

Boosting in Structured Additive Models

Benjamin Hofner

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

Disputation am Institut für Statistik
Ludwig-Maximilians-Universität München

05. Dezember 2011*



* slightly improved version

Red Kite Breeding in Bavaria

- **Aim:** Model the probability of breeding for Red Kites (*Milvus milvus*) in Bavaria in order to understand the impact of environmental variables and climatic changes on species distribution
 - **Response:** Red Kite breeding observed (yes/no)
 - **Observations:** 2 periods each with 1918 observational cells of size 40 km² (3836 observations in total)
 - **Observation periods:** 1979–1983 & 1996–1999
 - **Predictors:** Characterizing climatic conditions and land cover
- ▶ Spatio-temporal logistic regression model



- **Possible predictors:**

- 15 binary covariates (presence and absence of characteristics)
- 1 ordinal covariate (fraction of cities and villages [0%, 1-10%, >10%])
- 36 continuous variables

and

- Spatial information (longitude and latitude)
- Temporal information (period of observation)

▶ Variable selection and model choice required

▶ **Additional challenge:**

Some effects should be monotonic for biological reasons.

We will show how boosting . . .

is able to fulfill all these requirements.

Part I

Introduction to Boosting

Model Fitting with Component-Wise Boosting

Structured Additive Model

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

with response function h and **additive** predictor

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

- Model fitting aims at **minimizing the expected loss** with appropriate **loss function** ρ , e.g.,
 - for Gaussian model: **squared error loss** $\rho(y, \eta(\mathbf{x})) = (y - \eta(\mathbf{x}))^2$
 - for GLMs: **negative log-likelihood**
- In practice: Minimization of the **empirical risk**

$$n^{-1} \sum_{i=1}^n \rho(y_i, \eta_i(\mathbf{x}_i))$$

Boosting

- minimizes empirical risk (e.g., **negative log-likelihood**)
- in a stagewise fashion
- via functional gradient descent (FGD).

In each iteration m

- the negative gradient of the loss function

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

is estimated via penalized least squares base-learners ($\hat{u}^{[m]} = \hat{g}_j(\mathbf{x})$)

- update only model term corresponding to the **best-fitting base-learner** \hat{g}_j^* (based on the **RSS**):
 - add a **small fraction** ν of the estimate \hat{g}_j^* (e.g., 10%) to the model
- ▶ variable and model selection is achieved

Practical notes

- Base-learners represent functions $f_j(\cdot)$ from structured additive predictor.
- We get an interpretable model, similar to models from maximum likelihood estimation.
- Additionally, regularization is achieved via base-learner selection and shrinkage.

Practical notes

- Base-learners represent functions $f_j(\cdot)$ from structured additive predictor.
- We get an interpretable model, similar to models from maximum likelihood estimation.
- Additionally, regularization is achieved via base-learner selection and shrinkage.

Implementation

- All results discussed here are implemented in the R package **mboost** [Hothorn, Bühlmann, Kneib, Schmid, and Hofner 2010; 2012]
- The estimation problem is split into two (main) parts:
 - the loss function, which specifies the estimation problem (“family”)
 - the base-learners, which specify the types of effects.

Everything can be freely combined.

Practical notes

- Base-learners represent functions $f_j(\cdot)$ from structured additive predictor.
- We get an interpretable model, similar to models from maximum likelihood estimation.
- Additionally, regularization is achieved via base-learner selection and shrinkage.

Implementation

- All results discussed here are implemented in the R package **mboost** [Hothorn, Bühlmann, Kneib, Schmid, and Hofner 2010; 2012]
- The estimation problem is split into two (main) parts:
 - the loss function, which specifies the estimation problem (“family”)
 - the base-learners, which specify the types of effects.

Everything can be freely combined.

- In the next part we show how improve the fitting algorithm itself and propose improved base-learners for unbiased variable selection.
- In subsequent part we propose new monotonic base-learners to estimate new models.

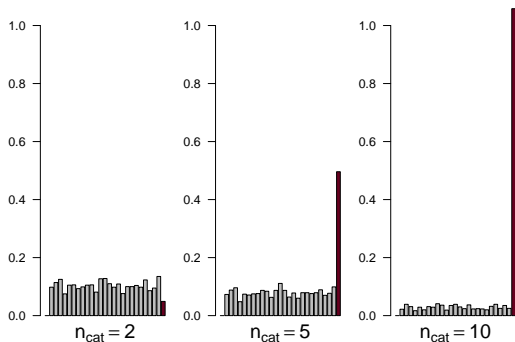
Part II

A Framework for Unbiased Model Selection

based on Hofner et al. [2011a]

