David Martin Rügamer

Estimation, Model Choice and Subsequent Inference: Methods for Additive and Functional Regression Models

Dissertation an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

Eingereicht am 08. Mai 2018

David Martin Rügamer

Estimation, Model Choice and Subsequent Inference: Methods for Additive and Functional Regression Models

Dissertation an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

Eingereicht am 08. Mai 2018

Erste Berichterstatterin: Prof. Dr. Sonja Greven
Zweiter Berichterstatter: Prof. Dr. Helmut Küchenhoff
Dritter Berichterstatter: Prof. Dr. Matthias Schmid

Tag der Disputation: 15. Juni 2018

Acknowledgments

First and foremost, I would like to thank my supervisor Prof. Dr. Sonja Greven. I am deeply grateful for all the help, the fruitful discussions, the proofreading and, in particular, for supporting all of my research. The supervision and support I received was exceptional and can by no means be taken for granted.

I would also like to thank Prof. Dr. Helmut Küchenhoff for serving as a second reviewer and for giving me the chance to work at the StaBLab, thereby getting to know the art of applied statistics as well as understanding the importance of some of the concepts addressed in this thesis.

My thanks also go to

... Prof. Dr. Matthias Schmid, who kindly agreed to serve as a reviewer and to Prof. Dr. Christian Heumann and Prof. Dr. Anne-Laure Boulesteix for complementing the doctoral committee.

... Prof. Dr. Klaus Scherer and Dr. Kornelia Gentsch for working together on this very interesting project, for taking the time to proofread the manuscript and for sharing their knowledge with me.

... Prof. Dr. Thomas Kneib and Dr. Benjamin Säfken for a fruitful and easy collaboration.

... all my (former) colleagues and all the other collaborators. Especially Fabian for initializing my first research project, for all the useful comments and the proofreading.

A big big thanks to Sarah for all your help, your trust, for sharing all your experience and clever ideas, and for letting me be part of the software package.

Aside from all the support, which led to this thesis ...

... I want to thank my former fellow student and good friend David. I definitely owe you much of my interest and early success in statistics.

... I want to express my deep gratitude to Almond und Meike, who were always interested in all the statistical and non-statistical things I wanted to share in- and outside of the office. Thanks also to Henry. Spending time with the three of you was a great pleasure.

... I want to thank all of my other friends and, in particular, my good longtime friends and colleagues Andreas and Schorschi for both, living the good life, and being my company on busy weekends in the department.

... I want to say thanks to my family, who supported me from the very first beginning in so much different ways, to Gabi and Chris, and especially to Natalie for all the help and mental support. This would not be possible without you. *Kamsahamnida.*

Summary

The thesis addresses model estimation, model choice and subsequent inference for regression coefficients in additive and functional regression models. The presented methods describe a framework for statistical modelling in practical relevant model classes, including efficient estimation of complex function-on-function regression models, commonly used model selection criteria in linear, mixed and additive models as well as valid inference procedures following model selection.

The first part of this thesis focuses on model selection and valid inference after model selection. After introducing the Akaike Information Criterion (AIC) as a commonly used model selection criterion, the conditional AIC (cAIC) as one possible extension of the AIC to the class of mixed and additive models is presented. In this context, the R package `cAIC4`, which provides an efficient implementation of the cAIC, is explained in detail. Due to invalidity of classical statistical inference after model selection, analytical expressions for inference after likelihood- or test-based model selection including AIC-based model selection are derived for linear models. Afterwards, this inference framework is also extended to models obtained after the selection process induced by L_2-boosting.

The second part of this thesis is concerned with model estimation, model choice and uncertainty quantification in function-on-function regression models. Motivated by research questions in the field of cognitive affective neuroscience, function-on-function regression models are extended to models including random historical effects, factor-specific historical effects, and factor-specific random historical effects. The estimation and model selection is conducted by a component-wise gradient boosting algorithm, which is implemented in the R add-on package `FDboost`. An introduction into the implementation in R concludes the first part of this thesis.

Zusammenfassung

Die vorliegende Arbeit beschäftigt sich mit Modellschätzung, Modellwahl sowie anschließender Inferenz für Regressionskoeffizienten in additiven und funktionalen Regressionsmodellen. Die vorgestellten Methoden beschreiben Ansätze für die statistische Modellierung in praxisrelevanten Modellklassen, einschließlich der effizienten Schätzung komplexer funktionaler Regressionsmodelle, Beschreibung und Implementierung häufig verwendeter Modellwahlkriterien in linearen, gemischten und additiven Modellen sowie Ansätze für gültige Inferenzverfahren nach Modellselektion.

