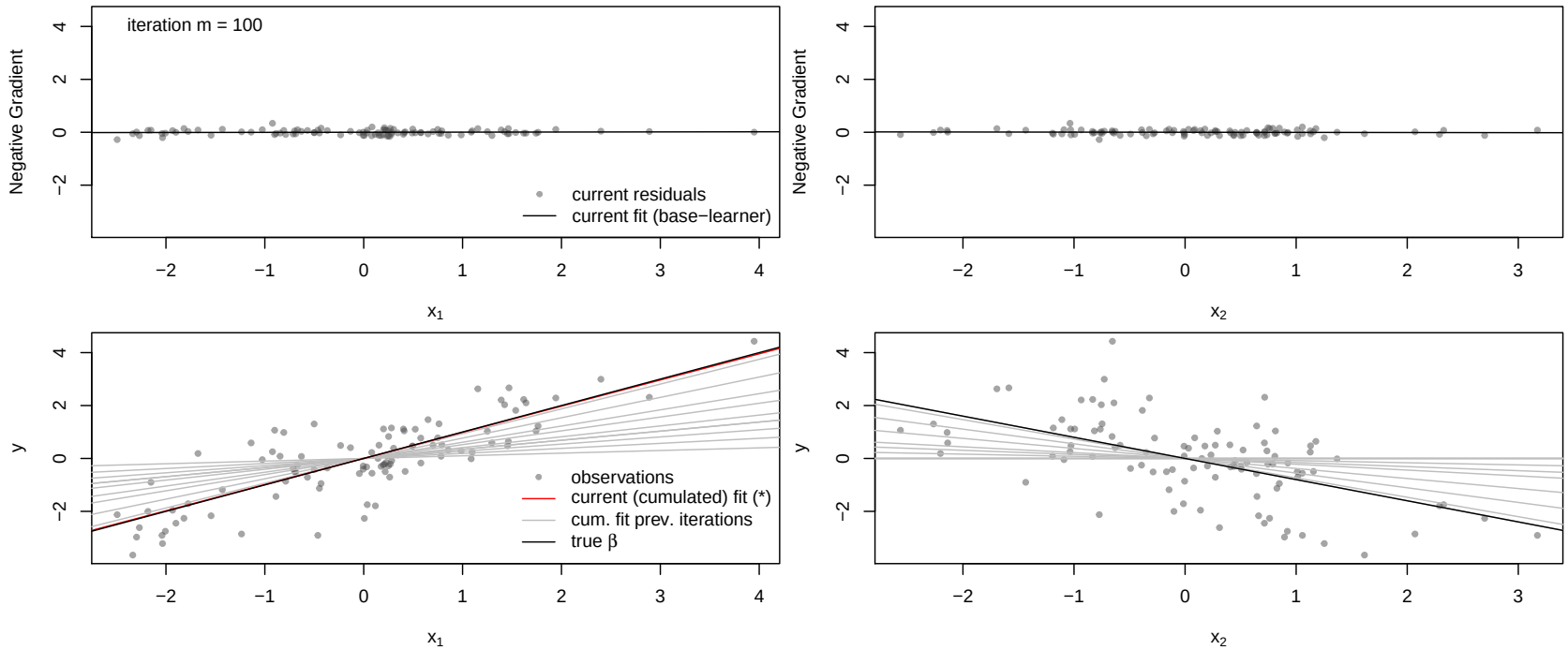


(*) current cumulated fit for β_j:

$$\hat{\beta}_j = \sum_m \underbrace{\nu}_{\text{step-length}} \cdot \underbrace{\hat{\beta}_j^{[m]}}_{\text{estimate in m-th step}} \cdot \underbrace{\mathbb{1}(j_m^* = j_m)}_{\text{selected in step m}}$$

In Short

Boosting in Graphs



(*) current cumulated fit for β_j:

$$\hat{\beta}_j = \sum_m \underbrace{\nu}_{\text{step-length}} \cdot \underbrace{\hat{\beta}_j^{[m]}}_{\text{estimate in m-th step}} \cdot \underbrace{\mathbb{1}(j_m^* = j_m)}_{\text{selected in step m}}$$

Remarks

- **Major tuning parameter:** m_{stop}
  (choose $\widehat{m}_{\text{stop,opt}}$ that minimizes empirical risk on "new data" via cross-validation, bootstrap, subsampling, ...)
- To avoid overly complex models use subsampling (see Mayr et al. 2012)
- Step-length ν is "no real tuning parameter" but governs how fast the algorithm converges (as long as it is small enough)
- If we use the $L2$ loss ("linear regression case"), the negative gradient reduces to least squares residuals as in a linear model
  - Boosting can be regarded as refitting residuals.

We stated that we use . . .

. . . component-wise boosting as a means of estimation and variable selection.

But how?

We stated that we use . . .

. . . component-wise boosting as a means of estimation and variable selection.

Variable Selection

. . . is achieved by

- selection of base-learner, i.e., component-wise boosting

  and

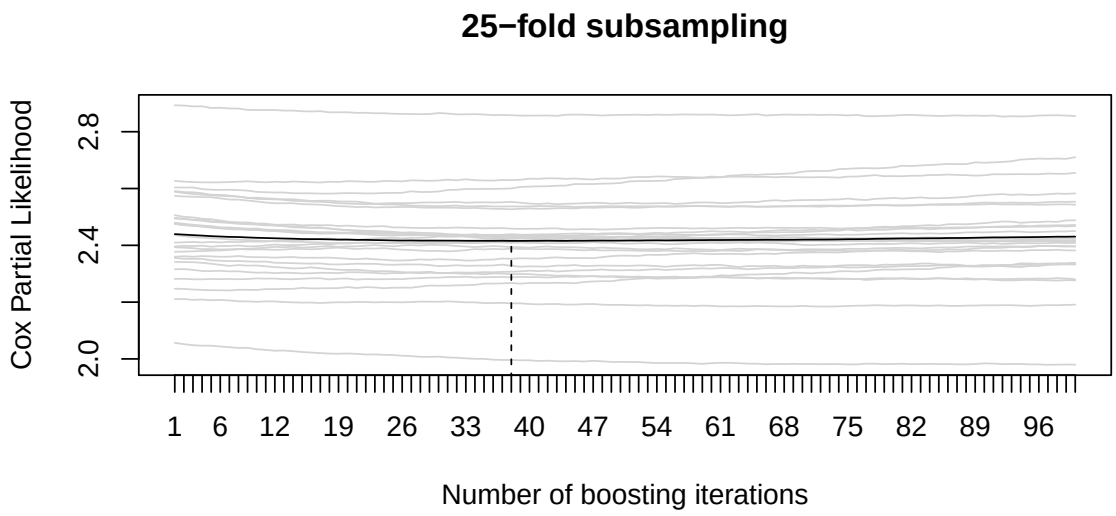
- early stopping,
  i.e., choose $\widehat{m}_{\text{stop,opt}}$ via cross-validation, bootstrap, resampling, out-of-bag sample, . . .

- Optimization of predictive performance.

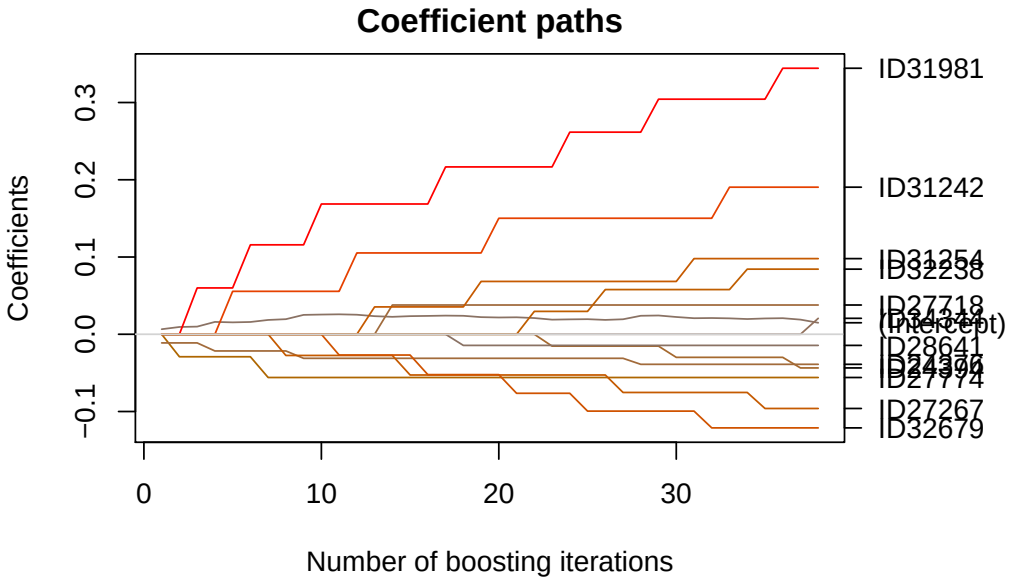
Rosenwald Data

```
> ## load packages
> library("mboost")
> library("survival")

> ## fit Cox model with boosting
> glm <- glmboost(Surv(time, status) ~ ., data = rosenwald,
>                 family = CoxPH())

> ## find optimal stopping iteration
> set.seed(1907)
> cv <- cv(model.weights(glm), type = "subsampling")
> cvr <- cvrisk(glm, folds = cv)
> mstop(cvr)
[1] 38

> ## subset model
> mstop(glm) <- mstop(cvr)
```

```
> plot(cvr)
```




```
> plot(glm)
```



```
> coef(glm, off2int = TRUE)

(Intercept)     ID27774     ID31242     ID31981     ID27718
 0.01489719 -0.05598384  0.19041074  0.34434745  0.03787356
    ID24394     ID24376     ID28641     ID27267     ID34344
-0.04357093 -0.03890142 -0.01442045 -0.09608845  0.02063545
    ID31254     ID32238     ID32679
 0.09791264  0.08429358 -0.12117418


> ## number of selected variables
> (psel <- length(coef(glm, off2int = TRUE)) - 1)

[1] 12

> ## fraction of selected variables
> round(psel / (ncol(rosenwald) - 2), 4)

[1] 0.0016
```

Take-Away Messages

- The boosting algorithm itself can be seen as a black box that can be used for model fitting.
- Boosting results in interpretable (linear) models.
- Boosting (intrinsically) allows for variable selection.
- Boosting models can be easily fitted using the package **mboost**.

Part 2: Boosting for GAMs and STARs

A Journey to Unbiased Variable Selection and Model Choice

Part 2: Boosting for GAMs and STARs

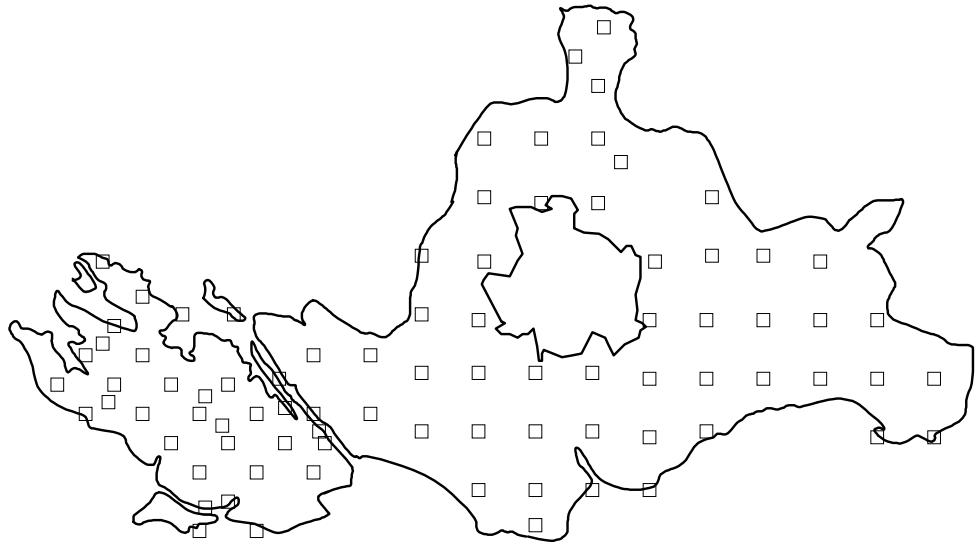
A Journey to Unbiased Variable Selection and Model Choice

joint work with
Torsten Hothorn, Universität Zürich,
Thomas Kneib, Georg-August-Universität Göttingen,
Matthias Schmid, IMBIE, Universität Bonn
based on Hofner et al. (2011)

Forest Health Data

- **Aim:** Identify predictors of the health status of beeches
- **Data:** Yearly visual forest health inventories carried out from 1983 to 2004 in a northern Bavarian forest district (Spessart)
- **Response:** binary defoliation indicator in year t ($y_{it} = 1$ defoliation above 25%)

- **Observation units:** 83 plots of beeches within a 15 km $\times$ 10 km area



- Large data set ($n = 1793$)
- Longitudinal data with spatial structure

Additional Covariates

Binary:
- type of stand
- application of fertilization

Categorical:
- thickness of humus layer in 5 ordered categories
- base saturation in 4 ordered categories

Continuous:
- average age of trees at the observation plot
- elevation above sea level in meters
- inclination of slope in percent
- depth of soil layer in centimeters
- pH-value at 0-2cm depth
- density of forest canopy in percent

Aims

Specific Aims

- Previous analyses resulted in models that contained linear and smooth effects as well as categorical covariates.
- Additionally, a spatial effect and a random effect for the plot could be identified.

Boosting

✓ Optimize prediction performance

✓ Resulting models are interpretable

✓ Variable selection

? Include non-linear effects

? Model choice, i.e., select the appropriate modeling alternative (linear vs. flexible vs. . . .)

Considered Model Class

Structured Additive Regression (STAR) Model

$$\mu_i = \mathbb{E}(y|\boldsymbol{x}_i) = h(\eta_i(\boldsymbol{x}_i))$$

with response function h and additive predictor

$$\eta_i(\boldsymbol{x}_i) = \beta_0 + \sum_{j=1}^{J} f_j(x_{ij}),$$

Generic representation of covariate effects $f_j(x_{ij})$ comprises

a) **linear effects:** $f_j(x_{ij}) = x_{ij}\beta_j$

b) **smooth effects:** $f_j(x_{ij}) = f_{j,\text{smooth}}(x_{ij})$

c) **categorical effects:** $f_j(x_{ij}) = \tilde{\boldsymbol{z}}_{ij}^\top \boldsymbol{\gamma}_j$
($\tilde{\boldsymbol{z}}_{ij}$ dummy-coded categorical covariate corresponding to x_{ij})

d) further effects as
spatial effects, **random effects**, **varying coefficients**, ...

- model is just a sum of linear, smooth, categorical and ... effects.

Component-wise FGD Boosting

For model fitting and variable selection and model choice, we use **component-wise** boosting.

- specify a separate base-learner for each covariate
  (= variable selection)
- base-learners represent functions $f_j(\cdot)$ from structured additive predictor

- possible extension: specify a separate base-learner for each modeling alternative (e.g., linear effect vs. smooth effect)
  (= model choice)

Component-wise FGD Boosting

For model fitting and variable selection and model choice, we use **component-wise** boosting.

- specify a separate base-learner for each covariate
  (= variable selection)
- base-learners represent functions $f_j(\cdot)$ from structured additive predictor

- possible extension: specify a separate base-learner for each modeling alternative (e.g., linear effect vs. smooth effect)
  (= model choice)

- boosting proceeds in a stagewise fashion and
- updates only the best-fitting base-learner in each step
- ▶ variable and model selection is achieved

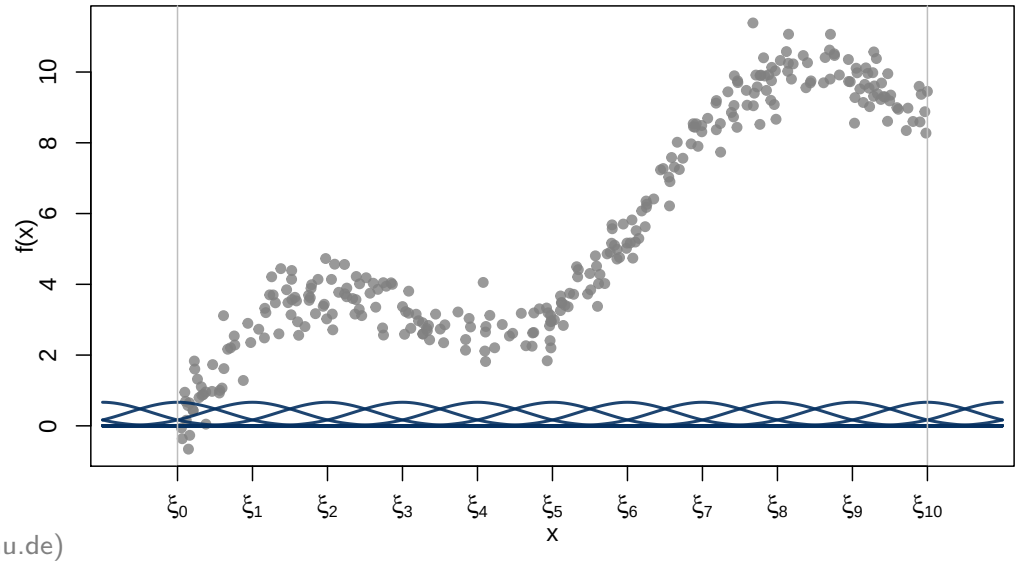
Estimating Smooth Effects

A Short Review of P-splines (Eilers and Marx 1996)

A **smooth function** can be expressed with B-splines as

$$f(x) = \sum_{j=1}^{J} \beta_j B_j(x; l) = \boldsymbol{\beta}^\top \boldsymbol{B}(x), \tag{1}$$

where $B_j(\cdot; l)$ is the j-th B-spline basis function of **degree** l defined on a **grid of knots** ξ_k.