Biased Selection: An Illustrative Example

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



Problem (and a Solution)

- The problem of variable selection bias has been widely discussed in the context of regression trees and random forests.
[e.g., Breiman et al. 1984, Loh 2002, Hothorn et al. 2006, Strobl et al. 2007]

In the Boosting Context

- **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 Choice Bias:
e.g., linear effect \prec smooth effect
- Unbiased (or at least improved) selection desired

Problem (and a Solution)

- The problem of variable selection bias has been widely discussed in the context of regression trees and random forests.
[e.g., Breiman et al. 1984, Loh 2002, Hothorn et al. 2006, Strobl et al. 2007]

In the Boosting Context

- **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 Choice 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

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

where \mathbf{X} is a suitable design matrix.

Examples of penalized LS base-learners

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

Penalized Least Squares Base-Learners

Central Idea

Set $df = 1$ for all base-learners to prevent selection bias

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

Penalized Least Squares Base-Learners

Central Idea

Set $df = 1$ 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

► Theorem

Instead of

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

define

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

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

Results

“Null Model” with Non-Informative Factor

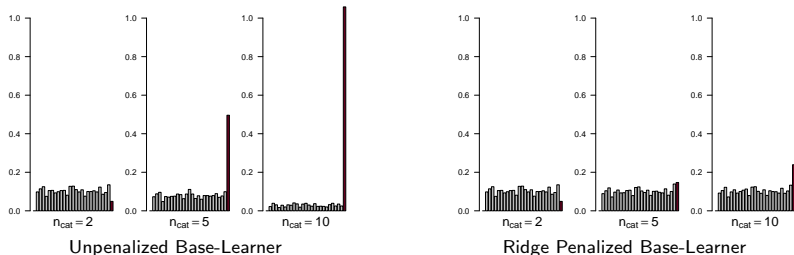
- 25 non-informative continuous covariates
- 1 **non-informative categorical** covariate with increasing # of categories
- $y \sim N(0, 1)$
- $n = 150, B = 1000$

Results

“Null Model” with Non-Informative Factor

- 25 non-informative continuous covariates
- 1 **non-informative categorical** covariate with increasing # of categories
- $y \sim N(0, 1)$
- $n = 150, B = 1000$

Selection Frequencies



Results

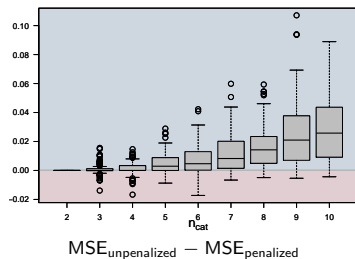
“Power Case” with Non-Informative Factor

- 5 continuous covariates with $\beta_{\text{info}} = (-2, -1, 1, 2, 3)^\top$
20 additional non-informative continuous covariates
- 1 **non-informative categorical** covariate with increasing # of categories
- $y|x \sim N(x^\top \beta, \sigma^2)$, with σ^2 such that $R^2 \approx 0.3$
- $n = 150$, $B = 100$

Results

“Power Case” with Non-Informative Factor

- 5 continuous covariates with $\beta_{\text{info}} = (-2, -1, 1, 2, 3)^\top$
20 additional non-informative continuous covariates
- 1 **non-informative categorical** covariate with increasing # of categories
- $y|x \sim N(x^\top \beta, \sigma^2)$, with σ^2 such that $R^2 \approx 0.3$
- $n = 150$, $B = 100$



Unbiased Selection for P-spline Base-learners

In a similar manner to categorical variables one can show that unbiased estimation requires

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

to be controlled for P-splines as well to achieve unbiased selection.

Unbiased Selection for P-spline Base-learners

In a similar manner to categorical variables one can show that unbiased estimation requires

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

to be controlled for P-splines as well to achieve unbiased selection.

Problem

- We cannot make df of smooth effects arbitrary small, i.e., $\text{df} > 1$ ($\lambda \rightarrow \infty$) (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 $df = 1$ to the centered effect (and add as P-spline base-learner)

Technical realization [see Fahrmeir et al. 2004]:

Decomposition of the vector of regression coefficients β into $(\tilde{\beta}_{\text{unpen}}, \tilde{\beta}_{\text{pen}})$ utilizing a spectral decomposition of the penalty matrix.