Der Schwerpunkt des ersten Teils dieser Arbeit liegt auf der Modellwahl sowie gültigen Inferenz nach Modellselektion. Ein häufig in der Praxis eingesetztes Modellwahlkriterium stellt das Akaike Informationskriteriums (AIC) dar, das zunächst eingeführt wird. Auf dieser Basis wird das konditionale AIC (cAIC) als eine mögliche Erweiterung der AICs auf die Klasse der gemischten und additiven Modelle vorgestellt. In diesem Zusammenhang wird das R-Paket `cAIC4`, das eine effiziente Implementierung des cAIC bereitstellt, näher erläutert. Aufgrund der Ungültigkeit klassischer statistischer Inferenz nach Modellselektion werden analytische Ausdrücke für die Inferenz nach Likelihood- oder testbasierter Modellwahl einschließlich der AIC-basierten Modellauswahl für lineare Modelle hergeleitet. Anschließend wird das vorgestellte Inferenzkonzept auf Modelle erweitert, die mithilfe des L_2 -Boosting Algorithmus selektiert wurden.

Der zweite Teil dieser Arbeit beschäftigt sich mit Modellschätzung, Modellwahl und Unsicherheitsquantifizierung in funktionalen Regressionsmodellen. Motiviert durch Forschungsfragen auf dem Gebiet der kognitiven affektiven Neurowissenschaft werden Funktions-auf-Funktions-Regressionsmodelle auf Modelle mit zufälligen historischen Effekten, faktorspezifischen historischen Effekten und faktorspezifischen zufälligen historischen Effekten erweitert. Die Schätzung und Modellauswahl erfolgt mithilfe eines komponentenweisen Gradientenabstiegsverfahren im Funktionsraum, welches im R Paket `FDboost` implementiert ist. Eine Einführung in die Implementierung schließt den zweiten Teil dieser Arbeit ab.

Contents

Chapter 1

Introduction

1.1 Overview

As it is the case for many research questions in the field of statistics nowadays, this thesis is concerned with challenges stemming from the ever larger and more complex data collections, which are available due to the continuously progressing digitalization and new technological innovations. Whereas some of these challenges can be solved without new statistical methodology, others motivate refinements and further development of existing methods, in particular, to adapt for the unprecedented nature of data collections and the increasingly data-driven use of statistics in practice. This thesis is concerned with additive and functional regression models and addresses the three topics *estimation*, *model choice* and *subsequent inference* in the light of complex data structures and the data-driven use of statistics. Whereas both parts of the thesis deal with all three topics, the first part of this thesis focuses on methods for additive regression models and is mainly concerned with model choice as well as valid inference after model selection. The second part of this thesis describes the estimation of functional regression models as well as challenges accompanied with complex data collections and hypotheses.

The following sections constitute a methodological preface for the contributing articles in this thesis, introducing different regression models, briefly summarizing the idea and development of statistical boosting and giving additional background information on post-selection inference. In all of these sections, an overview of the existing literature as well as scientific context is given and it is described, how the reprinted articles can be embedded in this context.

1.2 Generalized Additive and Mixed Models

In the following, an introduction into different regression models that are used in the contributing articles is given with special focus on additive models (AMs) as well as mixed models (MMs). Part I of this thesis is concerned with model choice and subsequent inference in additive and mixed models. A brief introduction in model selection is given in the Subsections 1.2.5 and 1.3. The following

subsection describes basic concepts of representation and estimation of AMs and MMs. Although strongly related, functional regression models (FRMs) are described separately in Section 1.5.

The general regression setup is first described and extended to different model classes. Practical examples as well as further details can, e.g., be found in Ruppert et al. (2003), Fahrmeir et al. (2013) and Wood (2017).

1.2.1 Regression Model Setup

In the following regression models, the *response* or *dependent variable* is denoted by $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^\top \in \mathbb{R}^n$ with realizations $\boldsymbol{y} = (y_1, \ldots, y_n)^\top$ and p fixed *covariates* or *independent variables* are denoted by $\boldsymbol{x}_j = (x_{1,j}, \ldots, x_{n,j})^\top, j = 1, \ldots, p$, usually summarized in a design matrix $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p)$ with rows denoted by $\boldsymbol{X}_i, i = 1, \ldots, n$. The ulterior motive is to build a model for the conditional response distribution $\boldsymbol{Y}|\boldsymbol{X} \sim \mathcal{F}$. To this end, the expectation of $\mathcal{F}$ is assumed to have some structural or parametric form, which can be modeled on the basis of $\boldsymbol{X}$. In a *classical linear regression*, one primary goal is to estimate the true expectation $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)^\top$ of $\boldsymbol{Y}$ using the assumption $\boldsymbol{\mu} = \boldsymbol{X}\boldsymbol{\beta}$ with full column rank matrix $\boldsymbol{X}$. When observing an erroneous version of $\boldsymbol{Y}$, the *classical normal regression* is given by