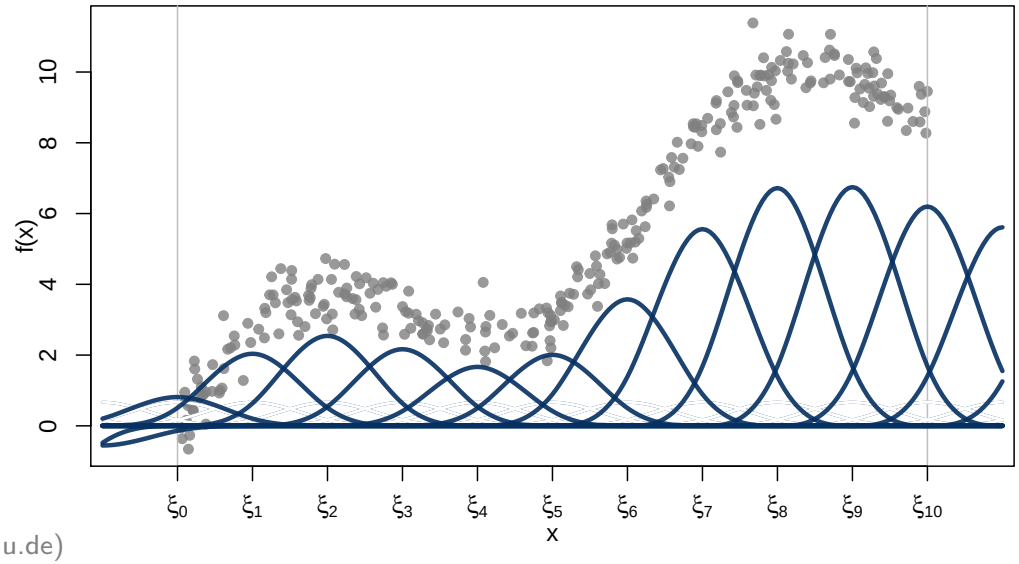
Estimating Smooth Effects

A Short Review of P-splines (Eilers and Marx 1996)

A **smooth function** can be expressed with B-splines as

$$f(x) = \sum_{j=1}^{J} \beta_j B_j(x; l) = \boldsymbol{\beta}^\top \boldsymbol{B}(x), \tag{1}$$

where $B_j(\cdot; l)$ is the j-th B-spline basis function of **degree** l defined on a **grid of knots** ξ_k.



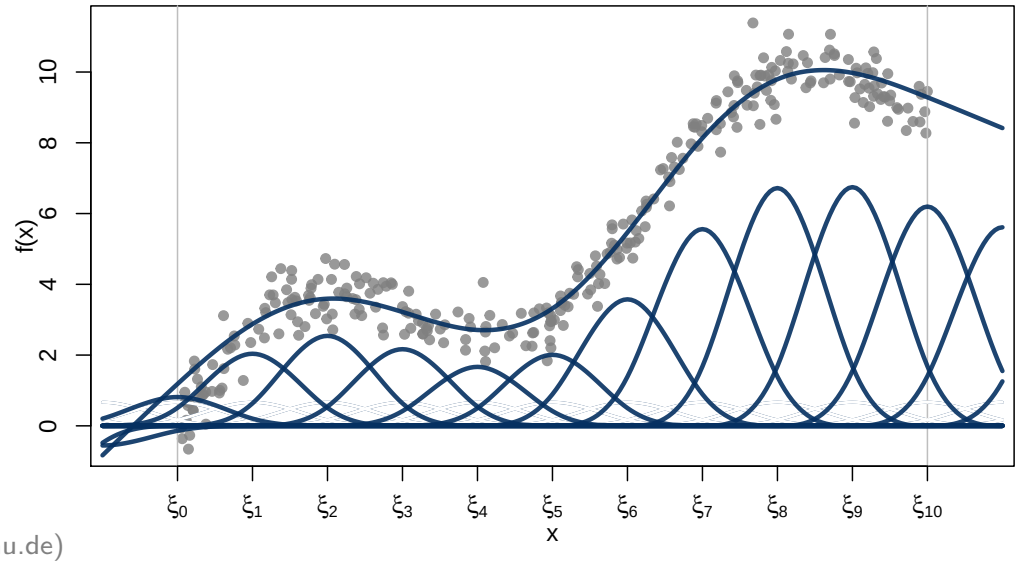
Estimating Smooth Effects

A Short Review of P-splines (Eilers and Marx 1996)

A **smooth function** can be expressed with B-splines as

$$f(x) = \sum_{j=1}^{J} \beta_j B_j(x; l) = \boldsymbol{\beta}^\top \boldsymbol{B}(x), \tag{1}$$

where $B_j(\cdot; l)$ is the j-th B-spline basis function of **degree** l defined on a **grid of knots** ξ_k.



- Smoothness enforced by additional **penalty on adjacent B-splines**:

$$\mathcal{J}(\boldsymbol{\beta}; d) = \sum_{j=d+1}^{J} (\Delta^d \beta_j)^2$$

  where d is the order of the difference penalty.
- **Difference operator** Δ^d is defined as:

$$\Delta \beta_j = \Delta^1 \beta_j = (\beta_j - \beta_{j-1})$$
$$\Delta^2 \beta_j = \Delta(\Delta \beta_j) = \beta_j - 2\beta_{j-1} + \beta_{j-2}$$

- Smoothness enforced by additional **penalty on adjacent B-splines**:

$$\mathcal{J}(\boldsymbol{\beta}; d) = \sum_{j=d+1}^{J} (\Delta^d \beta_j)^2$$

  where d is the order of the difference penalty.
- **Difference operator** Δ^d is defined as:

$$\Delta \beta_j = \Delta^1 \beta_j = (\beta_j - \beta_{j-1})$$
$$\Delta^2 \beta_j = \Delta(\Delta \beta_j) = \beta_j - 2\beta_{j-1} + \beta_{j-2}$$

Estimation

Use a penalized least squares base-learner that optimizes

$$(\boldsymbol{u} - \boldsymbol{B}\boldsymbol{\beta})^\top (\boldsymbol{u} - \boldsymbol{B}\boldsymbol{\beta}) + \lambda \mathcal{J}(\boldsymbol{\beta}; d)$$

- Smoothness enforced by additional **penalty on adjacent B-splines**:

$$\mathcal{J}(\boldsymbol{\beta}; d) = \sum_{j=d+1}^{J} (\Delta^d \beta_j)^2$$

where d is the order of the difference penalty.

- **Difference operator** Δ^d is defined as:

$$\Delta\beta_j = \Delta^1\beta_j = (\beta_j - \beta_{j-1})$$

$$\Delta^2\beta_j = \Delta(\Delta\beta_j) = \beta_j - 2\beta_{j-1} + \beta_{j-2}$$

Estimation

Use a penalized least squares base-learner that optimizes

$$(\boldsymbol{u} - \boldsymbol{B}\boldsymbol{\beta})^\top (\boldsymbol{u} - \boldsymbol{B}\boldsymbol{\beta}) + \lambda \mathcal{J}(\boldsymbol{\beta}; d)$$

Estimating Smooth Effects with Boosting

- Fix λ (corresponding to e.g. 4 degrees of freedom)
- Model adapts to complexity by multiple updates of base-learner

▸ Jump to Animation

End of insertion.

Variable and Model Selection

- Specify separate base-learners per variable and effect type.
- Use cross-validation methods (e.g. k-fold cross-validation, bootstrap, subsampling, ...) to find optimal m_{stop}.
- Boosting "decides" which base-learners are "important".

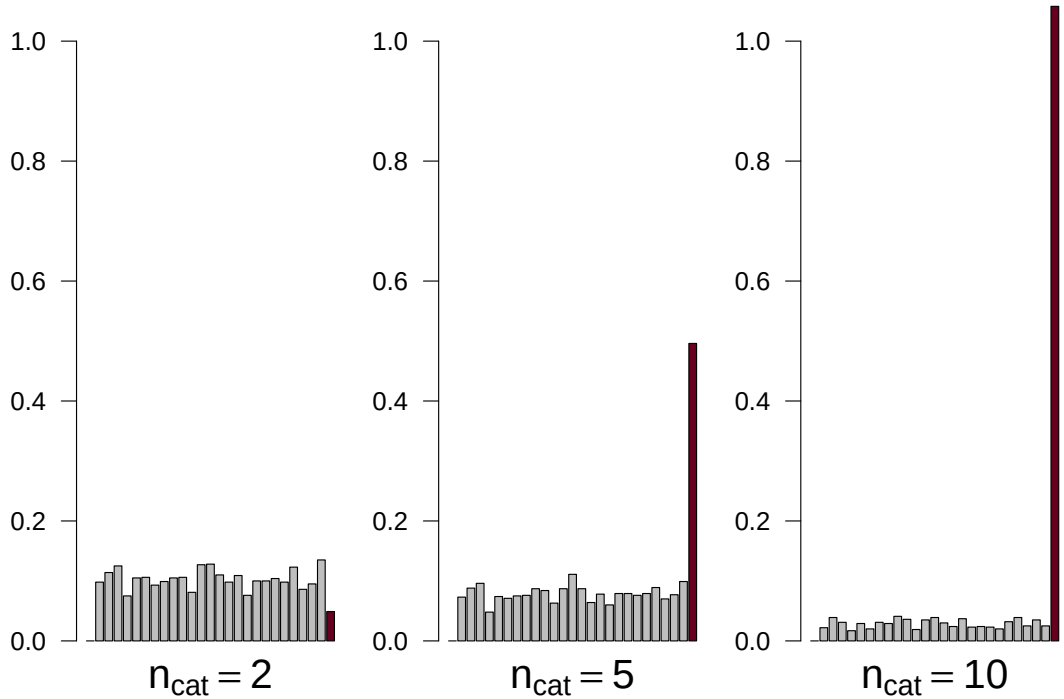
Variable and Model Selection

- Specify separate base-learners per variable and effect type.
- Use cross-validation methods (e.g. k-fold cross-validation, bootstrap, subsampling, ...) to find optimal m_{stop}.
- Boosting "decides" which base-learners are "important".

But ...

Biased Selection: An Illustrative Example

25 non-informative continuous variables, 1 non-informative categorical variable



- Increasing selection rate — only due to an increasing number of categories

Problems and a Solution

- **Variable selection** and **model choice** can be **seriously biased** if some base-learners offer higher flexibility.
  - Variable Selection Bias:
    e.g., continuous covariate $\prec$ categorical covariate (with many categories)
  - Model Selection Bias:
    e.g., linear effect $\prec$ smooth effect
- Unbiased (or at least improved) selection *desired*!

Problems and a Solution

- **Variable selection** and **model choice** can be **seriously biased** if some base-learners offer higher flexibility.
  - Variable Selection Bias:
    e.g., continuous covariate $\prec$ categorical covariate (with many categories)
  - Model Selection Bias:
    e.g., linear effect $\prec$ smooth effect
- Unbiased (or at least improved) selection *desired*!

- **Possible solution:**
  Make the competitors comparable with respect to their flexibility (measured by the degrees of freedom)

Penalized Least Squares Base-Learners

Consider (penalized) least squares base-learners

$$\hat{g}_j(\boldsymbol{x}) = \underbrace{\boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{K})^{-1} \boldsymbol{X}^\top}_{=: \boldsymbol{S} \text{ (smoother matrix)}} \boldsymbol{u}^{[m]},$$

where $\boldsymbol{X}$ is a suitable design matrix.

Penalized Least Squares Base-Learners

Consider (penalized) least squares base-learners

$$\hat{g}_j(\boldsymbol{x}) = \underbrace{\boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{K})^{-1} \boldsymbol{X}^\top}_{=: \boldsymbol{S} \text{ (smoother matrix)}} \boldsymbol{u}^{[m]},$$

where $\boldsymbol{X}$ is a suitable design matrix.

Examples of (penalized) LS base-learners

- Unpenalized base-learners ($\lambda = 0$)
- Ridge-penalized base-learners for unordered categorical covariates ($\boldsymbol{X}$ e.g., dummy coded)
- Base-learners with first order difference penalty for ordered categorical covariates (Gertheiss and Tutz 2009) ($\boldsymbol{X}$ e.g., dummy coded)
- P-spline base-learners with second order difference penalty for continuous covariates ($\boldsymbol{X}$ B-spline basis expansion)

Penalized Least Squares Base-Learners

Central Idea

Set $\text{df} = 1$ for all base-learners to prevent selection bias

NB: Final model can adopt (much) higher flexibility due to the iterative nature of boosting!

Theoretical Considerations (Hofner et al. 2011)

Instead of

$$\text{df} := \text{trace}(\boldsymbol{S})$$

define

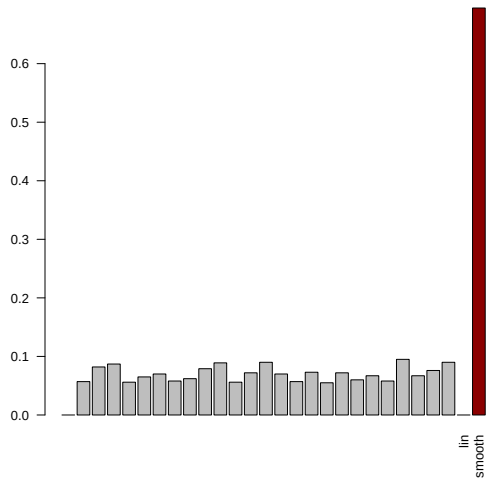
$$\text{df} := \text{trace}(2\boldsymbol{S} - \boldsymbol{S}^\top \boldsymbol{S})$$

(tailored for the comparison of RSS (see also Buja et al. 1989))

Biased Selection of Smooth Effects

25 non-informative continuous variables with linear base-learners,
1 non-informative continuous variable with linear and smooth base-learner

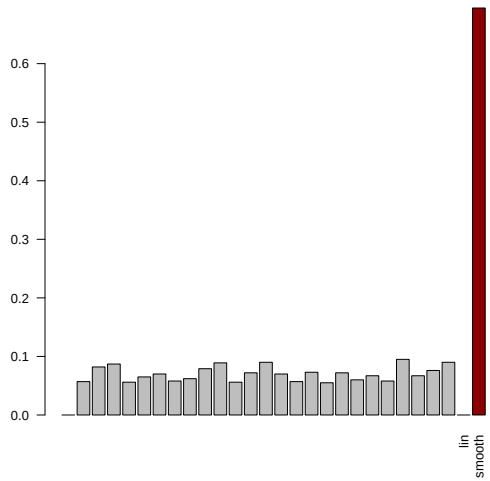
Degrees of freedom for linear effects ($\mathrm{df} = 1$) and smooth effects ($\mathrm{df} \gg 1$) are not comparable. If we use more flexible smooth base-learners (e.g., $\mathrm{df} = 4$) the selection is biased



Biased Selection of Smooth Effects

25 non-informative continuous variables with linear base-learners,
1 non-informative continuous variable with linear and smooth base-learner

Degrees of freedom for linear effects ($\mathrm{df} = 1$) and smooth effects ($\mathrm{df} \gg 1$) are not comparable. If we use more flexible smooth base-learners (e.g., df $= 4$) the selection is biased



- **Problem:** We cannot make df of smooth effects arbitrary small, i.e., $\mathrm{df} > 1\,(\lambda \to \infty)$ (for penalties of order $d \geq 2$)
- Hence: Polynomial of order $d - 1$ remains unpenalized

A Solution – P-Spline Decomposition

- For model choice we apply the decomposition

$$f_{\text{smooth}}(x) = \underbrace{\beta_0 + \beta_1 x}_{\text{unpenalized, parametric part}} + \underbrace{f_{\text{smooth,centered}}(x)}_{\text{deviation from polynomial}}$$

(based on Kneib et al. 2009)

- Add unpenalized part as separate, parametric base-learners
- Assign $\text{df} = 1$ to the centered effect (and add as P-spline base-learner)

A Solution – P-Spline Decomposition

- For model choice we apply the decomposition

$$f_{\text{smooth}}(x) = \underbrace{\beta_0 + \beta_1 x}_{\text{unpenalized, parametric part}} + \underbrace{f_{\text{smooth,centered}}(x)}_{\text{deviation from polynomial}}$$

(based on Kneib et al. 2009)

- Add unpenalized part as separate, parametric base-learners
- Assign $\text{df} = 1$ to the centered effect (and add as P-spline base-learner)

Thus:

- Modeling components are comparable (w.r.t. df)
- Model choice between: linear effects and smooth effects. Further modeling alternatives, as varying coefficient terms, spatial effects, ... can be added analogously

Technical realization (see Fahrmeir et al. 2004):

decomposing the vector of regression coefficients $\boldsymbol{\beta}$ into $(\widetilde{\boldsymbol{\beta}}_{unpen}, \widetilde{\boldsymbol{\beta}}_{pen})$ utilizing a spectral decomposition of the penalty matrix

Forest Health Prediction

Model Formula (abbrev.)

```
fm <- defoliation ~ bols(intercept, intercept = FALSE) +
   bols(fertilization, intercept = FALSE, df = 1) +
   bols(saturation, intercept = FALSE, df = 1) +
   ...
   bols(canopy, intercept = FALSE) +
   bbs(canopy, center = TRUE, df = 1, knots = 20) +
   bols(elevation, intercept = FALSE) +
   bbs(elevation, center = TRUE, df = 1, knots = 20) +
   bols(year, intercept = FALSE) +
   bbs(year, center = TRUE, df = 1, knots = 20) +
   bols(age, intercept = FALSE) +
   bbs(age, center = TRUE, df = 1, knots = 20) +
   ...
   bspatial(x, y, center = TRUE, df = 1, differences = 1, knots = 12) +
   brandom(id, df = 1)
```

Note: Covariates that are fitted in a base-learner without intercept (`intercept = FALSE`) need to be centered (by hand) ahead of fitting the model.