Results

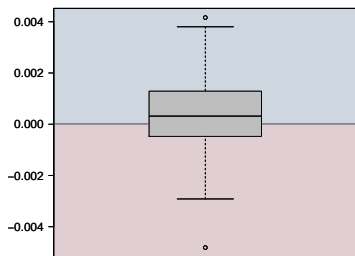
“Power Case” with (Potentially) Smooth Effects

- 5 continuous covariates with $\beta_{\text{info}} = (-2, -1, 1, 2, 3)^\top$
20 additional non-informative continuous covariates
- 1 **continuous** covariate with **linear effect** ($\beta_{z_1} = 1.5$)
- Otherwise same simulation setting as in “factor case”
- Add **(A)** linear effect + smooth effect (4 df)
or **(B)** linear effect + smooth deviation from linearity (1 df)

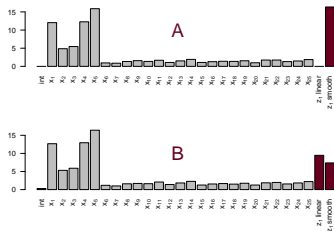
Results

“Power Case” with (Potentially) Smooth Effects

- 5 continuous covariates with $\beta_{\text{info}} = (-2, -1, 1, 2, 3)^\top$
20 additional non-informative continuous covariates
- 1 **continuous** covariate with **linear effect** ($\beta_{z_1} = 1.5$)
- Otherwise same simulation setting as in “factor case”
- Add **(A)** linear effect + smooth effect (4 df)
or **(B)** linear effect + smooth deviation from linearity (1 df)



Goodness of fit: partial deviation (A - B)



Selection Frequency

Summary

- Boosting (intrinsically) allows for **variable / model selection**.
- The selection may be seriously biased in naive specifications.
- We get a **severe reduction** of selection bias by using **penalized base-learners with equal df**.
- Use a **suitable definition** of degrees of freedom

$$\text{df} = \text{trace}(2\mathbf{S} - \mathbf{S}^\top \mathbf{S}).$$

Summary

- Boosting (intrinsically) allows for **variable / model selection**.
- The selection may be seriously biased in naive specifications.
- We get a **severe reduction** of selection bias by using **penalized base-learners with equal df**.
- Use a **suitable definition** of degrees of freedom

$$\text{df} = \text{trace}(2\mathbf{S} - \mathbf{S}^\top \mathbf{S}).$$

Not covered here

- Instead of the ridge penalty one can use a difference penalty for ordinal covariates
- Ordinal difference penalty superior to ridge penalty in case of ordinal covariates

Part III

Constrained Regression

based on Hofner et al. [2011b]

Introduction

- We want to include prior assumptions such as monotonicity assumptions, periodic behavior, etc. into the model.
- Include subject-matter knowledge into the model to achieve models that
 - are better to interpret
 - more stable

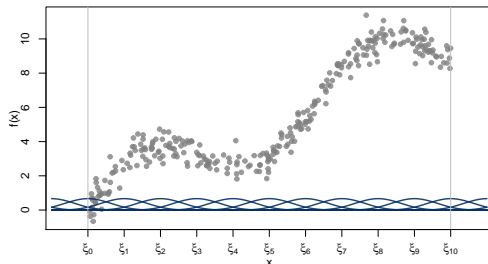
Short Review of P-splines

[Eilers and Marx 1996]

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

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

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



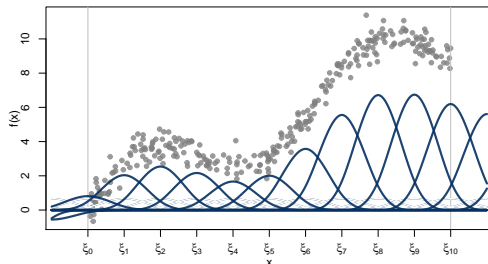
Short Review of P-splines

[Eilers and Marx 1996]

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

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

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



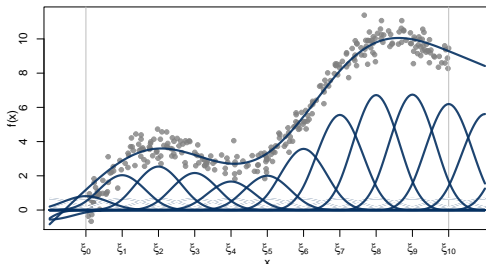
Short Review of P-splines

[Eilers and Marx 1996]

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

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

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**:

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

where d is the order of the difference penalty.

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

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

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

- **Difference matrix** $\mathbf{D}_{(d)}$

$$\mathbf{D}_{(1)} = \begin{pmatrix} -1 & 1 & 0 & \dots \\ 0 & -1 & 1 & \dots \\ 0 & 0 & \ddots & \ddots \end{pmatrix} \quad \mathbf{D}_{(2)} = \begin{pmatrix} 1 & -2 & 1 & 0 & \dots \\ 0 & 1 & -2 & 1 & \dots \\ 0 & 0 & \ddots & \ddots & \ddots \end{pmatrix}$$

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

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

where d is the order of the difference penalty.