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}_n(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_n), \tag{1.1}$$

where $\boldsymbol{I}_n$ is the n-dimensional identity matrix, $\sigma^2 > 0$ is an error variance of the independent and identically distributed (i.i.d.) errors $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)^\top$. When minimizing the expected squared error $\mathbb{E}_{\mathcal{F}}||\boldsymbol{\varepsilon}||^2$ with quadratic L_2-norm $||\cdot||^2$, the *regression coefficients* $\boldsymbol{\beta}$ can be derived by

$$\boldsymbol{\beta} = \underset{\boldsymbol{\gamma} \in \mathbb{R}^p}{\arg\min}\, \mathbb{E}_{\mathcal{F}}||\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\gamma}||^2 \overset{(1.1)}{=} (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{\mu}$$

and the least-squares (LS) estimator $\hat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$ is given by $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{Y} =: \boldsymbol{\eta}^\top \boldsymbol{Y}$ with *pseudo-inverse* $\boldsymbol{\eta} \in \mathbb{R}^{n \times p}$ of $\boldsymbol{X}$. This also corresponds to the solution of a maximum-likelihood (ML) estimation, i.e., finding the maximizer of the (log-)likelihood under the normality assumption (1.1).

For "first-order wrong models", $\boldsymbol{\mu}$ does not coincide with $\boldsymbol{X}\boldsymbol{\beta}$ (see, e.g., Berk et al., 2013) and $\boldsymbol{\beta} = \boldsymbol{\eta}^\top \boldsymbol{\mu}$ can be considered as the coefficients of the best linear approximation $\boldsymbol{X}\boldsymbol{\eta}^\top \boldsymbol{\mu}$ of $\boldsymbol{\mu}$ with design matrix $\boldsymbol{X}$. Following the agenda of Berk et al. (2013) and other recent publications, this linear approximation is the target of inference as first-order correctness of the linear model is usually not realistic in practice. Section 1.4 will describe this in more detail in the context of post-selection inference.

In all of the contributing articles, variable or model selection is utilized to obtain a better interpretable and potentially more predictive model or to simply facilitate the estimation of some form of best linear approximation of the true expectation when $\boldsymbol{X}$ is not of full-rank. Model or variable

selection methods considered in this thesis potentially determine the structural form of the model assumption of $\boldsymbol{\mu}$ as well as select a set of covariates, which are used to model $\boldsymbol{\mu}$. Given the linear regression model assumption and selected columns $\mathcal{A} \subset \{1, \ldots, p\}$ of $\boldsymbol{X}$ determined by some variable selection method, the target of inference changes to $\boldsymbol{\beta}_{\mathcal{A}} = \boldsymbol{\eta}_{\mathcal{A}}^{\top}\boldsymbol{\mu}$. It is noteworthy that in the case of variable selection within a class of models, e.g., linear models, some conventions with respect to the meaning of the *full* and *submodel* (for example, with structural assumption $\boldsymbol{\mu} = \boldsymbol{X}\boldsymbol{\beta}$ and $\boldsymbol{\mu} = \boldsymbol{X}_{\mathcal{A}}\boldsymbol{\beta}_{\mathcal{A}}$, respectively) are useful in order to clearly define the role and meaning of corresponding parameters. Following Berk et al. (2013), for methods involving a variable selection by subsetting the full design matrix in linear models, the model associated with $\boldsymbol{X}$ (the full model) has no special status, such as "the true underlying model", and $\boldsymbol{X}$ itself constitutes only a repository of available predictors. The coefficients of covariates, which are not selected by the variable selection procedure are not zero but are simply not defined. Furthermore, coefficients have different interpretation across different selected subsets $\mathcal{A}$, as they represent coefficients of best linear approximations based on different covariate sets $\boldsymbol{X}_{\mathcal{A}}$.

1.2.2 Linear Mixed Models

The linear model can be extended when incorporating so-called *random effects* $\boldsymbol{b} \in \mathbb{R}^r$, which, in contrast to the fixed regression coefficients $\boldsymbol{\beta}$, are assumed to be random variables following a mean zero normal distribution. Random effects can be motivated from various angles. In statistical applications, incorporating random effects into a regression model is done to appropriately model a given correlation structure, often induced by dependent observations. Random effects can also be seen as a regularization technique, which allows to "borrow strength" of more similar observations to improve or facilitate model estimation.