▸ Importance of centering

Forest Health Prediction

Model Fitting

```
## fit model
forest <- gamboost(fm, data = beeches,
                   family = Binomial())

## stratified bootstrap
stratified_cv <- function() {
    ...
}
cv_folds <- stratified_cv(beeches$id, replace = TRUE)
mstop_10 <- cvrisk(forest, folds = cv_folds, grid = 1:5000)

## subset model to optimal number of boosting iterations
## (i.e., continue until mstop is reached in this case)
mstop(forest) <- mstop(mstop_10)
```

Results

Using component-wise (penalized) least squares base-learners with $1\,\mathrm{df}$ each, we get a **final model** with

Parametric effects for fertilization (binary), base saturation (ordinal), age and calender time

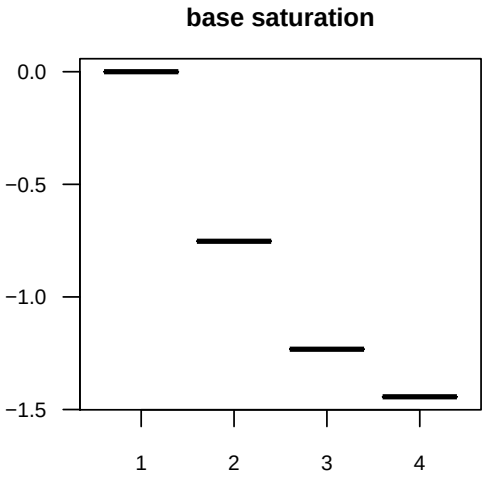
Nonparametric effect for canopy density

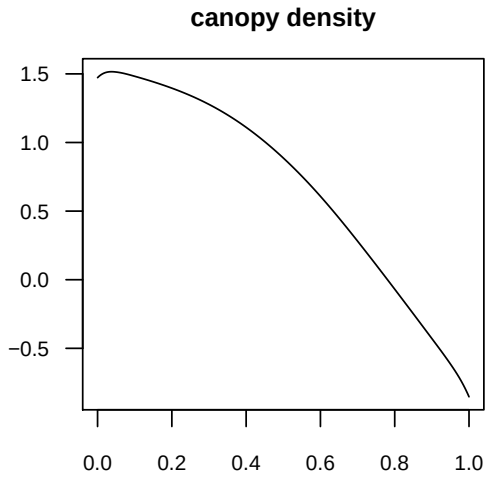
Spatial effect + unstructured random effect (with a clear domination of the latter)

Not selected: thickness of humus layer, ph-value, soil depth, type of stand, inclination of slope, elevation above sea level

Forest Health Data

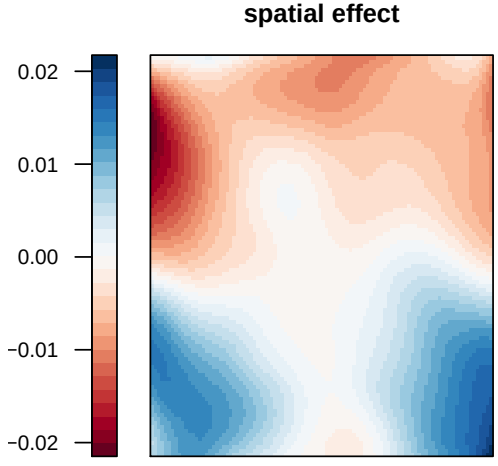
Results (ctd.)

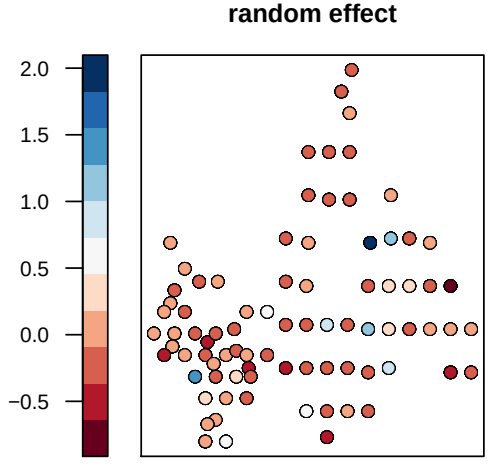




Further linear effects

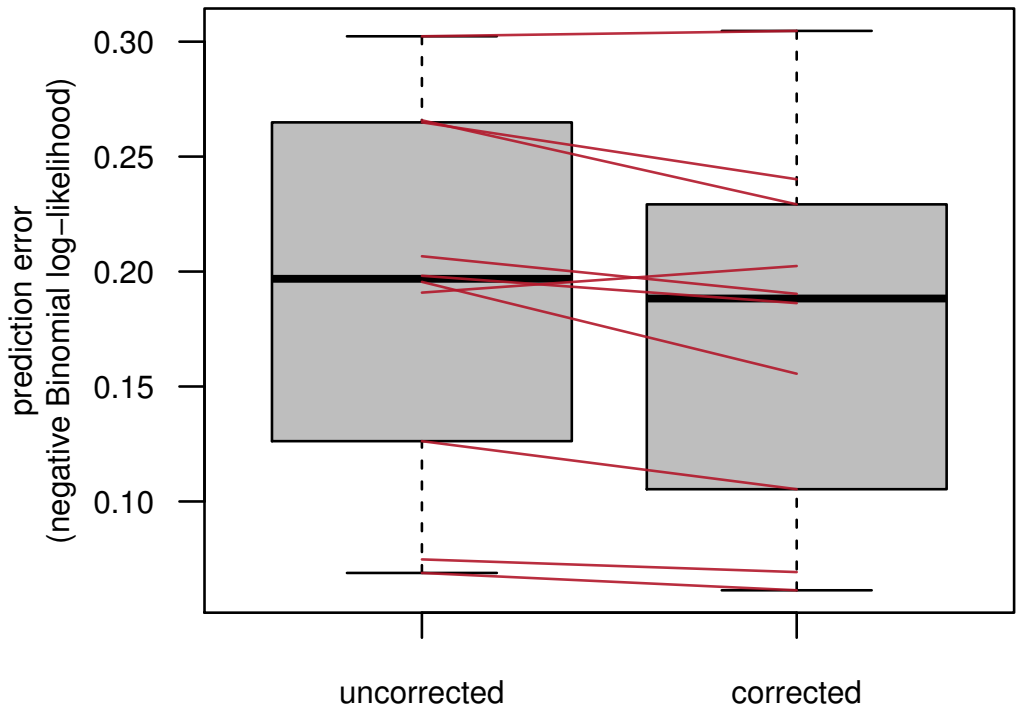
Covariates	β
Fertilization	-0.766
Age	0.016
Year	0.070





Prediction Accuracy

- **Risk on 10 test samples.** For each sample, each model was tuned separately by 25-fold cross-validation on the learning sample.



- equal degrees of freedom lead to bias corrected selection of base-learners **and** improved prediction accuracy.

Take-Away Messages

- Boosting results in interpretable models, if one uses linear or smooth base-learners (i.e., no tree-based base-learners).
- Boosting (intrinsically) allows for variable and model selection.
- We get a severe reduction of selection bias by using penalized base-learners with equal df.
- Use a suitable definition of degrees of freedom $\mathrm{df} = \mathrm{trace}(2\boldsymbol{S} - \boldsymbol{S}^\top \boldsymbol{S})$.

Part 3: Biomarker Discovery

Controlling False Discoveries in High-dimensional Situations

in cooperation with
Markus Göker, DSMZ, Braunschweig
Luigi Boccuto, GCC, Greenwood, USA

Identification of Biomarkers for ASD patients (yes/no)

Autism Spectrum Disorders (ASD):

- relatively common neurodevelopmental disease
- biological basis incompletely determined
- no laboratory test for these conditions
- (relatively) hard to diagnose

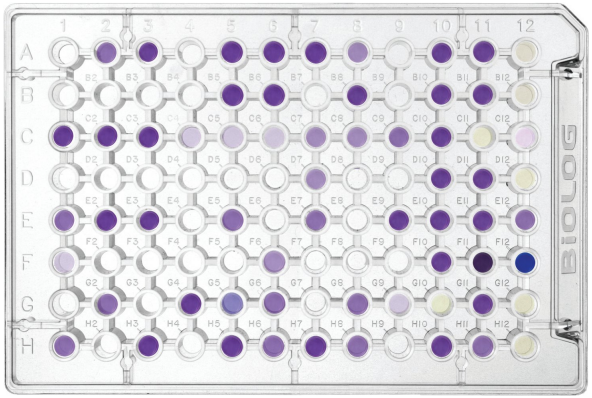
Aim:

Detect differentially expressed amino acid pathways,

i.e., amino acid pathways that differ between healthy subjects and ASD patients.

Identification of Biomarkers for ASD patients (yes/no)

- **Available Data:**
  - Cell lines of $n = 35$ subjects (17 ASD patients and 18 controls)
- **Measurements:**
  - Phenotype Microarrays (PM)
    - 96-well array per patient
    - Each well has a different carbon energy source
    - Maximum reaction ($=$ cellular activity) per well is measured (by a color reaction)



(Source: Biolog Inc., http://www.biolog.com)

- Measurements describe metabolism of subjects (on cell basis)

Modeling Strategy

- Fit a log-linear model for the PM measurements
- Differential expressions modeled using interactions
- Use boosting methods for variable selection

Modeling Strategy

- Fit a log-linear model for the PM measurements
- Differential expressions modeled using interactions
- Use boosting methods for variable selection

Yet,

- in high-dimensional settings, i.e., with many predictors, we might select a lot of uninformative variables.
- In many situations a **formal selection procedure with error control** seems advisable.

Modeling Strategy

- Fit a log-linear model for the PM measurements
- Differential expressions modeled using interactions
- Use boosting methods for variable selection

Yet,

- in high-dimensional settings, i.e., with many predictors, we might select a lot of uninformative variables.
- In many situations a **formal selection procedure with error control** seems advisable.

- Use stability selection.

Stability Selection (Meinshausen and Bühlmann 2010)

... is a versatile approach, which can be combined with (all) high-dimensional variable selection approaches.

... is based on subsampling (▶ draw samples without replacement).

... controls the per-family error rate PFER $= \mathbb{E}(V)$, where V is the number of false positives.

Insertion

Overview of Error Rates (see e.g. Dudoit et al. 2003)

per-family error rate (PFER): $\mathbb{E}(V)$

per-comparison error rate (PCER): $\mathbb{E}(V)/m$

standard testing procedure, no multiplicity correction

family-wise error rate (FWER): $\mathbb{P}(V \geq 1)$

false discovery rate (FDR): $\mathbb{E}(\frac{V}{R})$

	Keep H_0	Reject H_0	
H_0 true	U	V	m_0
H_1 true	T	S	$m - m_0$
	$m - R$	R	m

Insertion

Overview of Error Rates (see e.g. Dudoit et al. 2003)

per-family error rate (PFER): $\mathbb{E}(V)$

per-comparison error rate (PCER): $\mathbb{E}(V)/m$

standard testing procedure, no multiplicity correction

family-wise error rate (FWER): $\mathbb{P}(V \geq 1)$

false discovery rate (FDR): $\mathbb{E}(\frac{V}{R})$

Note: The PFER is very conservative

- For fixed α, PFER is more conservative than FWER
  FWER is more conservative than PCER
- For fixed α, FWER is more conservative than FDR
  and thus PFER is more conservative than FDR

Stability Selection

Algorithm (simplified)

1. Select a random subset of size $\lfloor n/2 \rfloor$ of the data.
2. Fit boosting model until q variables are selected (out of p).
3. Record which variables were selected.
4. Repeat $B = 100$ times.

5. Compute selection frequency per variable.
6. Select variables with frequency $\geq \pi_{\text{thr}}$.

Stability Selection

Algorithm (simplified)

1. Select a random subset of size $\lfloor n/2 \rfloor$ of the data.
2. Fit boosting model until q variables are selected (out of p).
3. Record which variables were selected.
4. Repeat $B = 100$ times.

5. Compute selection frequency per variable.
6. Select variables with frequency $\geq \pi_{\text{thr}}$.

- Conservative upper bound for the per-family error rate (PFER):

$$\text{PFER} = \mathbb{E}(V) \leq \frac{q^2}{(2\pi_{\text{thr}} - 1)p}$$

(if exchangeability assumption holds for all noise variables)

Improved Stability Selection (Shah and Samworth 2013)

- Tighter, i.e., less conservative error bounds can be derived under certain conditions.

  a) **If distribution of (simultaneous) selection probabilities is unimodal:**

$$\mathbb{E}(V) \leq \frac{q^2}{c(\pi_{\text{thr}}, B) \cdot p} \leq \frac{q^2}{(2\pi_{\text{thr}} - 1)p}$$

  b) **If distribution of (simultaneous) selection probabilities is r-concave:**

$$\begin{aligned} \mathbb{E}(V) &\leq \min\left\{ D\left(2\pi_{\text{thr}} - 1; \frac{q^2}{p^2}, B, -\frac{1}{2}\right), D\left(\pi_{\text{thr}}; \frac{q}{p}, 2B, -\frac{1}{4}\right)\right\} p \\ &\leq \frac{q^2}{(2\pi_{\text{thr}} - 1)p} \end{aligned}$$

Improved Stability Selection (Shah and Samworth 2013)

- Tighter, i.e., less conservative error bounds can be derived under certain conditions.

  a) **If distribution of (simultaneous) selection probabilities is unimodal:**

$$\mathbb{E}(V) \leq \frac{q^2}{c(\pi_{\text{thr}}, B) \cdot p} \leq \frac{q^2}{(2\pi_{\text{thr}} - 1)p}$$

  b) **If distribution of (simultaneous) selection probabilities is r-concave:**

$$\begin{aligned} \mathbb{E}(V) &\leq \min\left\{ D\left(2\pi_{\text{thr}} - 1; \frac{q^2}{p^2}, B, -\frac{1}{2}\right), D\left(\pi_{\text{thr}}; \frac{q}{p}, 2B, -\frac{1}{4}\right) \right\} p \\ &\leq \frac{q^2}{(2\pi_{\text{thr}} - 1)p} \end{aligned}$$

Condition b) is stronger than a) and might not always hold, especially for larger numbers of subsamples B. Thus a) is usually recommended.

Implementation

- Stability selection is implemented in the *R* package **mboost** (Hothorn et al. 2014, Hofner et al. 2014b) in the function

  ```
  stabsel()
  ```

- **mboost** also implements

  ```
  stabsel_parameters(cutoff, q, PFER)
  ```

  to compute error bounds for combinations of two of the three parameters without running the resampling algorithm.

- Stability selection can also be used with other fitting functions.

Implementation

- Stability selection is implemented in the *R* package **mboost** (Hothorn et al. 2014, Hofner et al. 2014b) in the function

  `stabsel()`

- **mboost** also implements

  `stabsel_parameters(cutoff, q, PFER)`

  to compute error bounds for combinations of two of the three parameters without running the resampling algorithm.

- Stability selection can also be used with other fitting functions.

Practical recommendation:

- Choose an upper bound for the PFER and specify *either* q or π_{thr}.
- Check that the computed value is sensible (e.g., that is q large enough if π_{thr} and PFER were specified).