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

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

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

- **Difference matrix** $\mathbf{D}_{(d)}$

$$\mathbf{D}_{(1)} = \begin{pmatrix} -1 & 1 & 0 & \dots \\ 0 & -1 & 1 & \dots \\ 0 & 0 & \ddots & \ddots \end{pmatrix} \quad \mathbf{D}_{(2)} = \begin{pmatrix} 1 & -2 & 1 & 0 & \dots \\ 0 & 1 & -2 & 1 & \dots \\ 0 & 0 & \ddots & \ddots & \ddots \end{pmatrix}$$

Estimation: Penalized least squares criterion

$$\mathcal{Q}(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{B}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{B}\boldsymbol{\beta}) + \lambda \mathcal{J}(\boldsymbol{\beta}; d)$$

Monotonic P-splines

- ✓ **Smoothness Constraint:**

Difference penalty on coefficients of adjacent B-splines

- **Monotonicity Constraint:**

$$f'(x) = \frac{\partial}{\partial x} \sum \beta_j B_j(x; l) = \frac{1}{h} \sum \Delta^1 \beta_{j+1} B_j(x; l - 1)$$

(distance of knots $h > 0$, B-spline basis $B_j(x; l - 1) \geq 0$)

- ▶ depends only on the **first differences of the adjacent coefficients**

Monotonic P-splines

- ✓ **Smoothness Constraint:**

Difference penalty on coefficients of adjacent B-splines

- **Monotonicity Constraint:**

$$f'(x) = \frac{\partial}{\partial x} \sum \beta_j B_j(x; l) = \frac{1}{h} \sum \Delta^1 \beta_{j+1} B_j(x; l - 1)$$

(distance of knots $h > 0$, B-spline basis $B_j(x; l - 1) \geq 0$)

▶ depends only on the **first differences of the adjacent coefficients**

- Monotonic increasing function:

$$f'(x) > 0 \quad \Leftrightarrow \quad \Delta^1 \beta_{j+1} > 0 \quad \forall j$$

Monotonic P-splines

✓ Smoothness Constraint:

Difference penalty on coefficients of adjacent B-splines

• Monotonicity Constraint:

$$f'(x) = \frac{\partial}{\partial x} \sum \beta_j B_j(x; l) = \frac{1}{h} \sum \Delta^1 \beta_{j+1} B_j(x; l - 1)$$

(distance of knots $h > 0$, B-spline basis $B_j(x; l - 1) \geq 0$)

▶ depends only on the **first differences of the adjacent coefficients**

- Monotonic increasing function:

$$f'(x) > 0 \quad \Leftrightarrow \quad \Delta^1 \beta_{j+1} > 0 \quad \forall j$$

- Monotonic decreasing function:

$$f'(x) < 0 \quad \Leftrightarrow \quad \Delta^1 \beta_{j+1} < 0 \quad \forall j$$

Monotonic P-splines

✓ Smoothness Constraint:

Difference penalty on coefficients of adjacent B-splines

• Monotonicity Constraint:

$$f'(x) = \frac{\partial}{\partial x} \sum \beta_j B_j(x; l) = \frac{1}{h} \sum \Delta^1 \beta_{j+1} B_j(x; l - 1)$$

(distance of knots $h > 0$, B-spline basis $B_j(x; l - 1) \geq 0$)

► depends only on the **first differences of the adjacent coefficients**

- Monotonic increasing function:

$$f'(x) > 0 \quad \Leftrightarrow \quad \Delta^1 \beta_{j+1} > 0 \quad \forall j$$

- Monotonic decreasing function:

$$f'(x) < 0 \quad \Leftrightarrow \quad \Delta^1 \beta_{j+1} < 0 \quad \forall j$$

• Convexity / Concavity Constraint:

$$f''(x) = \frac{1}{h^2} \sum \Delta^2 \beta_{j+2} B_j(x; l - 2)$$

Penalty

only needed for differences that violate the monotonicity assumption

- “Monotonic coefficients” can be achieved by (additional) **asymmetric difference penalties** on adjacent coefficients [Eilers 2005]:

$$\mathcal{J}_{\text{asym}}(\boldsymbol{\beta}; c) = \sum_{j=c+1}^J v_j (\Delta^c \beta_j)^2, = \boldsymbol{\beta}^\top \mathbf{D}_{(c)}^\top \mathbf{V} \mathbf{D}_{(c)} \boldsymbol{\beta}, \quad (2)$$

where c is the order of the difference penalty.