Let $\boldsymbol{Z} \in \mathbb{R}^{n \times r}$ be a design matrix for the random effects and let $\boldsymbol{G}$ and $\boldsymbol{R}$ be two positive semi-definite covariance matrices. The linear mixed model is given by the structural assumption

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{b} + \boldsymbol{\varepsilon}$$

and the distributional assumption

$$\boldsymbol{b} \sim \mathcal{N}_r(\boldsymbol{0}, \boldsymbol{G}), \boldsymbol{\varepsilon} \sim \mathcal{N}_n(\boldsymbol{0}, \boldsymbol{R}), \boldsymbol{b} \perp \boldsymbol{\varepsilon}.$$

For the linear mixed model two different perspectives exist, which stem from the conditional distribution $\boldsymbol{Y}|\boldsymbol{b} \sim \mathcal{N}_n(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{b}, \boldsymbol{R})$ and the marginal distribution $\boldsymbol{Y} \sim \mathcal{N}_n(\boldsymbol{X}\boldsymbol{\beta}, \boldsymbol{V} := \boldsymbol{Z}\boldsymbol{G}\boldsymbol{Z}^{\top} + \boldsymbol{R})$ of the response, where the marginal formulation can be derived from the conditional distribution, but not vice versa (see, e.g., Fahrmeir et al., 2013).

Moreover, both the marginal and the conditional distribution assumption are used in the estimation of linear mixed models. For known covariance matrices $\boldsymbol{G}$ and $\boldsymbol{R}$ and existence of $\boldsymbol{V}^{-1}$, the

fixed effects vector can be estimated using the marginal formulation by minimizing the weighted or generalized least-squares (GLS) criterion

$$(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top \boldsymbol{V}^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \to \min_{\boldsymbol{\beta}},$$

which is equivalent to maximizing the likelihood $L(\boldsymbol{\beta})$ given by the marginal distribution assumption of $\boldsymbol{Y}$. For the prediction of random effects, the joint distribution

$$\begin{pmatrix} \boldsymbol{Y} \\ \boldsymbol{b} \end{pmatrix} \sim \mathcal{N}_{n+r}\left(\begin{pmatrix} \boldsymbol{X}\boldsymbol{\beta} \\ \boldsymbol{0} \end{pmatrix}, \begin{pmatrix} \boldsymbol{V} & \boldsymbol{Z}\boldsymbol{G} \\ \boldsymbol{G}\boldsymbol{Z}^\top & \boldsymbol{G} \end{pmatrix} \right)$$

can be used. The corresponding log-likelihood $\ell(\boldsymbol{Y}, \boldsymbol{b})$ can be considered as penalized log-likelihood

$$\ell(\boldsymbol{Y}, \boldsymbol{b}) = -\frac{1}{2}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Z}\boldsymbol{b})^\top \boldsymbol{R}^{-1}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Z}\boldsymbol{b}) - \frac{1}{2}\boldsymbol{b}^\top \boldsymbol{G}^{-1}\boldsymbol{b}$$

with penalty term $\boldsymbol{b}^\top \boldsymbol{G}^{-1}\boldsymbol{b}$ and its maximization is equivalent to the minimization of the so-called penalized least-squares (PLS) criterion

$$(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Z}\boldsymbol{b})^\top \boldsymbol{R}^{-1}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Z}\boldsymbol{b}) + \boldsymbol{b}^\top \boldsymbol{G}^{-1}\boldsymbol{b}.$$

This, in turn, is an extension of the GLS criterion as it additionally incorporates a penalization for deviations of $\boldsymbol{b}$ from the zero mean assumption $\mathbb{E}(\boldsymbol{b}) = \boldsymbol{0}$. The resulting estimator is given by

$$\begin{pmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\boldsymbol{b}} \end{pmatrix} = (\boldsymbol{C}^\top \boldsymbol{R}^{-1}\boldsymbol{C} + \boldsymbol{A})^{-1}\boldsymbol{C}^\top \boldsymbol{R}^{-1}\boldsymbol{Y}$$

with composed matrix $\boldsymbol{C} = (\boldsymbol{X}, \boldsymbol{Z})$ and block-diagonal matrix $\boldsymbol{A} = \text{blockdiag}(\boldsymbol{0}_{p\times p}, \boldsymbol{G}^{-1})$.