Identification of Biomarkers for ASD patients (yes/no)

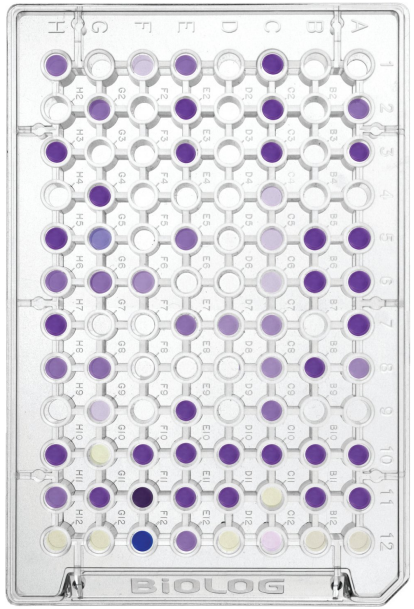
1) Obtain amino acid pathway annotation for each well
   using the R package **opm** (Vaas et al. 2013)
2) Fit main effects model for maximum reaction (`y`),
   given disease status (`group`), pathway annotation (`pathway`), and patient (`id`)

```
y ~ group + pathway + (1 | id)
```

   - Each well constitutes one observation!
   - Each well can belong to multiple pathways!
3) Use 2) as offset model and add group specific effects:

```
y ~ ... + pathway:group
```

4) **Which of the group-specific pathway effects is selected additionally to the offset model (with PFER ≤ 1)?**



Source: Biolog Inc.

Biomarkers for ASD

Model Fitting

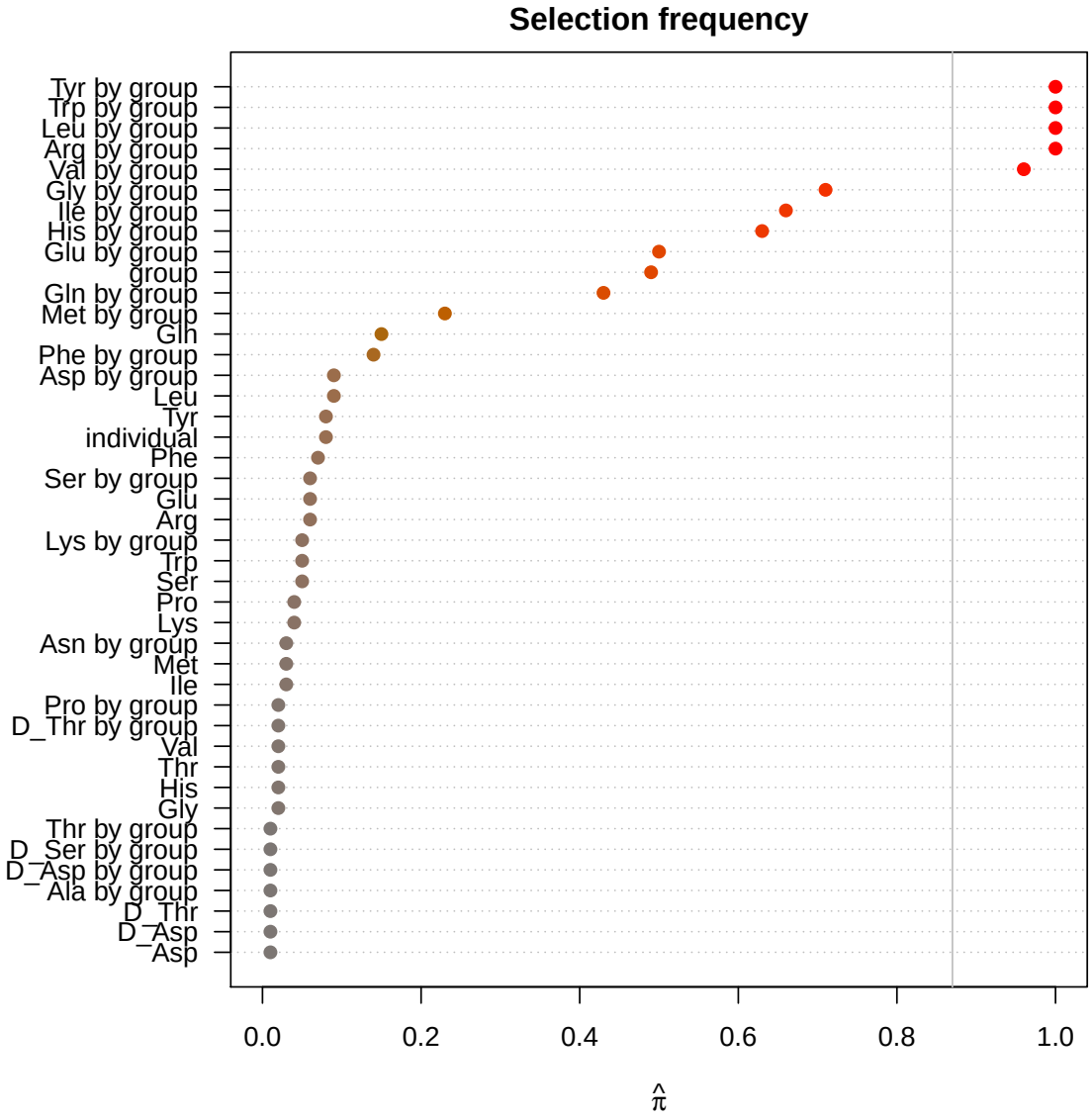
```
## fit offset model with main effects only (with very many iterations):
offsetmod <- gamboost(Value ~ ., data = data, baselearner = bols,
                      control = boost_control(mstop = 5000, nu = 0.2))

## now start from the offset model and add interaction effects
## (which represent differences between strains).
options(contrasts = c("contr.sum", "contr.poly"))
mod <- gamboost(fm, data = data, baselearner = bols,
                offset = fitted(offsetmod),
                control = boost_control(mstop = 500))

## use stability selection to extract differential AAs
stabsel_parameters(p = length(variable.names(mod)), q = 10, PFER = 1)
```

```
Stability Selection with unimodality assumption
Cutoff: 0.87; q: 10; PFER(*):  0.963
   (*) or expected number of low selection probability variables
```

```
stab <- stabsel(mod, q = 10, PFER = 1, mc.cores = 4)
```

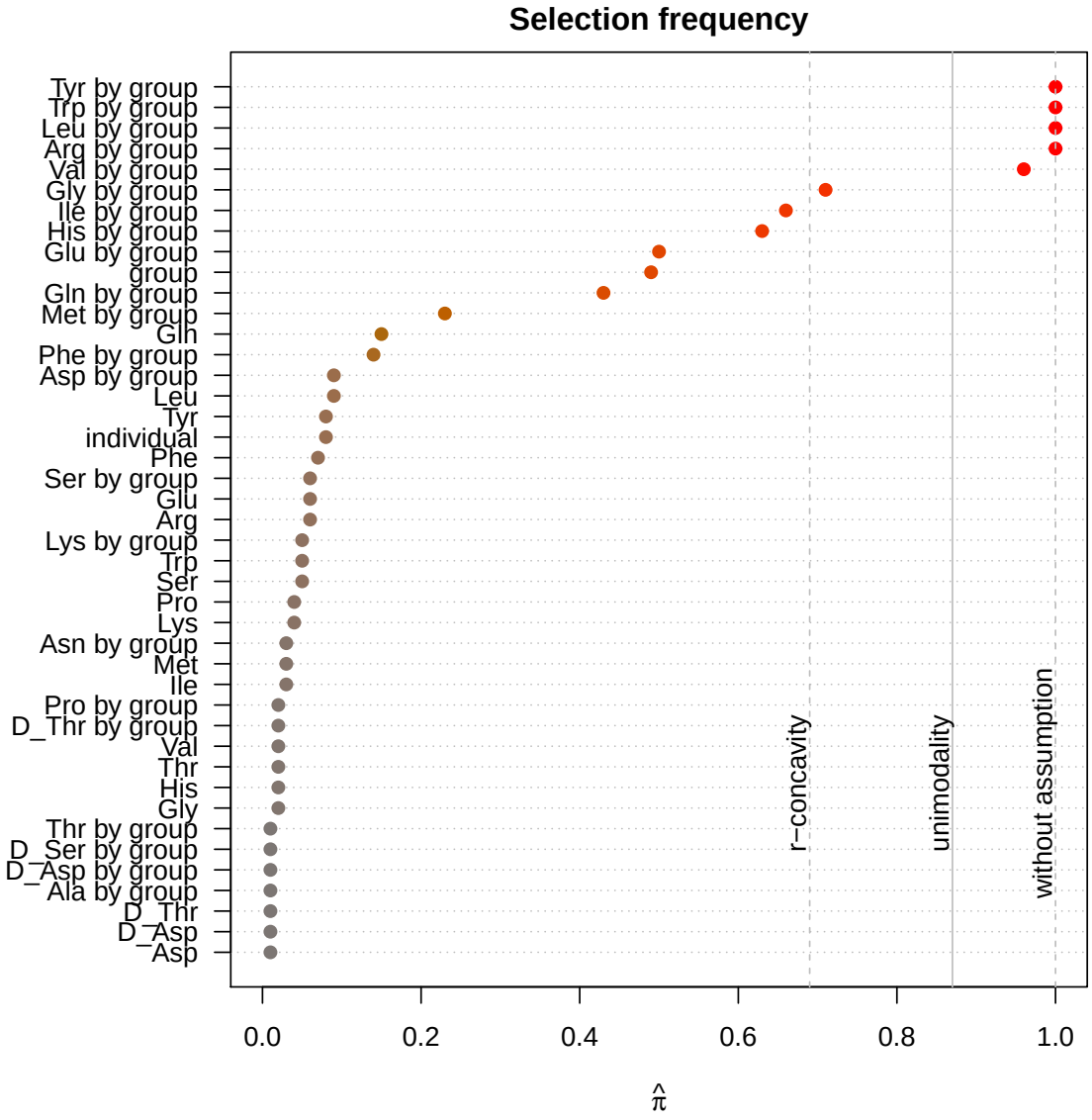
Results: Biomarkers for ASD

Stability selection with PFER ≤ 1 and $q = 10$



Results: Biomarkers for ASD

Stability selection with PFER ≤ 1 and $q = 10$



Results: Biomarkers for ASD

Differentially Expressed Amino Acids

- tyrosine (`Tyr`), tryptophan (`Trp`), leucine (`Leu`), arginine (`Arg`)
  - selection frequency $\hat{\pi} = 100\%$
- valine (`Val`)
  - selection frequency $\hat{\pi} = 96\%$
- glycine (`Gly`)
  - selection frequency $\hat{\pi} = 71\%$

Results: Biomarkers for ASD

Differentially Expressed Amino Acids

- tyrosine (`Tyr`), tryptophan (`Trp`), leucine (`Leu`), arginine (`Arg`)
  - selection frequency $\hat{\pi} = 100\%$
- valine (`Val`)
  - selection frequency $\hat{\pi} = 96\%$
- glycine (`Gly`)
  - selection frequency $\hat{\pi} = 71\%$

Biomedical Conclusion

- Confirms abnormal metabolism of tryptophan in ASD cells (see Boccuto et al. 2013)
- \+ Additional amino acids seem to be affected, although on a milder level
- Suggest an abnormal metabolism of large amino acids

Take-Away Messages

- Stability selection works well in conjunction with boosting.
- It controls the PFER and is especially useful in sparse, high-dimensional settings.
- Stability selection results in a fundamentally new solution, which cannot be recreated elsewise

  (e.g. by selecting a certain regularization parameter, here the stopping iteration m_{stop}).
- Stability selection can also be used for boosted GAM models and other fitting approaches.

- Yet, stability selection is quite conservative
  - as it controls the PFER
  - and as even this control seems to be conservative (at least for the standard error bound).
- Higher selection numbers (i.e. higher TPR) can be obtained by tighter error bounds, yet, sometimes error bounds do not hold any more.

Summary and Outlook

- One can fit a wide range of models by boosting:
  (generalized) linear models, survival models, ...
  (generalized) additive models, structured additive models, ...
- Variable and model selection is possible (and easy).

- R-package **mboost** available on CRAN (Hothorn et al. 2014) to fit all the models covered in this talk (and many more).
  - Tutorial available (Hofner et al. 2014b)

- Extension to GAMLSS models via package **gamboostLSS** (Hofner et al. 2014a)
  - Tutorial available (Hofner et al. 2014c)

Slides and further information available from
`http://benjaminhofner.de`

References I

L Boccuto, C-F Chen, A Pittman, C Skinner, H McCartney, K Jones, B Bochner, R Stevenson, and C Schwartz. Decreased tryptophan metabolism in patients with autism spectrum disorders. *Molecular Autism*, 4(1):16, 2013.

A Buja, T Hastie, and R Tibshirani. Linear smoothers and additive models (with discussion). *The Annals of Statistics*, 17:453–555, 1989.

S Dudoit, J Popper Shaffer, and JC Boldrick. Multiple hypothesis testing in microarray experiments. *Statistical Science*, 18:71–103, 2003.

PHC Eilers and BD Marx. Flexible smoothing with B-splines and penalties (with discussion). *Statistical Science*, 11:89–121, 1996.

L Fahrmeir, T Kneib, and S Lang. Penalized structured additive regression: A Bayesian perspective. *Statistica Sinica*, 14:731–761, 2004.

J Gertheiss and G Tutz. Penalized regression with ordinal predictors. *International Statistical Review*, 77:345–365, 2009.

B Hofner, T Hothorn, T Kneib, and M Schmid. A framework for unbiased model selection based on boosting. *Journal of Computational and Graphical Statistics*, 20:956–971, 2011.

References II

B Hofner, A Mayr, N Fenske, and M Schmid. *gamboostLSS: Boosting Methods for GAMLSS Models*, 2014a. URL http://CRAN.R-project.org/package=gamboostLSS. R package version 1.1-2.

B Hofner, A Mayr, N Robinzonov, and M Schmid. Model-based boosting in R – A hands-on tutorial using the R package mboost. *Computational Statistics*, 29:3–35, 2014b.

B Hofner, A Mayr, and M Schmid. gamboostLSS: An R package for model building and variable selection in the GAMLSS framework. arXiv:1407.1774, 2014c. URL http://arxiv.org/abs/1407.1774.

T Hothorn, P Bühlmann, T Kneib, M Schmid, and B Hofner. *mboost: Model-Based Boosting*, 2014. URL http://CRAN.R-project.org/package=mboost. R package version 2.3-0.

T Kneib, T Hothorn, and G Tutz. Variable selection and model choice in geoadditive regression models. *Biometrics*, 65:626–634, 2009.

A Mayr, B Hofner, and M Schmid. The importance of knowing when to stop – a sequential stopping rule for component-wise gradient boosting. *Methods of Information in Medicine*, 51:178–186, 2012.

References III

N Meinshausen and P Bühlmann. Stability selection (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72: 417–473, 2010.

A Rosenwald, G Wright, WC Chan, and others. The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma. *New England Journal of Medicine*, 346(25):1937–1947, 2002.

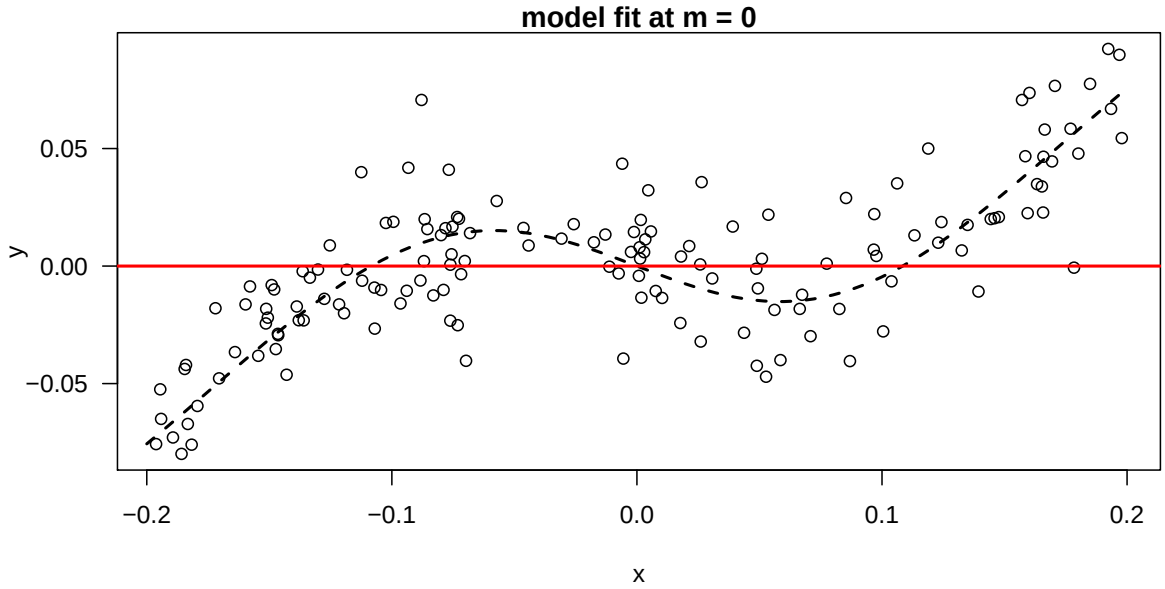
RD Shah and RJ Samworth. Variable selection with error control: another look at stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75:55–80, 2013.