- “Monotonic coefficients” can be achieved by (additional) **asymmetric difference penalties** on adjacent coefficients [Eilers 2005]:

$$\mathcal{J}_{\text{asym}}(\boldsymbol{\beta}; c) = \sum_{j=c+1}^J v_j (\Delta^c \beta_j)^2, = \boldsymbol{\beta}^\top \mathbf{D}_{(c)}^\top \mathbf{V} \mathbf{D}_{(c)} \boldsymbol{\beta}, \quad (2)$$

where c is the order of the difference penalty.

- Important difference to P-spline penalty are **weights** v_j , which are specified as

$$v_j = \begin{cases} 0 & \text{if } \Delta^c \beta_j > 0 \\ 1 & \text{if } \Delta^c \beta_j \leq 0, \end{cases} \text{ monotonic increasing} \quad (3)$$

matrix notation: $\mathbf{V} = \text{diag}(\mathbf{v})$, and difference matrices as above.

- “Monotonic coefficients” can be achieved by (additional) **asymmetric difference penalties** on adjacent coefficients [Eilers 2005]:

$$\mathcal{J}_{\text{asym}}(\boldsymbol{\beta}; c) = \sum_{j=c+1}^J v_j (\Delta^c \beta_j)^2, = \boldsymbol{\beta}^\top \mathbf{D}_{(c)}^\top \mathbf{V} \mathbf{D}_{(c)} \boldsymbol{\beta}, \quad (2)$$

where c is the order of the difference penalty.

- Important difference to P-spline penalty are **weights** v_j , which are specified as

$$v_j = \begin{cases} 0 & \text{if } \Delta^c \beta_j < 0 \\ 1 & \text{if } \Delta^c \beta_j \geq 0, \end{cases} \text{ monotonic decreasing} \quad (3)$$

matrix notation: $\mathbf{V} = \text{diag}(\mathbf{v})$, and difference matrices as above.

- “Monotonic coefficients” can be achieved by (additional) **asymmetric difference penalties** on adjacent coefficients [Eilers 2005]:

$$\mathcal{J}_{\text{asym}}(\boldsymbol{\beta}; c) = \sum_{j=c+1}^J v_j (\Delta^c \beta_j)^2, = \boldsymbol{\beta}^\top \mathbf{D}_{(c)}^\top \mathbf{V} \mathbf{D}_{(c)} \boldsymbol{\beta}, \quad (2)$$

where c is the order of the difference penalty.

- Important difference to P-spline penalty are weights v_j , which are specified as

$$v_j = \begin{cases} 0 & \text{if } \Delta^c \beta_j < 0 \\ 1 & \text{if } \Delta^c \beta_j \geq 0, \end{cases} \text{ monotonic decreasing} \quad (3)$$

matrix notation: $\mathbf{V} = \text{diag}(\mathbf{v})$, and difference matrices as above.

- Monotonicity constraint if $c = 1$
- Convex / concave constraint if $c = 2$

Estimation of Constrained P-splines

Weights v_j depend on β . Thus:

- 1) Start with standard P-spline estimate
- 2) Compute weights v_j for $\hat{\beta}$
- 3) Minimize

$$Q(\beta) = (\mathbf{y} - \mathbf{B}\beta)^\top (\mathbf{y} - \mathbf{B}\beta) + \lambda_1 \mathcal{J}(\beta; d) + \lambda_2 \mathcal{J}_{\text{asym}}(\beta; c) \quad (4)$$

► **penalized least squares** estimate

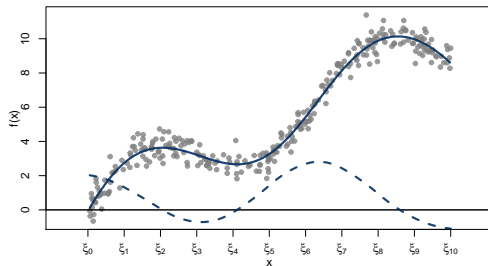
$$\hat{\beta} = (\mathbf{B}^\top \mathbf{B} + \lambda_1 \mathbf{D}_{(d)}^\top \mathbf{D}_{(d)} + \lambda_2 \mathbf{D}_{(c)}^\top \mathbf{V} \mathbf{D}_{(c)})^{-1} \mathbf{B}^\top \mathbf{y}$$

- 4) Recompute weights v_j with $\hat{\beta}$
- 5) Iterate 3) and 4) until no more changes in v_j (usually after 2-3 steps)

Smoothing parameter λ_2 is chosen quite large (e.g., 10^6), where larger values are associated with a stronger impact of the monotonic constraint