When $\boldsymbol{G}$ or $\boldsymbol{R}$ involve unknown parameters, denoted as vector $\boldsymbol{\tau} \in \mathbb{R}^\iota$, an estimator $\hat{\boldsymbol{\tau}}_{ML}$ for unknown parameters is given as maximizer of the profile-log-likelihood

$$\ell_P(\boldsymbol{\tau}) = -\frac{1}{2}\left\{\log|\boldsymbol{V}(\boldsymbol{\tau})| + (\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}(\boldsymbol{\tau}))^\top \boldsymbol{V}(\boldsymbol{\tau})^{-1}(\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}(\boldsymbol{\tau}))\right\}.$$

This estimator, however, tends to underestimate variance components (see, e.g., Fahrmeir et al., 2013). An alternative estimator is the restricted maximum-likelihood (REML) estimator $\hat{\boldsymbol{\tau}}_{REML}$, which is less biased downwards towards zero. By integrating out $\boldsymbol{\beta}$ in the joint log-likelihood $\log \int L(\boldsymbol{\beta}, \boldsymbol{\tau}) d\boldsymbol{\beta} =: \ell_R(\boldsymbol{\tau})$ and maximizing the marginal or *restricted* log-likelihood ℓ_R with respect to $\boldsymbol{\tau}$, which can be done with a Newton-Raphson(-type) algorithm in practice, the estimator $\hat{\boldsymbol{\tau}}_{REML}$ is obtained.

More details on linear mixed models, their estimation and extensions can be found in Fahrmeir et al. (2013, Section 7), Wood (2017, Section 2 and 3.4).

1.2.3 Additive Models

Another extension of the class of linear models is given by the class of additive models, which extends the linear model by incorporating non-parametric additive terms in the linear predictor:

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \sum_{j=1}^{J} f_j(\boldsymbol{z_j}) + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}_n(\boldsymbol{0}, \boldsymbol{R}) \tag{1.2}$$

with residual covariance $\boldsymbol{R}$ as defined in the previous subsection. The smooth functions $f_j(\boldsymbol{z}_j)$ of observed covariates $\boldsymbol{z}_j \in \mathbb{R}^n, j = 1, \ldots, J$, are infinite dimensional smooth effects, which can be approximated by a finite number of basis functions. This allows to embedded the representation and estimation of additive models in the class of linear models. As this basis representation is used in most of the contributing articles, the idea of basis function approaches is briefly sketched in the following.

Basis Representation

The fundamental principle of the basis representation is to approximate an infinite-dimensional function $f \in \mathcal{L}^2(\mathcal{T})$ from the squared-integrable space of functions on a domain $\mathcal{T}$ by the linear combination of a finite number of basis functions $B_1, \ldots, B_K \in \mathcal{L}^2(\mathcal{T})$ and coefficients $\vartheta_1, \ldots, \vartheta_K \in \mathbb{R}$: $f \approx \sum_{k=1}^{K} B_k \vartheta_k$. For a given covariate $\boldsymbol{z} = (z_1, \ldots, z_n)^\top$, for which a smooth effect f is assumed, the basis functions are evaluated at the observed values z_i, yielding a $(n \times K)$-matrix $\boldsymbol{B} = [B_k(z_i)]_{k=1,\ldots,K, i=1,\ldots,n}$, where the evaluation of the kth basis function at the ith observation z_i corresponds to the kth column and ith row of $\boldsymbol{B}$. If $K \leq n$, an L_2-loss-optimal representation of f for the given data points and specified basis functions can then be found by estimating the coefficient vector $\boldsymbol{\vartheta} = (\vartheta_1, \ldots, \vartheta_K)^\top$ via the least squares criterion:

$$(f(\boldsymbol{z}) - \boldsymbol{B}\boldsymbol{\vartheta})^\top (f(\boldsymbol{z}) - \boldsymbol{B}\boldsymbol{\vartheta}) \rightarrow \min_{\boldsymbol{\vartheta}}. \tag{1.3}$$

Under the assumption of a negligible approximation error when using only a finite number of basis functions, additive terms f_j in (1.2) can be estimated using the LS criterion. This can be done by first computing the evaluated basis functions $\boldsymbol{B}_j$ and then estimating regression coefficients $\boldsymbol{\beta}$ of linear terms together with basis coefficients $\boldsymbol{\vartheta}_j$ of functions f_j using a composed design matrix $(\boldsymbol{X}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_J)$ in a corresponding linear model (see, e.g., Ruppert et al., 2003; Fahrmeir et al., 2013).

In this thesis and in many statistical applications, a commonly used basis representation is the *B-spline basis*, introduced by Schoenberg (1946a) and Schoenberg (1946b). Given a partition of the covariate domain $\mathcal{T}$ by so-called *knots* $\kappa_1, \ldots, \kappa_d$, the function $f(\boldsymbol{z})$ is represented by $K = d + q - 1$ B-spline basis functions *of degree* q, which are $q + 1$ piecewise, continuously differentiable connected polynomial functions of degree q. For observations $z_i, i = 1, \ldots, n$, the kth B-spline of degree 0 is

calculated as an indicator function based on adjacent knots: $B_k^0(z_i) = I(\kappa_k \leq z_i \leq \kappa_{k+1})$. Similarly, B-splines of higher degree can be defined recursively

$$B_k^q(z_i) = \frac{z_i - \kappa_{k-1}}{\kappa_k - \kappa_{k-1}} B_{k-1}^{q-1}(z_i) + \frac{\kappa_{k+1} - z_i}{\kappa_{k+1} - \kappa_{k+1-q}} B_k^{q-1}(z_i).$$

For sufficiently high degree q, resulting splines are continuous and differentiable functions, which can be evaluated efficiently (de Boor, 1972), provide directly accessible (higher-order) derivatives and yield other numerically and mathematically desirable properties (see, e.g., Eilers and Marx, 1996; de Boor, 2001).