LAI Vaas, J Sikorski, B Hofner, N Buddruhs, A Fiebig, H-P Klenk, and M Göker. opm: An R package for analysing Omnilog®Phenotype MicroArray data. *Bioinformatics*, 29(14):1823–1824, 2013.

Figure Sources

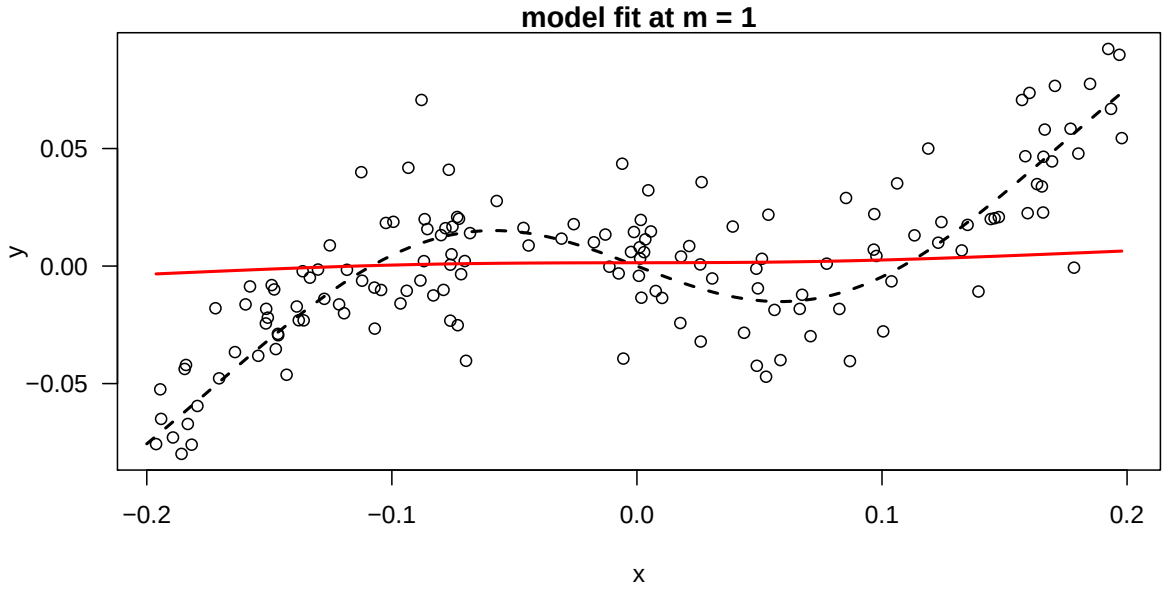
- Page 30: Based on a picture by Nepenthes, CC BY-SA 3.0, Source: Wikipedia
- Page 32 (Beeches): Picture by Darkone, CC BY-SA 1.0, Source: Wikipedia
- Page 32 (Spessart): Based on a picture by Carport, derivative work by Milseburg, CC BY-SA 3.0, Source: Wikipedia