P-Splines, Monotonic Splines

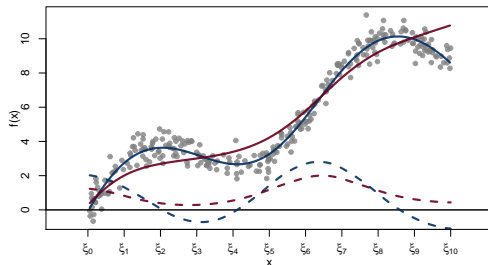
and their Derivatives



- P-spline estimate and derivative (dashed)

P-Splines, Monotonic Splines

and their Derivatives



- P-spline estimate and derivative (dashed)
- Monotonic spline estimate and derivative (dashed)

Incorporating Monotonic Effects into Boosting

- We use one base-learner per monotonic effect.
- The constrained estimation is done completely within the base-learner.
- ▶ There is no need to change the boosting algorithm itself.
- ▶ Monotonic effects can be freely combined with other (complex) types of effects in one model.

Incorporating Monotonic Effects into Boosting

- We use one base-learner per monotonic effect.
- The constrained estimation is done completely within the base-learner.
- ▶ There is no need to change the boosting algorithm itself.
- ▶ Monotonic effects can be freely combined with other (complex) types of effects in one model.

Alternatives (in the Boosting Context):

- Leitenstorfer and Tutz [2007] propose a (similar) approach based on groups of B-splines.
- Tutz and Leitenstorfer [2007] propose an approach based on monotonic basis functions (e.g., I-splines).

Results of Simulations (Example)

Model:

$$y = \alpha + f_1(x_1) + f_2(x_2) + \varepsilon \quad \text{with smooth, monotonic effects } f_1 \text{ and } f_2$$

PMSE (for 10,000 new observations) of monotonicity-constrained and unconstrained models estimated from 100 simulation runs.

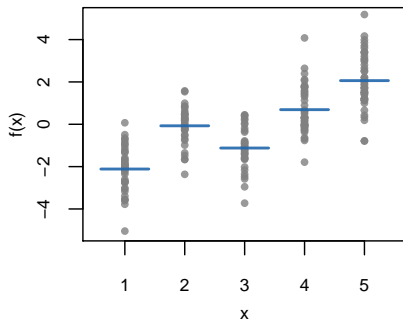
n	σ^2	monotonic	(se)	unconstrained	(se)
100	1.0	0.0759*	(0.0035)	0.0981 [†]	(0.0046)
	0.4	0.0349*	(0.0015)	0.0431 [†]	(0.0019)
	0.1	0.0122*	(0.0004)	0.0129 [†]	(0.0005)
200	1.0	0.0405*	(0.0020)	0.0449 [†]	(0.0022)
	0.4	0.0189*	(0.0008)	0.0198 [†]	(0.0008)
	0.1	0.0076*	(0.0002)	0.0063 [†]	(0.0002)
500	1.0	0.0155*	(0.0008)	0.0171 [†]	(0.0007)
	0.4	0.0083*	(0.0003)	0.0081[†]	(0.0003)
	0.1	0.0046*	(0.0001)	0.0032[†]	(0.0001)

* Monotonic estimates in **all** cases

[†] Non-monotonic estimates in most cases (threshold $\Delta\beta_j > -10^{-4}$)

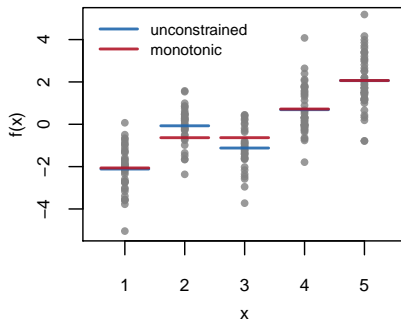
Further Constraints

- Monotonicity constraints can also be imposed on **ordinal, categorical variables** using the same ideas.



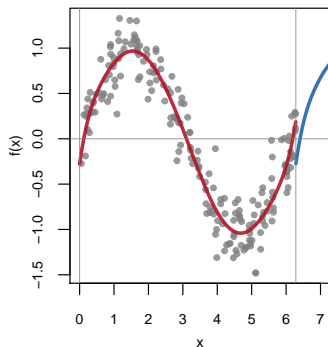
Further Constraints

- Monotonicity constraints can also be imposed on **ordinal, categorical variables** using the same ideas.