Whereas the location of knots can, e.g., be defined on an equidistant grid or on the basis of quantiles of the observed covariate $\boldsymbol{z}$, the smoothness of the function also depends on d, the number of knots, and different choices can have a crucial influence on the quality of $\hat{f}(\boldsymbol{z}) = \boldsymbol{B}\hat{\boldsymbol{\vartheta}}$. Eilers and Marx (1996) proposed a penalized version of B-splines, known as *P-splines*, which exhibits useful properties and, in particular, circumvents the problem of having to define an appropriate number d. By estimating coefficients for a generous number of B-spline basis functions with an appropriate penalty, P-splines allow for a flexible definition of f while preventing too rough estimates through a penalty term. As for the estimation of regression coefficients in linear mixed models, the least squares criterion in (1.3) is therefore extended by a quadratic penalty

$$(f(\boldsymbol{z}) - \boldsymbol{B}\boldsymbol{\vartheta})^\top (f(\boldsymbol{z}) - \boldsymbol{B}\boldsymbol{\vartheta}) + \lambda \boldsymbol{\vartheta}^\top \boldsymbol{P} \boldsymbol{\vartheta} \rightarrow \min_{\boldsymbol{\vartheta}},$$

with $\boldsymbol{P} \in \mathbb{R}^{K \times K}$ penalty matrix and λ a smoothing parameter controlling the influence of the penalty and thus the smoothness of the resulting function estimator $\hat{f}$. Eilers and Marx (1996) propose a quadratic penalization of coefficients of adjacent B-splines by defining $\boldsymbol{P}$ such that $\boldsymbol{\vartheta}^\top \boldsymbol{P} \boldsymbol{\vartheta} = \sum_{k=r+1}^{K} (\Delta^r \vartheta_k)^2$ with recursively defined rth-order differences Δ^r of coefficients $\vartheta_k, k = 1, \ldots, K$ and $r \in \mathbb{N}$ (e.g., $\Delta^1 \vartheta_k = \vartheta_k - \vartheta_{k-1}$; $\Delta^2 \vartheta_k = \Delta^1(\Delta^1 \vartheta_k)$). In practice, an optimal λ can be found via different approaches, including (generalized) cross-validation, using the Akaike Information Criterion or by utilizing the connection of the penalized least squares criterion and REML estimation in linear mixed models. Further details can, e.g., be found in Wood (2017).

Another way to estimate generalized additive (mixed) models is given by boosting, which from a statistical point of view can also be seen as regularization technique. In the Chapters 4 to 6 a special boosting routine is used for model estimation and regularization. A short introduction to statistical boosting based on its historical development is therefore given in Section 1.3.

1.2.4 Beyond Normality

Linear (mixed) and additive models can be extended to generalized linear (mixed) and additive models (GL(M)Ms and GAMs) for a vector $\boldsymbol{Y}$ of independent response variables $Y_i, i = 1, \ldots, n$, which follow

a distribution $\mathcal{F}_i$ from the exponential family conditional on the observed covariates $\boldsymbol{X}_i$ (McCullagh, 1984; Wood, 2017). The structural assumption in GL(M)Ms and GAMs is

$$\mu_i = \mathbb{E}(Y_i|\boldsymbol{X}_i) = g^{-1}(\zeta_i),$$

where g is a link function determining the relationship of the conditional expectation of Y_i and the linear predictor ζ_i, which includes additive terms for linear, random and/or additive effects. When parameterizing the conditional distribution of Y_i in terms of μ_i, the Likelihood $L(\boldsymbol{\beta})$ can be set up with respect to regression coefficients by, e.g., defining $\mu_i = g^{-1}(\boldsymbol{X}_i\boldsymbol{\beta})$ and (asymptotic) theory on likelihood-based estimation and inference becomes available.