Estimating Smooth Effects with Boosting



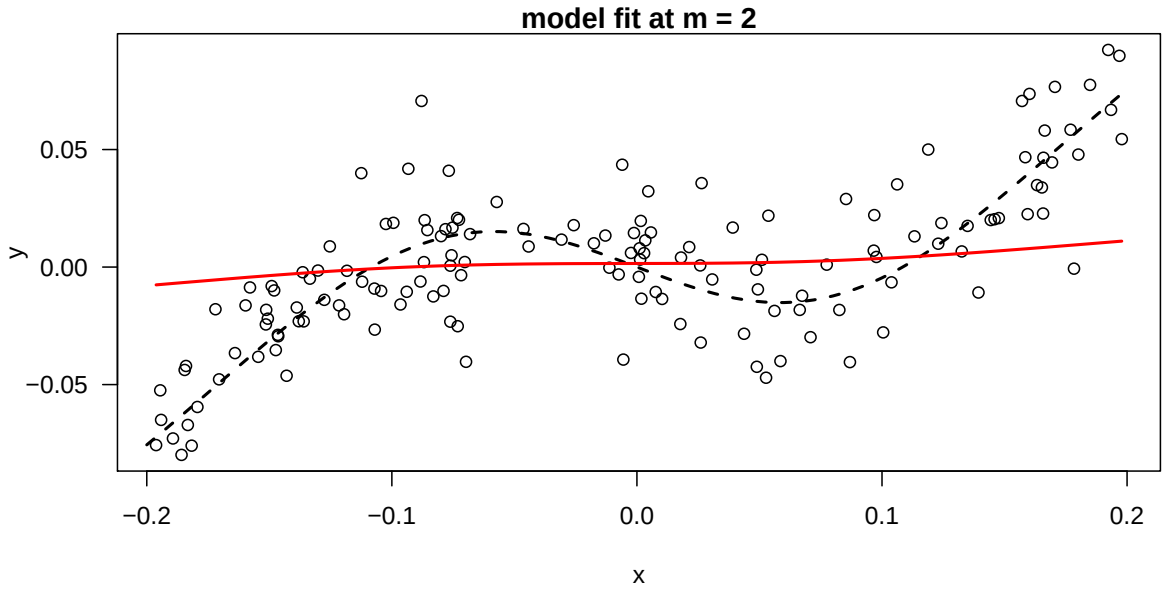
▸ Jump Back

Estimating Smooth Effects with Boosting



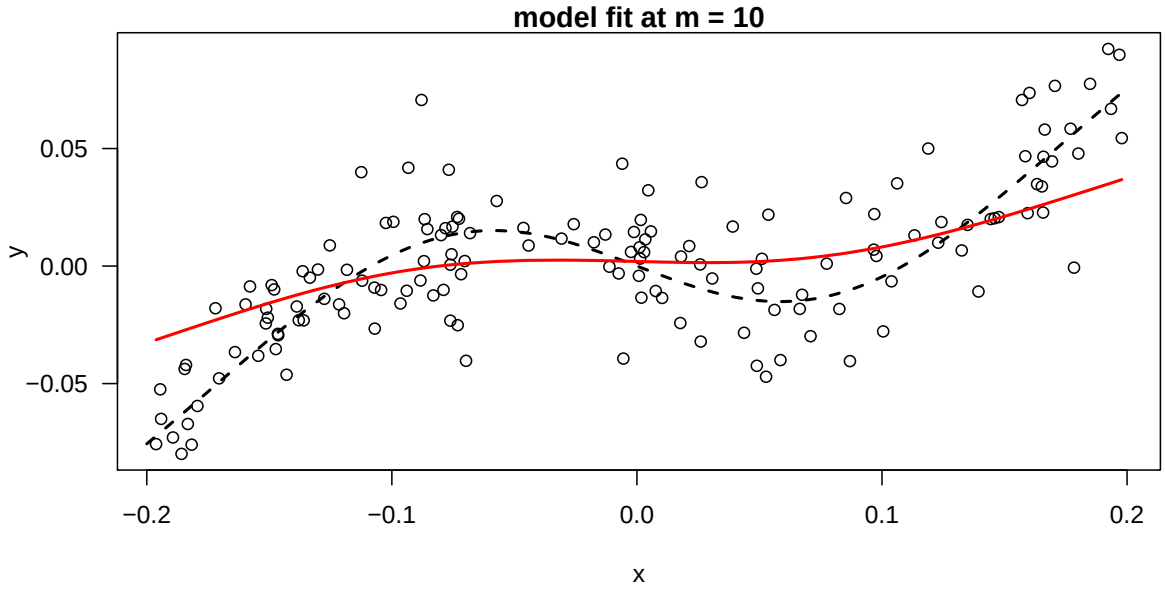
▸ Jump Back

Estimating Smooth Effects with Boosting



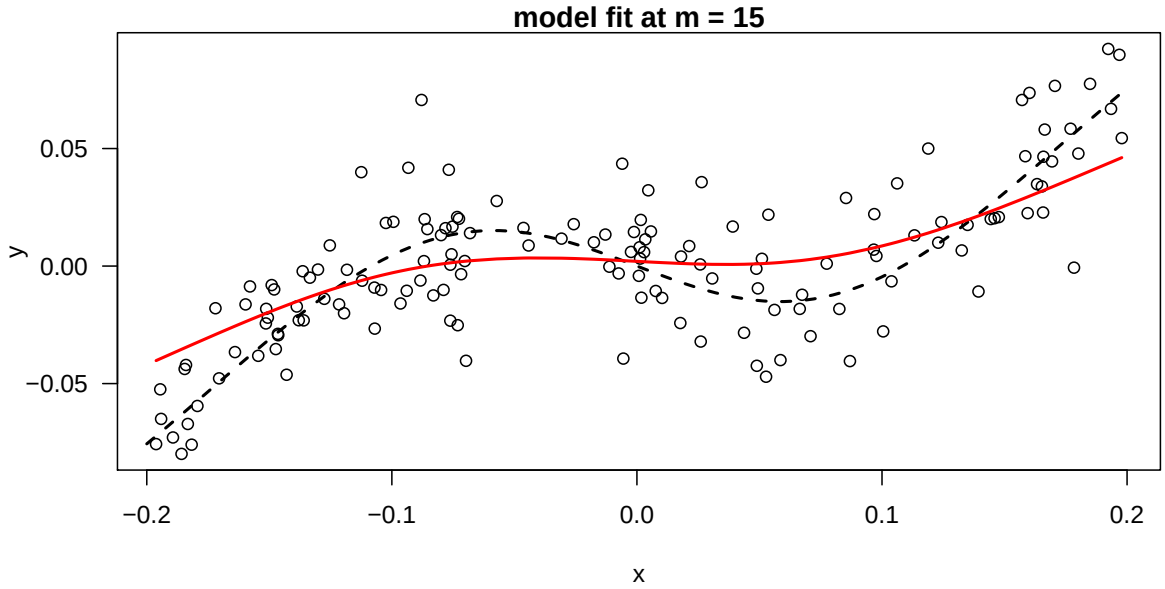
▸ Jump Back

Estimating Smooth Effects with Boosting



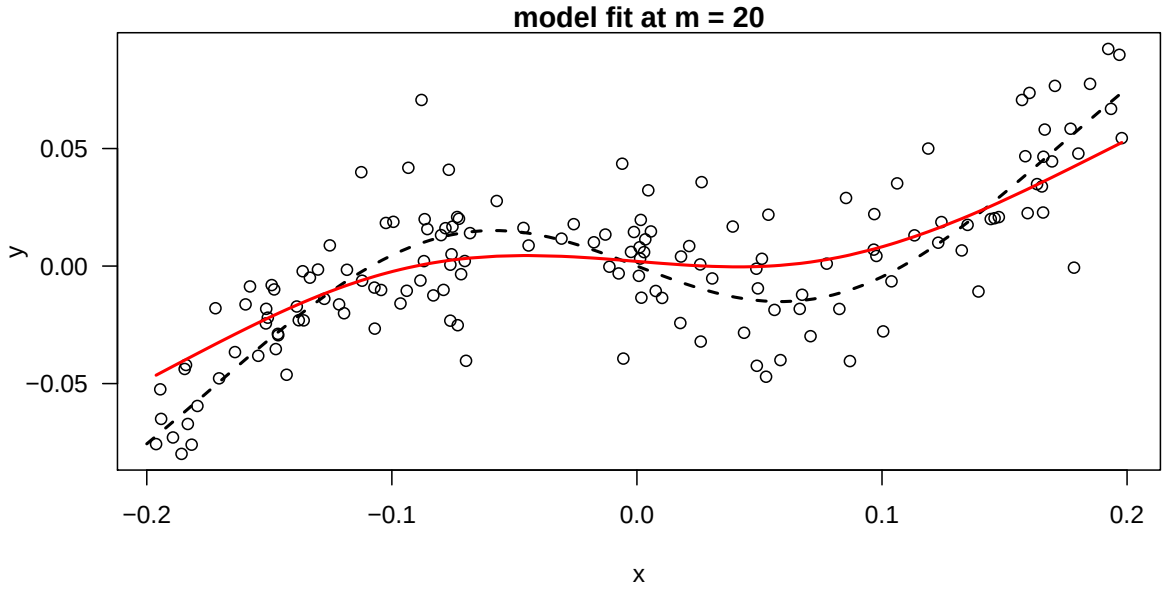
▸ Jump Back

Estimating Smooth Effects with Boosting



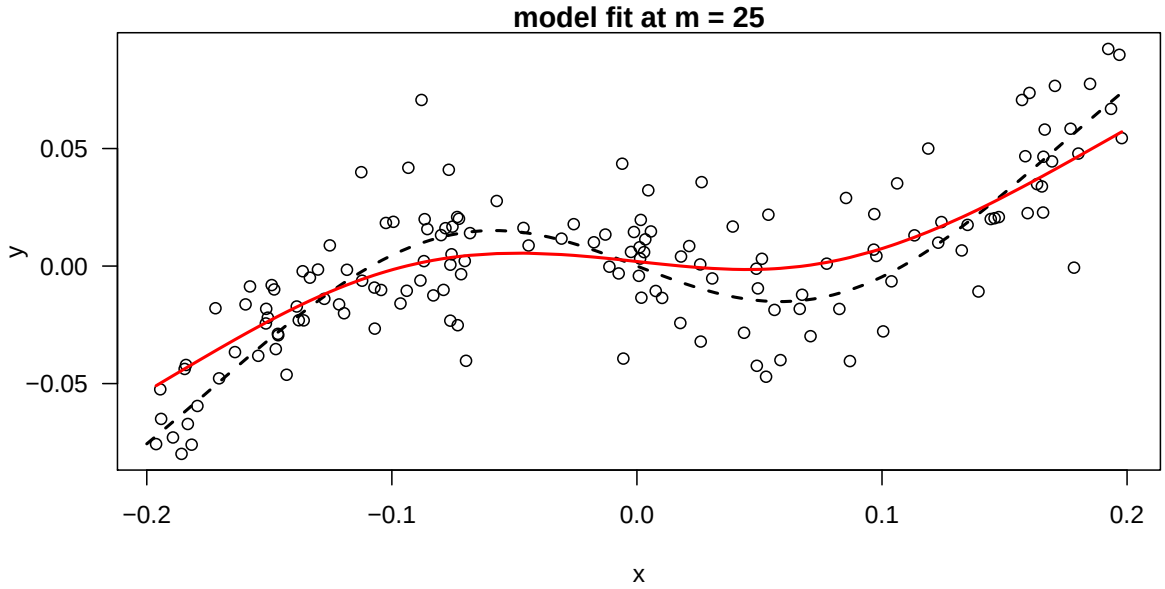
▸ Jump Back

Estimating Smooth Effects with Boosting



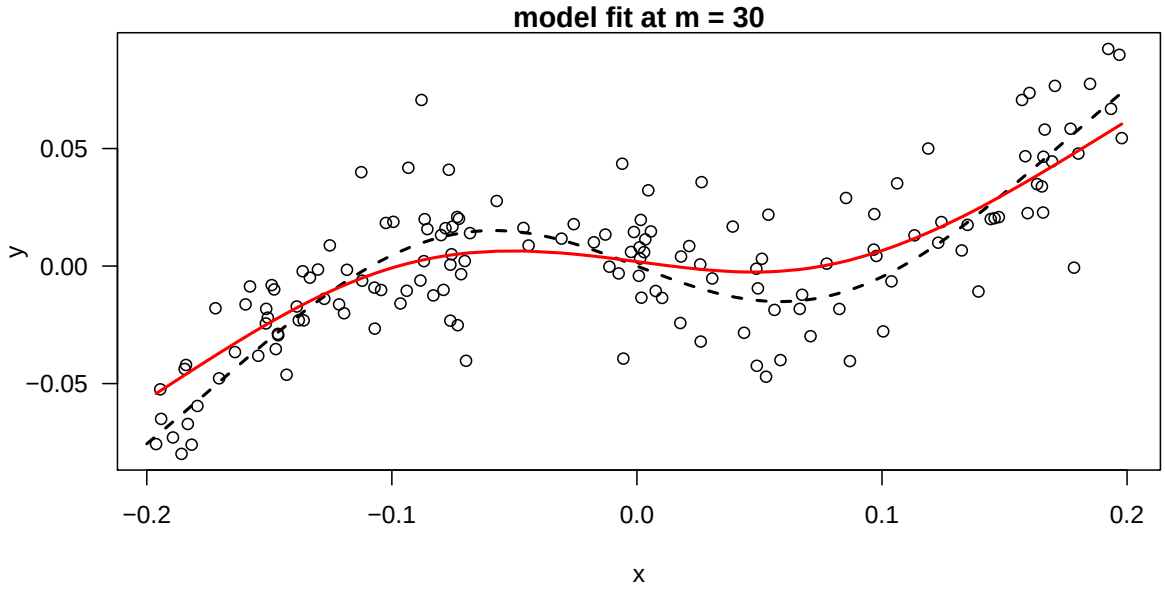
▸ Jump Back

Estimating Smooth Effects with Boosting



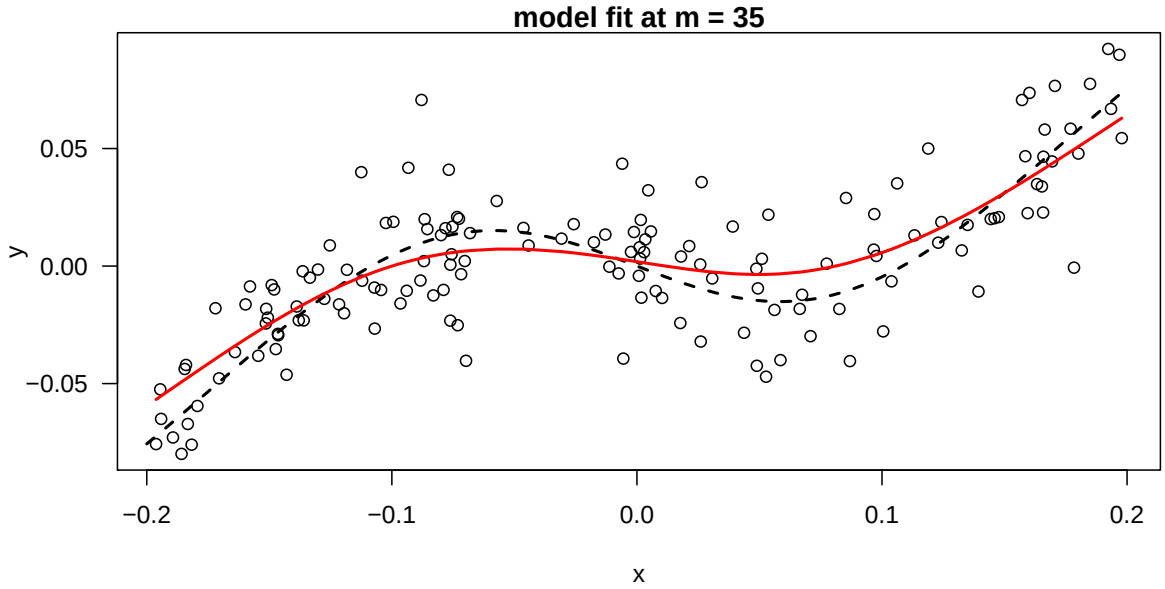
▸ Jump Back

Estimating Smooth Effects with Boosting



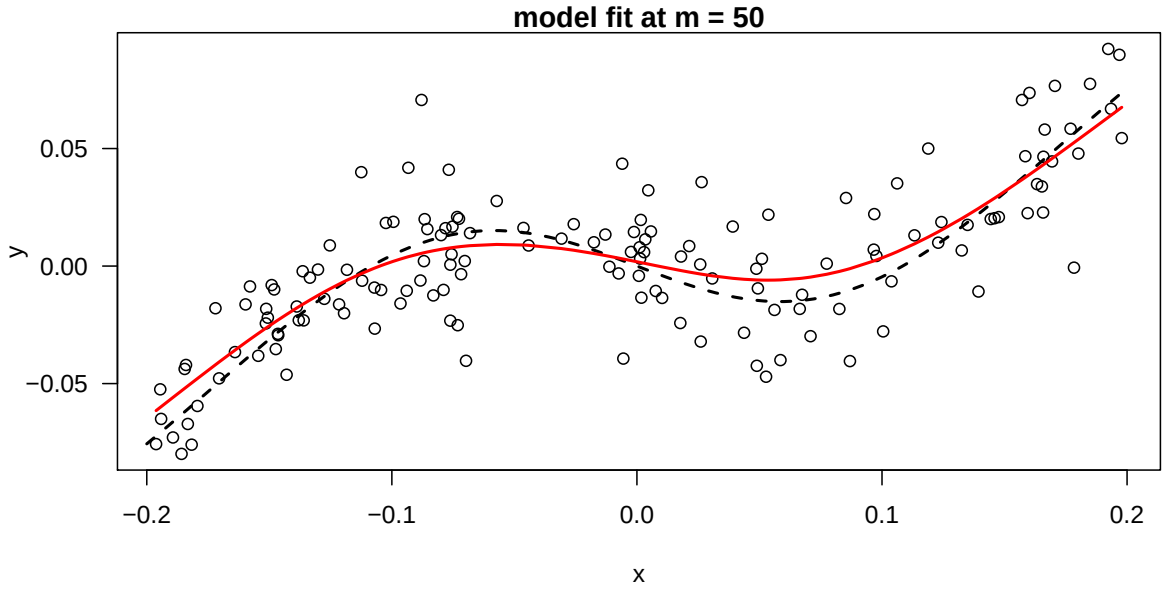
▸ Jump Back

Estimating Smooth Effects with Boosting



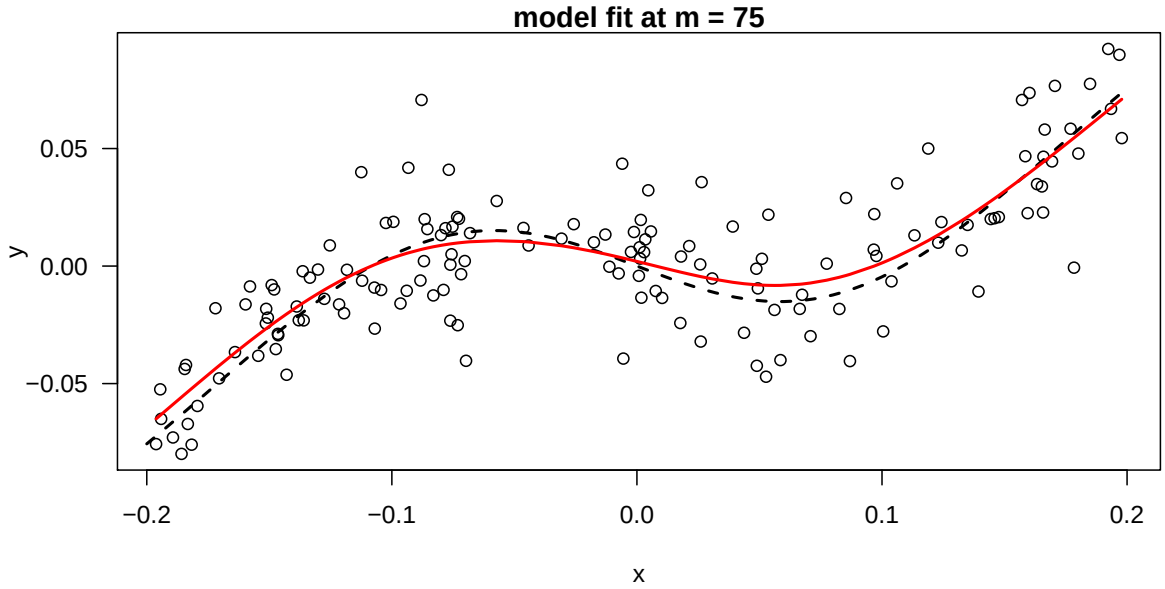
▸ Jump Back

Estimating Smooth Effects with Boosting



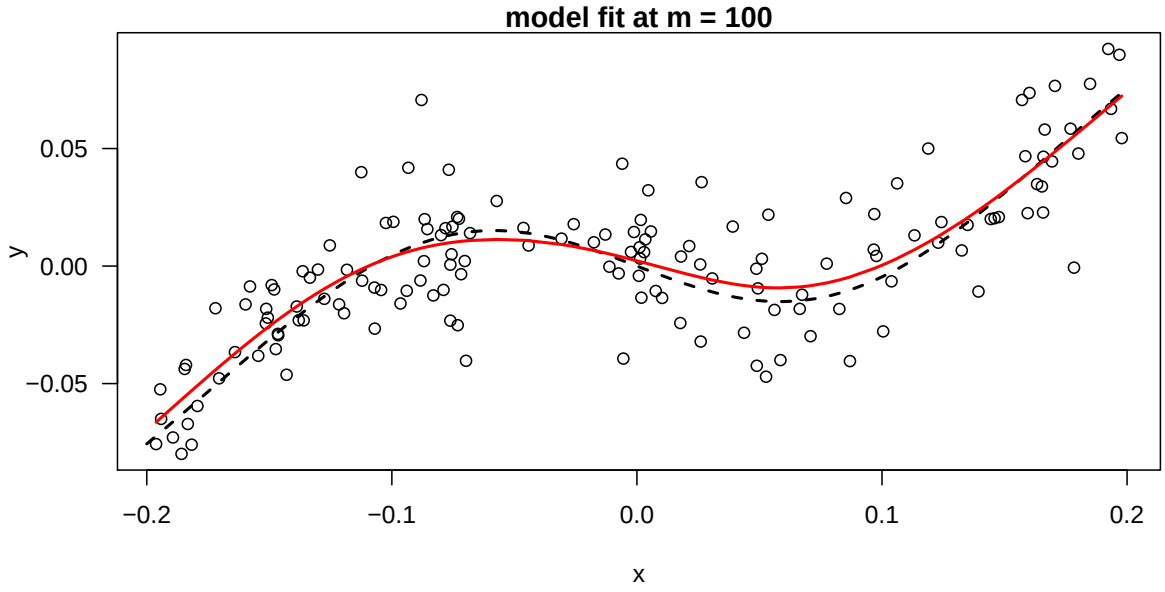
▸ Jump Back

Estimating Smooth Effects with Boosting



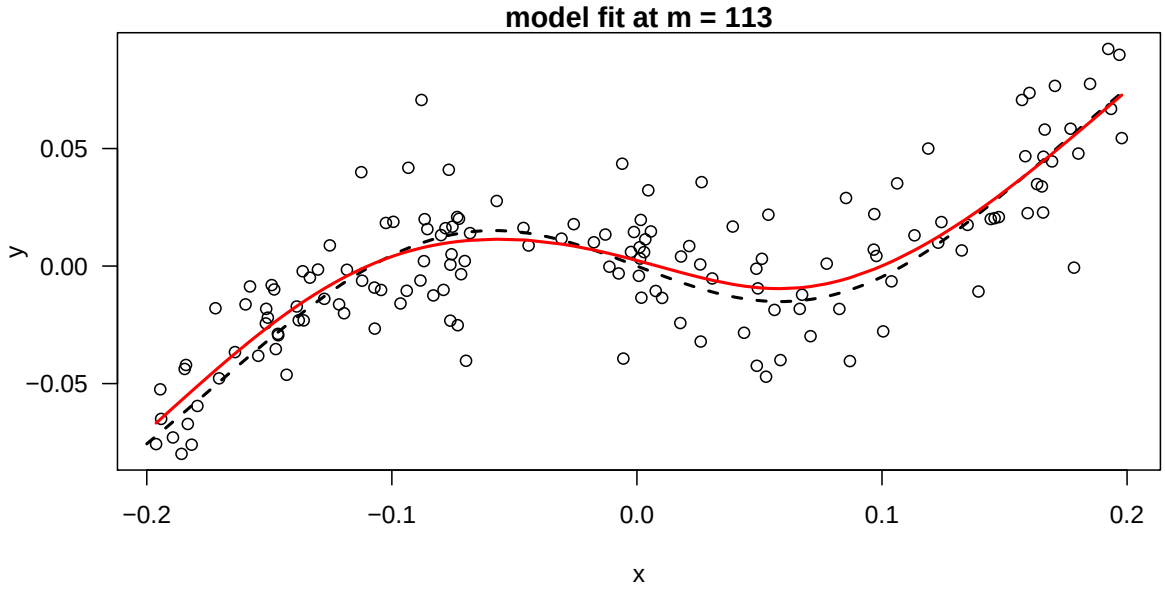
▸ Jump Back

Estimating Smooth Effects with Boosting