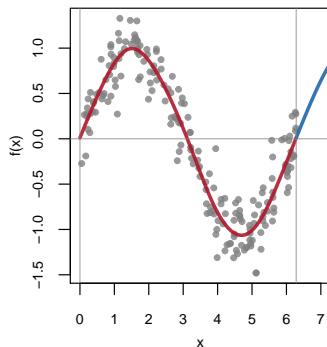
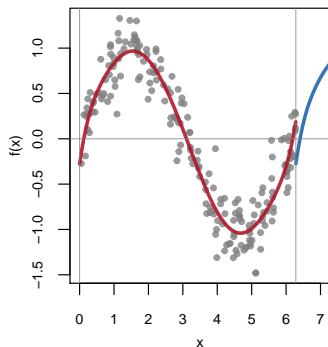
Further Constraints

- Monotonicity constraints can also be imposed on ordinal, categorical variables using the same ideas.
- **Cyclic constraints** interesting in the setting of time-series or longitudinal models [cf. Eilers and Marx 2010]



Further Constraints

- Monotonicity constraints can also be imposed on ordinal, categorical variables using the same ideas.
- **Cyclic constraints** interesting in the setting of time-series or longitudinal models [cf. Eilers and Marx 2010]



Further Constraints

- Monotonicity constraints can also be imposed on ordinal, categorical variables using the same ideas.
- Cyclic constraints interesting in the setting of time-series or longitudinal models [cf. Eilers and Marx 2010]

Simulations

- Settings similar to above
- For monotonic, ordinal effects and cyclic effects
 - ▶ constrained model outperforms unconstrained model in **all** settings

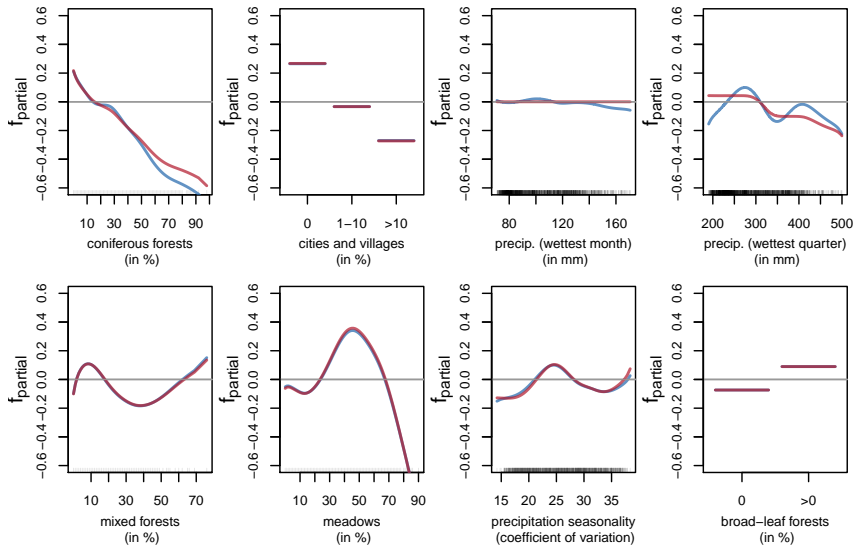
Application: Red Kite Breeding Distribution

Let us come back to have a look at the Red Kite breeding data.

Aims:

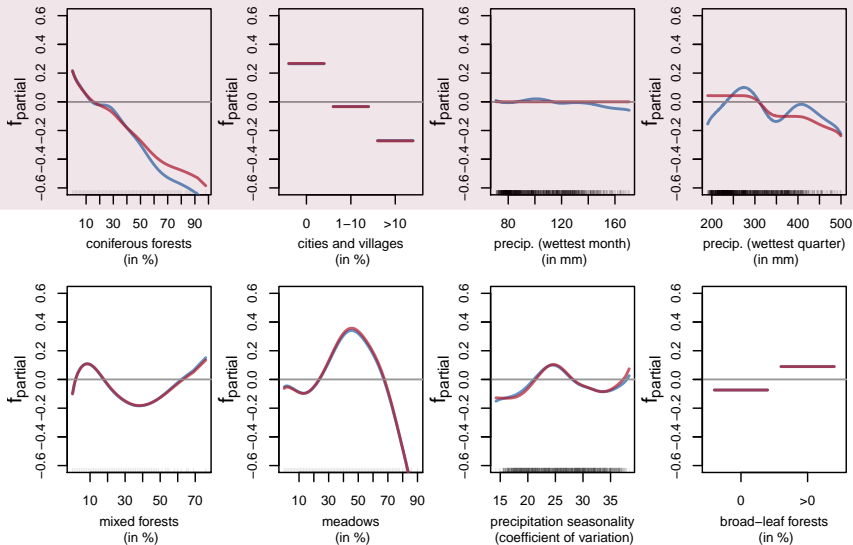
- Model the breeding probability of Red Kites in Bavaria
- Following subject-matter knowledge (► Jörg Müller) we assume monotonically decreasing effects for
 - coniferous forests (continuous; in %)
 - cities and villages (ordinal; in %)
 - precipitation wettest month (continuous; in mm)
 - precipitation wettest quarter (continuous; in mm)
- Include variable selection (as we have 55 base-learners in total)