For generalized models a closed form solution for estimators of regression coefficients does not exist in general. Estimation of regression coefficients in GL(M)Ms and GAMs is therefore usually done using iterative optimization techniques such as Fisher-Scoring for GLMs or the penalized iterative reweighted least squares algorithm for GLMMs (see, e.g., Wood, 2017). For more complex models or in high-dimensional settings, where the number of columns p in the design matrix $\boldsymbol{X}$ exceeds n, the component-wise functional gradient descent (CFGD) algorithm provides a possible alternative (see, e.g., Fahrmeir et al., 2013; Mayr et al., 2017). Also referred to as *component-wise boosting*, this algorithm iteratively adjusts the model fit based on (a function proportional to) the negative log-likelihood until a pre-specified criterion for convergence is met. A more detailed procedure of the CFGD algorithm will be given in Subsection 1.3, embedded in the historical development of boosting algorithms for statistical regression analysis.

As mentioned in the previous subsections, practical applications of regression models may require some sort of variable or model selection. In the contributing articles, two of the primary tools to achieve this are the CFGD algorithm and the *Akaike Information Criterion* (AIC; Akaike, 1973). The basic idea of the AIC is introduced in the following subsection for the class of GLMs. Extensions of the AIC to other model classes are described in the contributing article in Chapter 2 and, e.g., in Wood (2017, Chapter 6.11) for GAMs.

1.2.5 Akaike Information Criterion

The theoretical derivation of the AIC is based on the *Kullback-Leibler distance* (KLD; Kullback and Leibler, 1951), which is a measure of distance between two distributions. Let $\mathfrak{F}_{\boldsymbol{\theta}} = \{f(\boldsymbol{Y}|\boldsymbol{\theta}), \boldsymbol{\theta} \in \Theta\}$ be a family of distributions defined by the corresponding set of parametric density functions $f(\cdot|\boldsymbol{\theta}) =: f_{\boldsymbol{\theta}}$, where the parameter space $\Theta = \mathbb{R}^p$, except for a change of coordinates. This family of distributions results from an assumed statistical model such as (1.1), which can be seen as an approximation of the true but unknown distribution $\mathcal{G}$ with density g. To measure the goodness of fit of $\mathfrak{F}_{\boldsymbol{\theta}}$ the KLD $D(f_{\boldsymbol{\theta}}, g)$ can be used

$$D(f_{\boldsymbol{\theta}}, g) = \mathbb{E}_{\mathcal{G}_Y}\left[\log \frac{g(\boldsymbol{Y})}{f_{\boldsymbol{\theta}}(\boldsymbol{Y})}\right] = \int \log \frac{g(\boldsymbol{Y})}{f(\boldsymbol{Y}|\boldsymbol{\theta})}\, g(\boldsymbol{Y})\, \mathrm{d}\boldsymbol{Y}, \tag{1.4}$$

where $\mathbb{E}_{\mathcal{G}_Y}$ is the expectation with respect to the true distribution of $\boldsymbol{Y}$. A smaller distance $D(f_{\boldsymbol{\theta}}, g)$ then corresponds to a better model fit with special case $D(f_{\boldsymbol{\theta}}, g) = 0 \Leftrightarrow f_{\boldsymbol{\theta}} = g$. As minimization of (1.4) is equal to the minimization of $-2\mathbb{E}_{\mathcal{G}_Y}\left[\log f(\boldsymbol{Y}|\boldsymbol{\theta})\right]$ for a fixed true density function g, the *Akaike Information* (see, e.g., Greven and Kneib, 2010)

$$\mathrm{AI} = -2\,\mathbb{E}_{\mathcal{G}_Y}\left\{\mathbb{E}_{\mathcal{G}_{\tilde{\boldsymbol{Y}}}}\left[\log f(\tilde{\boldsymbol{Y}}|\hat{\boldsymbol{\theta}}(\boldsymbol{Y}))\right]\right\} \tag{1.5}$$

serves as a measure of expected quality of f approximating g for an independent sample $\tilde{\boldsymbol{Y}} \sim \mathcal{G}$ when $\boldsymbol{\theta}$ is estimated by $\hat{\boldsymbol{\theta}} := \hat{\boldsymbol{\theta}}(\boldsymbol{Y})$ using $\boldsymbol{Y}$. From Akaike's perspective, the problem of model selection can be regarded as choosing a model $f_{\hat{\boldsymbol{\theta}}_k}$ approximating the true model g as good as possible by minimizing (1.5) with respect to $\boldsymbol{\theta}_k$, where $\boldsymbol{\theta}_k$ lies in the k-parametric subspace $\Theta_k \subset \Theta_p$ given by the restriction $\theta_{k+1} = \theta_{k+2} = \ldots = \theta_p = 0$ for $\Theta_p = \mathbb{R}^p$ after a change of coordinates (Bozdogan, 1987; deLeeuw, 1992).