▸ Jump Back

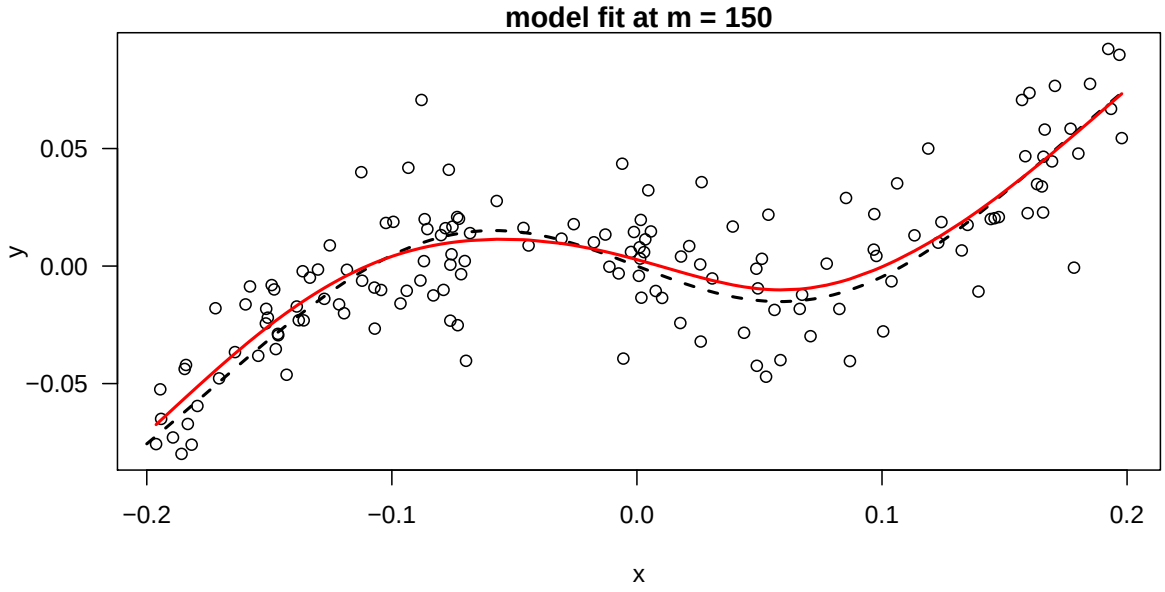
Estimating Smooth Effects with Boosting



- optimal mstop

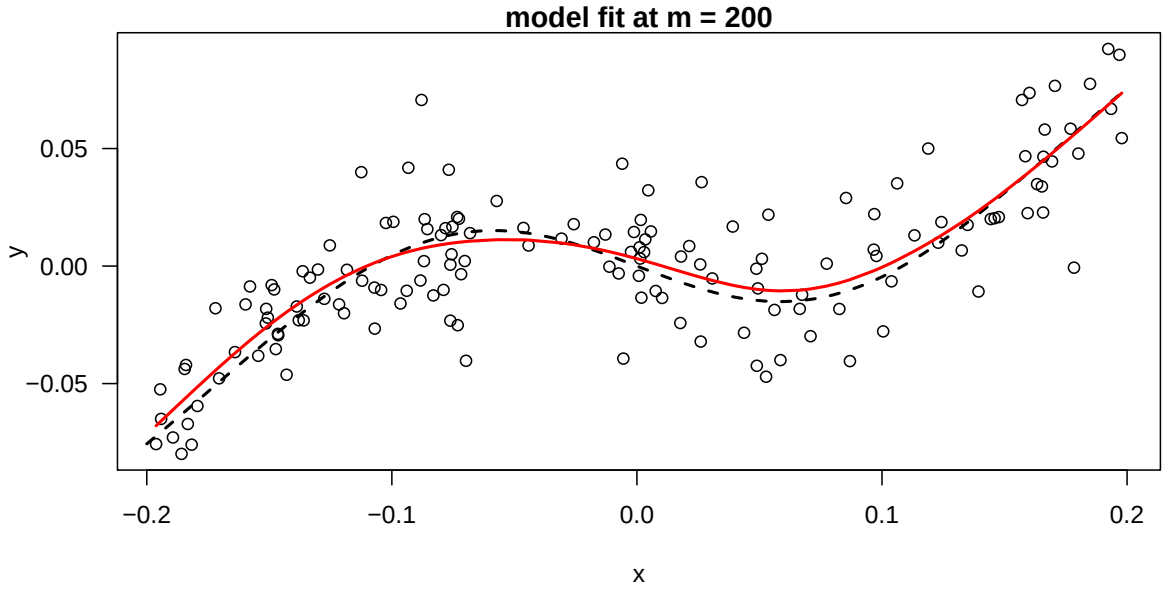
▸ Jump Back

Estimating Smooth Effects with Boosting



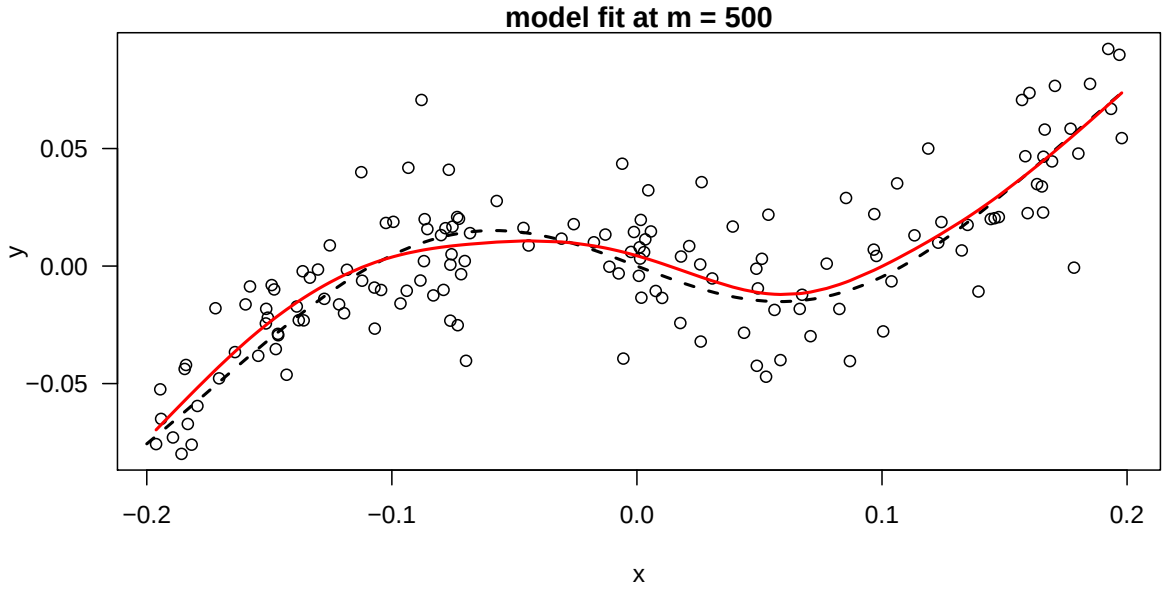
▸ Jump Back

Estimating Smooth Effects with Boosting



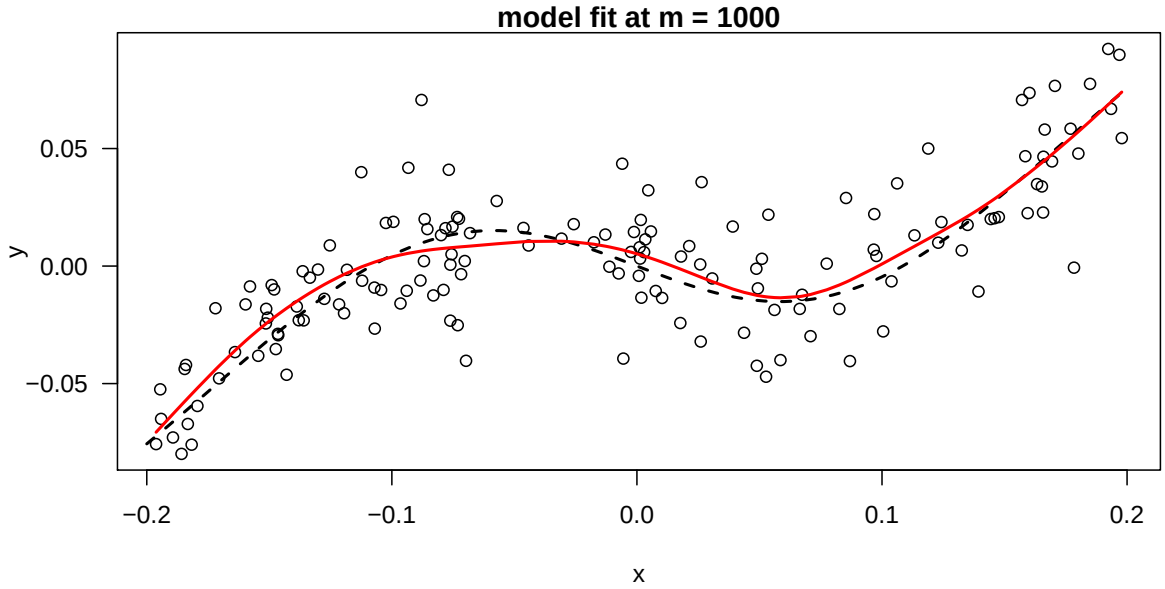
▸ Jump Back

Estimating Smooth Effects with Boosting



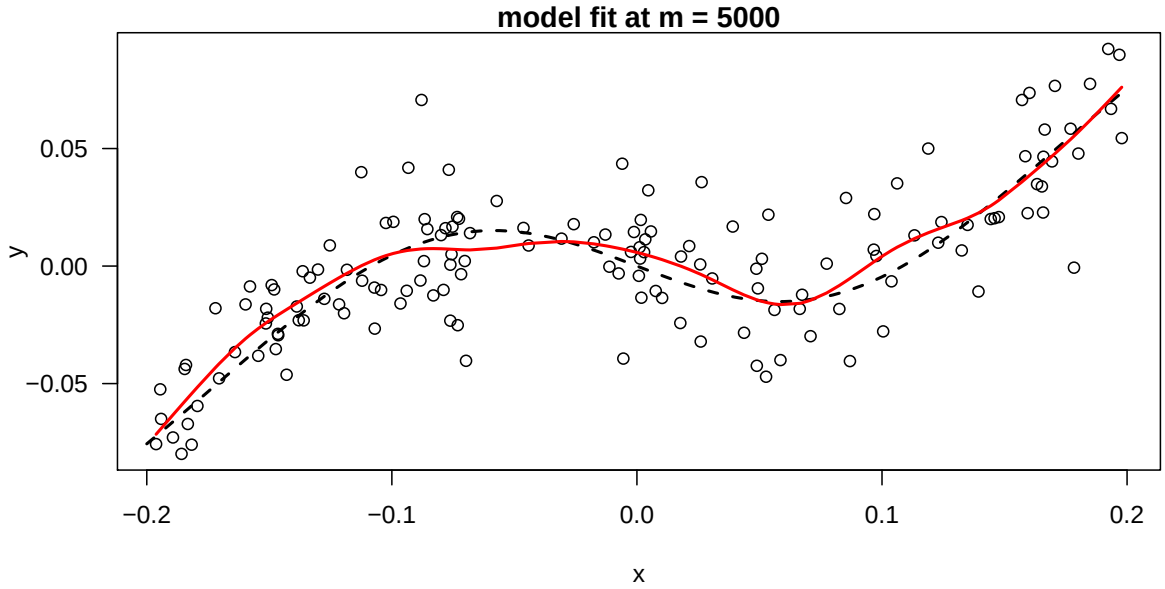
▸ Jump Back

Estimating Smooth Effects with Boosting



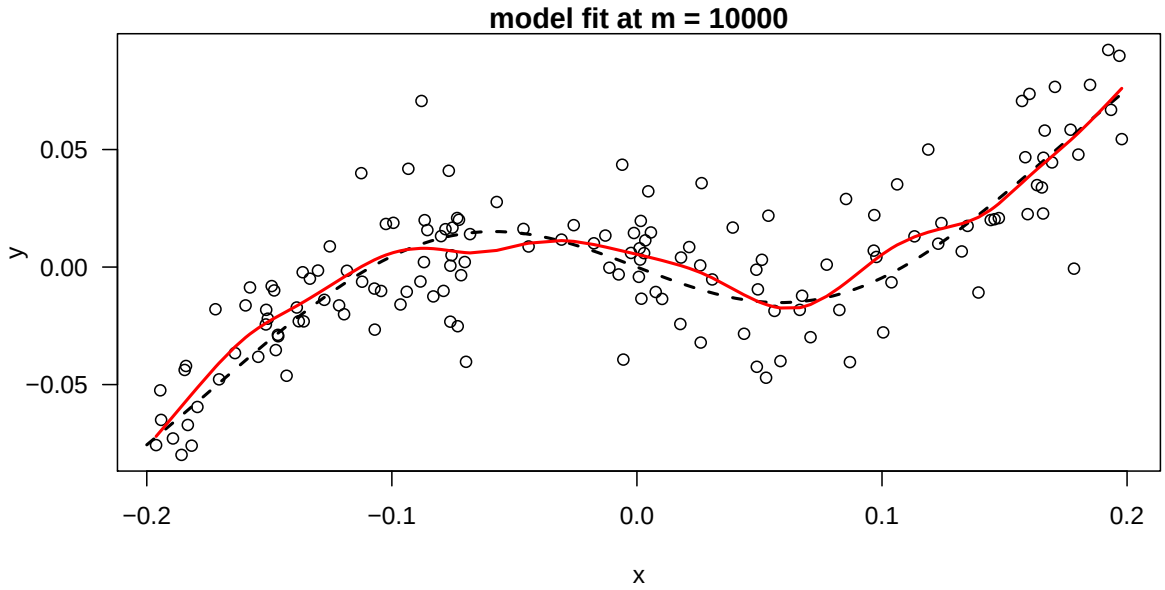
▸ Jump Back

Estimating Smooth Effects with Boosting



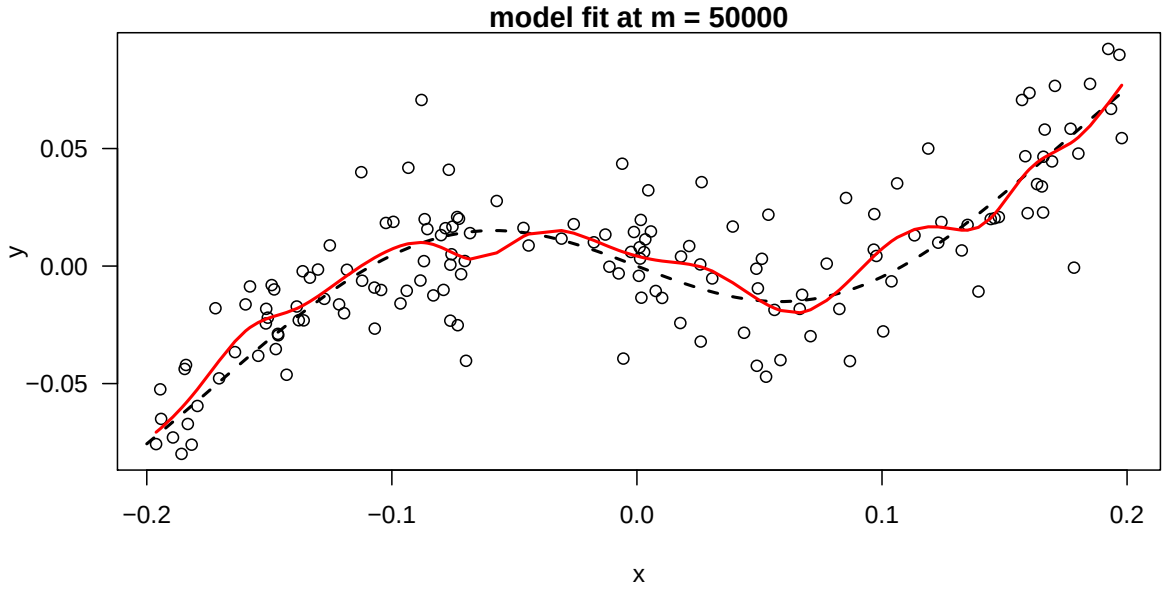
▸ Jump Back

Estimating Smooth Effects with Boosting



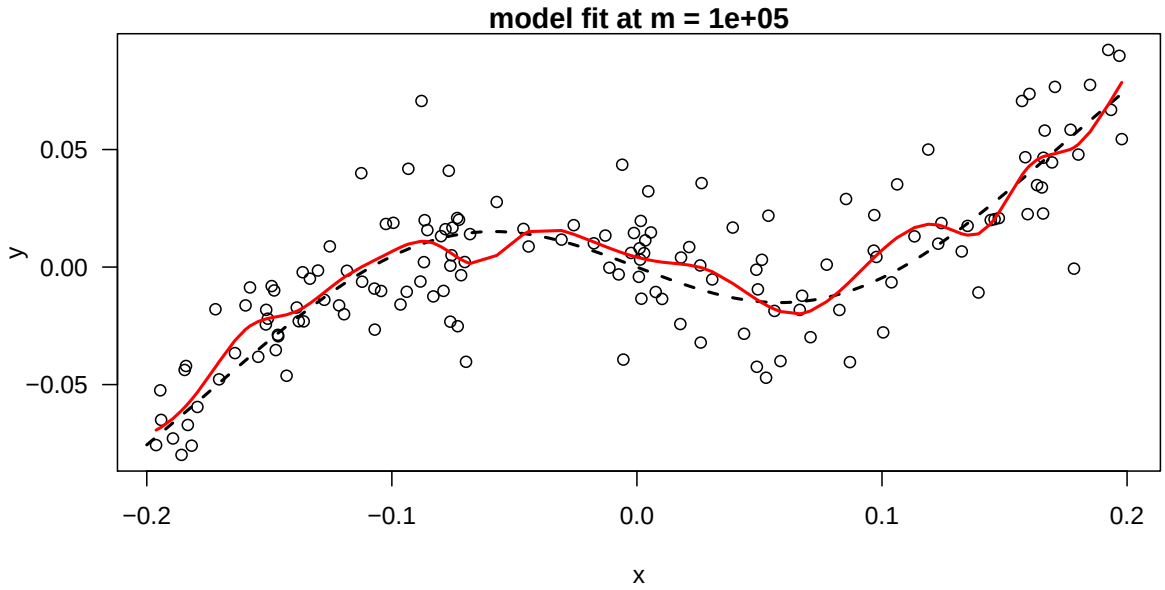
▸ Jump Back

Estimating Smooth Effects with Boosting



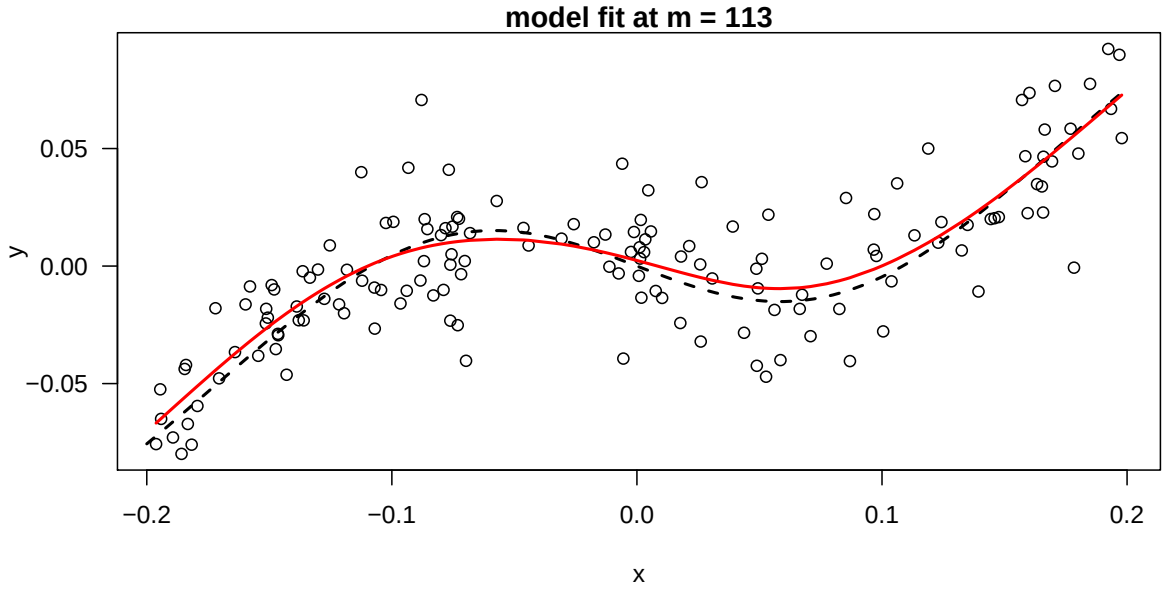
▸ Jump Back

Estimating Smooth Effects with Boosting



▸ Jump Back

Estimating Smooth Effects with Boosting



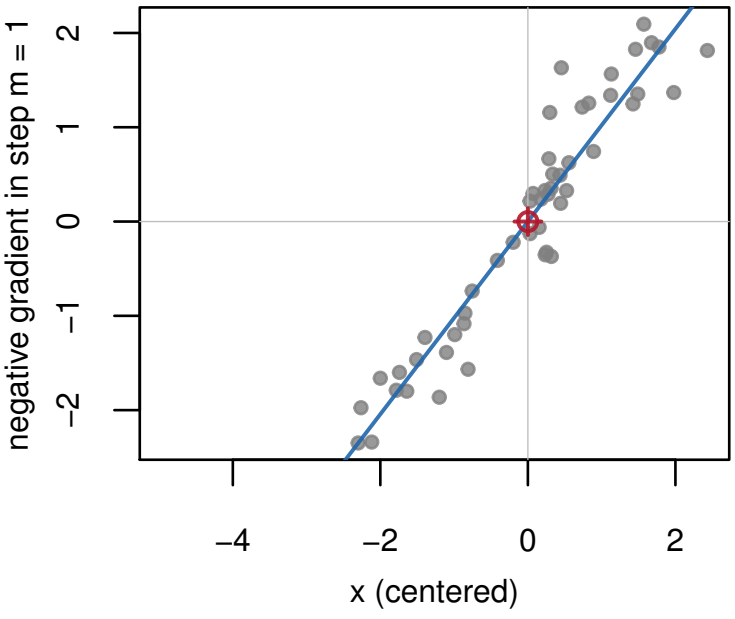
- optimal mstop

- Boosting generally has a slow overfitting behaviour.
- Adaption to higher order degrees of freedom is possible.