Results (selection)

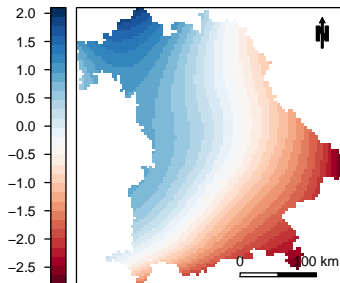


Results (selection)

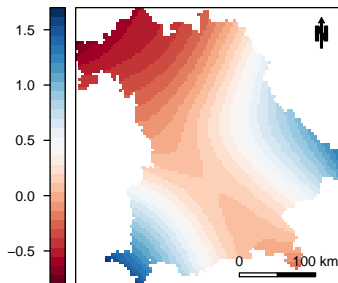
Monotonic



Results (selection)



Spatial Effect (1979–1983)



Change of Spatial Effect
(between 1979–1983 and 1996–1999)

Summary

- Sparse model (only 10 out of 55 base-learners selected)
- Monotonicity constraints lead to models that are better to interpret
- Complex effects can be estimated within the same model (e.g., spatial or spatio-temporal effects and other non-restricted smooth effects)
- Other effects hardly affected (here)

Remark

- Constrained effects can be extended to **bivariate** monotonic or **bivariate** cyclic effects

Part IV

Summary & Outlook

Summary

- A framework for unbiased boosting was proposed. To achieve unbiased selection
 - use appropriate degrees of freedom
 - make the degrees of freedom comparable for the base-learners
- Boosting allows to incorporate constrained effect estimates
 - No need to change the algorithm
 - Implementation using base-learners only
 - ▶ Can be combined with arbitrary loss functions and arbitrary other base-learners

R-package **mboost** available on **CRAN** to fit all the effects and models covered in this talk (and many more) [Hothorn et al. 2010; 2012]

Outlook

- Unbiased variable selection in likelihood-based boosting
To what extent is this possible?
- Partial monotonicity constraints (only on a certain subspace)
Use only a subset of the knots in the asymmetric difference penalty
- Use constraints on boundaries (e.g., constant boundaries for effects that are supposed to level out)
Use an additional penalty only for the boundaries with a pre-specified large penalty parameter
- Adapt cyclic effects to ordinal variables (e.g., day of the week)
Use a wrapped difference penalty

References I

- L Breiman, JH Friedman, RA Olshen, and CJ Stone. *Classification and Regression Trees*. Wadsworth, California, 1984.
- A Buja, T Hastie, and R Tibshirani. Linear smoothers and additive models (with discussion). *The Annals of Statistics*, 17:453–555, 1989.
- PHC Eilers. Unimodal smoothing. *Journal of Chemometrics*, 19:317–328, 2005.
- PHC Eilers and BD Marx. Flexible smoothing with B-splines and penalties (with discussion). *Statistical Science*, 11:89–121, 1996.
- PHC Eilers and BD Marx. Splines, knots, and penalties. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2:637–653, 2010.
- 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*, 2011a. doi: 10.1198/jcgs.2011.09220. to appear.
- B Hofner, J Müller, and T Hothorn. Monotonicity-constrained species distribution models. *Ecology*, 92:1895–1901, 2011b.
- T Hothorn, K Hornik, and A Zeileis. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15: 651–674, 2006.

References II

- T Hothorn, P Bühlmann, T Kneib, M Schmid, and B Hofner. Model-based boosting 2.0. *Journal of Machine Learning Research*, 11:2109–2113, 2010.
- T Hothorn, P Bühlmann, T Kneib, M Schmid, and B Hofner. *mboost: Model-Based Boosting*, 2012. URL <http://CRAN.R-project.org/package=mboost>. R package version 2.1-2.
- T Kneib, T Hothorn, and G Tutz. Variable selection and model choice in geospatial regression models. *Biometrics*, 65:626–634, 2009.
- F Leitenstorfer and G Tutz. Generalized monotonic regression based on B-splines with an application to air pollution data. *Biostatistics*, 8:654–673, 2007.
- WY Loh. Regression trees with unbiased variable selection and interaction detection. *Statistica Sinica*, 12:361–386, 2002.
- C Strobl, AL Boulesteix, A Zeileis, and T Hothorn. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, 8:25, 2007.
- G Tutz and F Leitenstorfer. Generalized smooth monotonic regression in additive modeling. *Journal of Computational and Graphical Statistics*, 16:165–188, 2007.