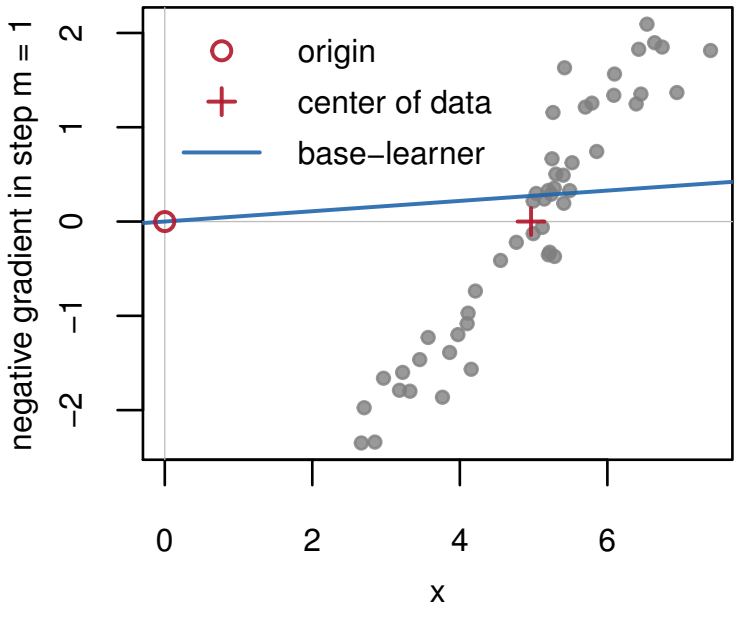
▸ Jump Back

Importance of Centering

Negative gradient and estimated base-learner (without shrinkage factor ν)



covariates centered

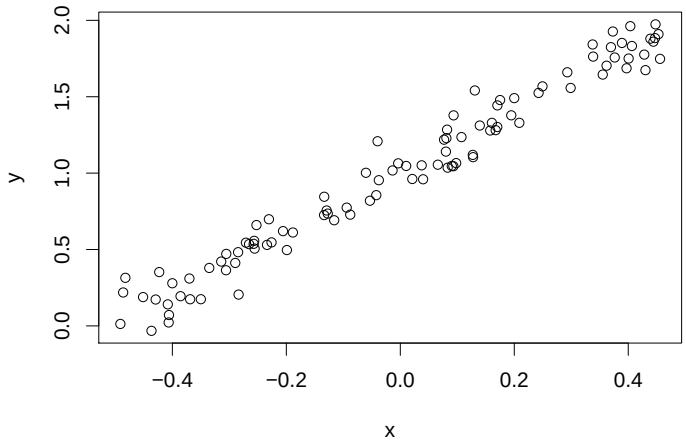


covariates not centered

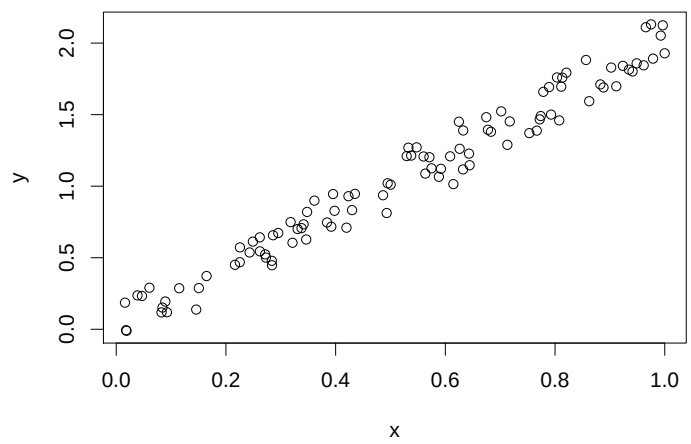
- the base-learner has *no intercept* and thus is forced through the origin (where there is no data without centering)

Importance of Centering (ctd.)

Negative gradient and estimated base-learner (without shrinkage factor ν)



covariates centered



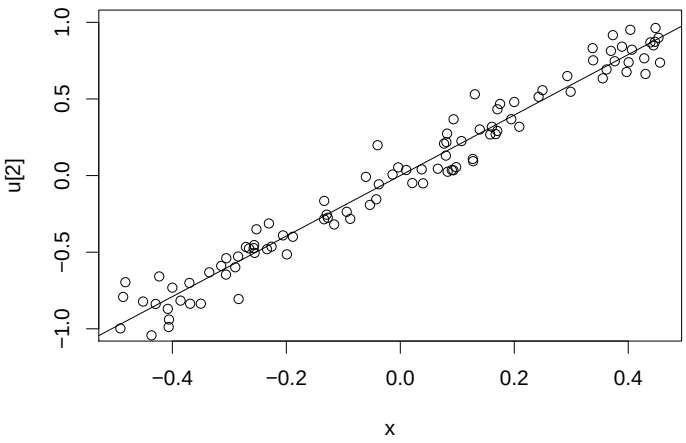
covariates not centered

step 1

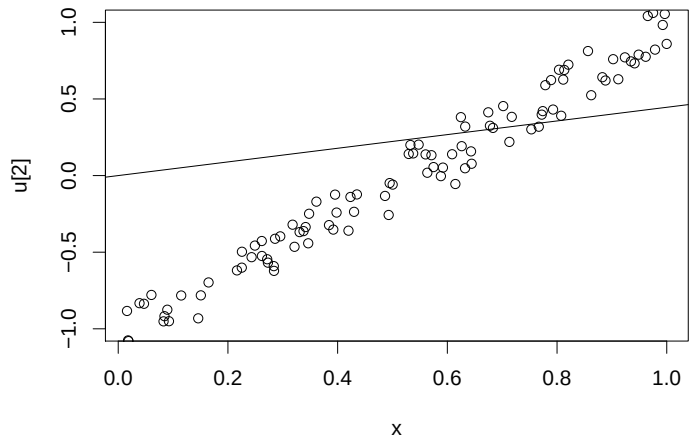
▸ Jump Back

Importance of Centering (ctd.)

Negative gradient and estimated base-learner (without shrinkage factor ν)



covariates centered



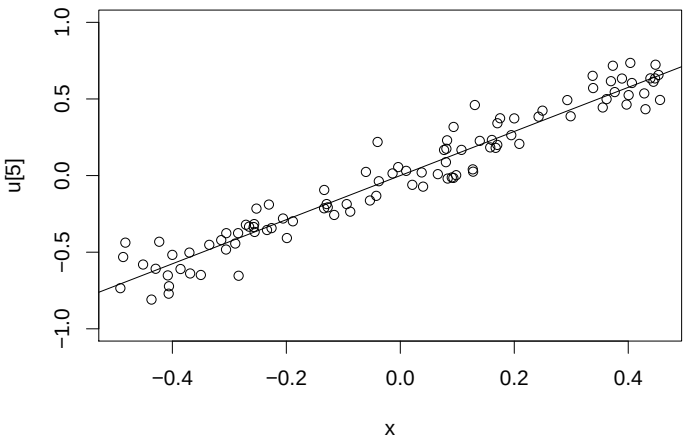
covariates not centered

step 2

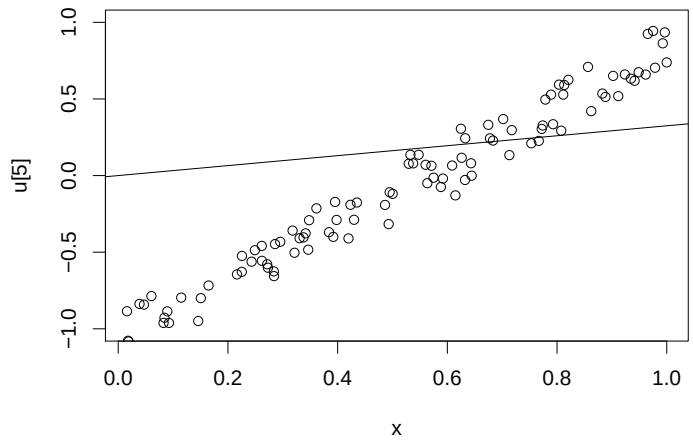
▸ Jump Back

Importance of Centering (ctd.)

Negative gradient and estimated base-learner (without shrinkage factor ν)



covariates centered



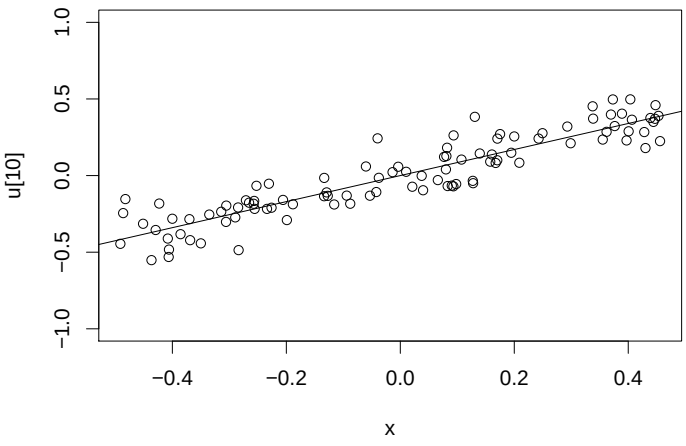
covariates not centered

step 5

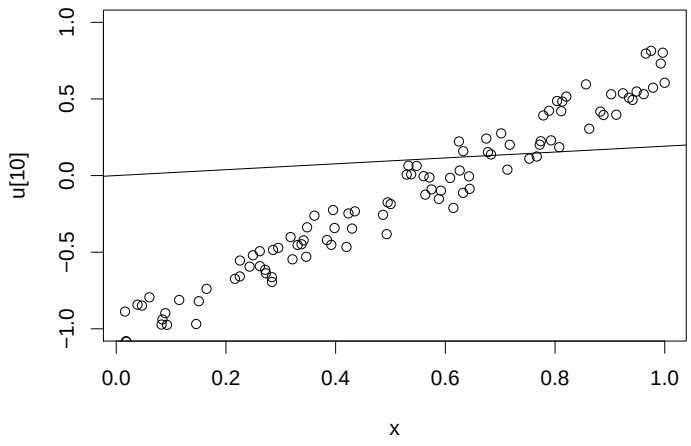
▸ Jump Back

Importance of Centering (ctd.)

Negative gradient and estimated base-learner (without shrinkage factor ν)



covariates centered



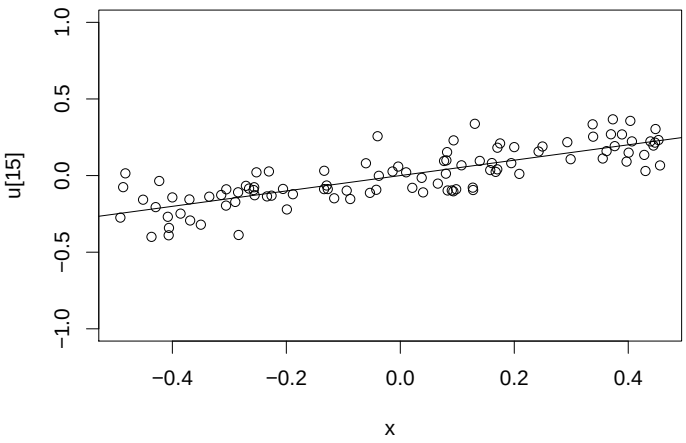
covariates not centered

step 10

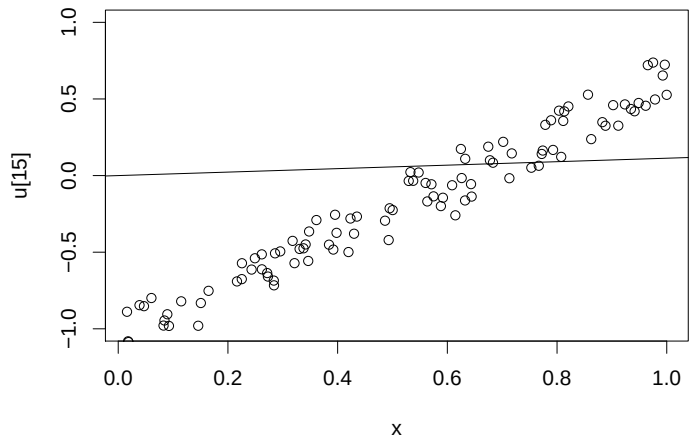
▸ Jump Back

Importance of Centering (ctd.)

Negative gradient and estimated base-learner (without shrinkage factor ν)



covariates centered



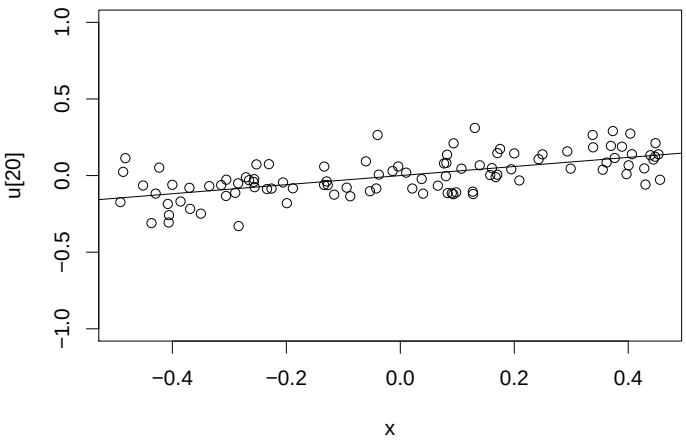
covariates not centered

step 15

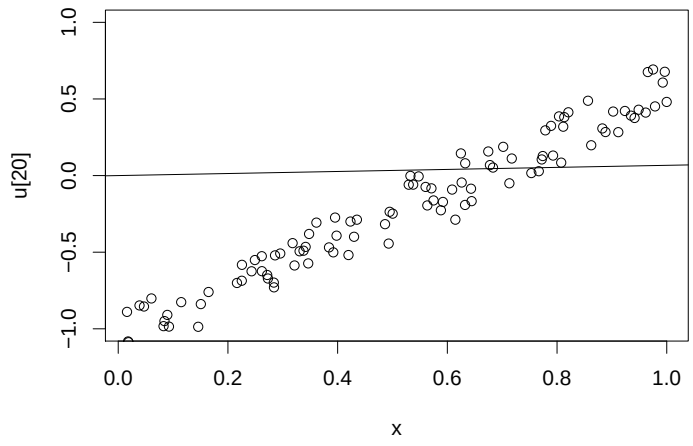
▸ Jump Back

Importance of Centering (ctd.)

Negative gradient and estimated base-learner (without shrinkage factor ν)



covariates centered



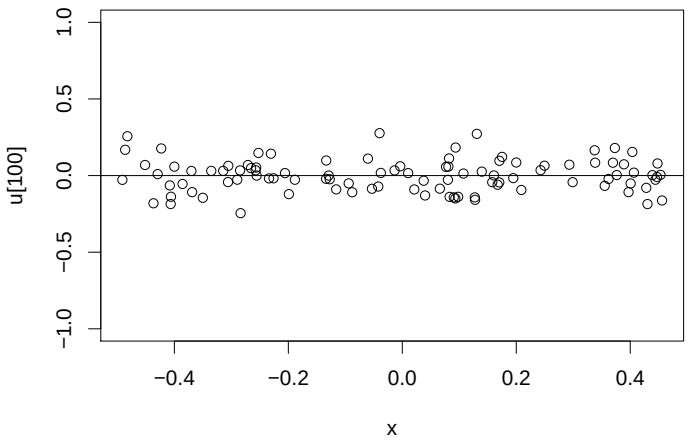
covariates not centered

step 20

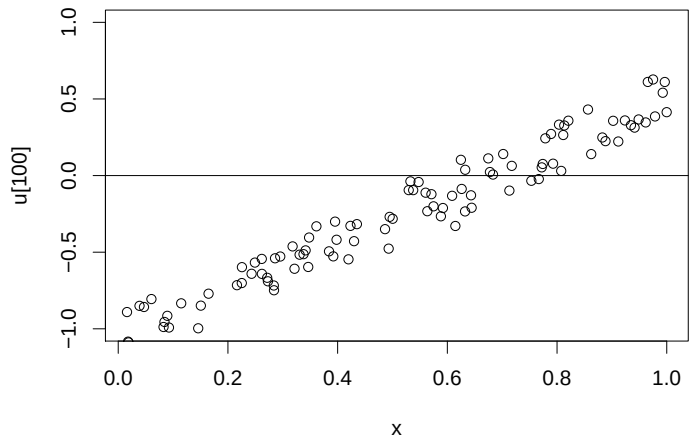
▸ Jump Back

Importance of Centering (ctd.)

Negative gradient and estimated base-learner (without shrinkage factor ν)



covariates centered



covariates not centered

step
100

▸ Jump Back