When using the maximized log-likelihood $\ell_{\hat{\boldsymbol{\theta}}_k}(\boldsymbol{Y}) := \log f_{\hat{\boldsymbol{\theta}}_k}(\boldsymbol{Y})$ to estimate (1.5), a bias correction must be used to correct for the dependence of $\ell_{\hat{\boldsymbol{\theta}}_k}(\boldsymbol{Y})$ on the given realization $\boldsymbol{Y}$ (see, e.g., Greven and Kneib, 2010). For this purpose, Akaike proposed an adjustment constant $2\,\Psi := \mathrm{AI} - \mathbb{E}_{\mathcal{G}_Y}[2\,\ell_{\hat{\boldsymbol{\theta}}_k}(\boldsymbol{Y})]$, which can be estimated by twice the dimension of $\boldsymbol{\theta}$ under certain assumptions. The Akaike Information Criterion as an asymptotic unbiased estimator of the AI is then defined by

$$\mathrm{AIC}(\hat{\boldsymbol{\theta}}_k) := -2\,\ell_{\hat{\boldsymbol{\theta}}_k}(\boldsymbol{Y}) + 2k. \tag{1.6}$$

In this context, the bias correction term is also referred to as *degrees of freedom* (see e.g. Vaida and Blanchard, 2005).

The derivation of the AIC in (1.6) is based on certain regularity conditions, such as i.i.d. observations $Y_1, \ldots, Y_n$ and the assumption that the parameter space is a transformation of $\mathbb{R}^p$. These conditions are not fulfilled in certain modeling approaches, in particular, for repeated measurements and the linear mixed model. In this light, Greven and Kneib (2010) proposed an extension of the AIC for LMMs, which is presented in the first contributing article.

1.3 The History of Statistical Boosting

This section gives background information on the use of boosting algorithms in statistical applications and, especially, introduces the component-wise functional gradient descent algorithm, which is employed and studied in various ways in the Chapters 4, 5 and 6 as model selection as well as estimation technique.

1.3.1 AdaBoost

The idea of boosting was originally proposed for binary classification problems by Schapire (1990), Freund (1995) and Freund and Schapire (1997) under the name *AdaBoost* with the purpose of "converting a 'weak' [...] learning algorithm that performs just slightly better than random guessing into one with arbitrarily high accuracy" (Freund and Schapire, 1997). The basic principle is to repeatedly apply the "weak" learner or *base learner*

$$\begin{aligned} g(\boldsymbol{y}, \boldsymbol{X}, \cdot): \quad & \mathbb{R}^n \longrightarrow \mathbb{R}^n \\ & \boldsymbol{w}^{[m]} \longmapsto g(\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{w}^{[m]}) =: \hat{\boldsymbol{g}}^{[m]} \end{aligned}$$

to the (reweighted) dataset $(\boldsymbol{y}, \boldsymbol{X})$ with weights $\boldsymbol{w}^{[m]}$ for $m = 1, \ldots, M$ iterations. In each iteration larger weights $w_i^{[m]}, i = 1, \ldots, n$, are given to observations y_i that had been predicted poorly in the previous iteration $m-1$ with $w_i^{[1]} \equiv n^{-1} \, \forall \, i \in \{1, \ldots, n\}$. The boosted learner $\hat{\boldsymbol{f}}^{[M]}$ is finally given as a majority vote $\hat{\boldsymbol{f}}^{[M]} = \text{sign}(\sum_{m=1}^{M} \hat{\boldsymbol{g}}^{[m]})$ over all model fits $\hat{\boldsymbol{g}}^{[m]}$ of all iterations $m = 1, \ldots, M$.

1.3.2 Boosting from a Statistical Point of View

Although the idea of combining or "mixing" models was not completely new at the time of AdaBoost's invention, a theoretical justification for the good prediction performance from a statistical point of view was only given later by Friedman et al. (2000), who showed that AdaBoost is actually an optimization method to minimize a particular exponential loss (Ridgeway, 1999). The relevance of this new concept was soon accepted in the statistical community and on the basis of the initial idea to reweight observations Ridgeway (1999) and Friedman (2001) proposed to iteratively update the final model in a gradient descent manner using a small step-length $\boldsymbol{\nu} \in (0, 1)$: $\hat{\boldsymbol{f}}^{[m]} = \hat{\boldsymbol{f}}^{[m-1]} + \nu \hat{\boldsymbol{g}}^{[m]}$. Therefore, the base procedure

$$g(\boldsymbol{X}, \cdot) : \boldsymbol{u}^{[m]} \mapsto g(\boldsymbol{X}, \boldsymbol{u}^{[m]}) = \hat{\boldsymbol{g}}^{[m]}$$

is iteratively fitted to the negative functional gradient

$$\boldsymbol{u}^{[m]} := - \frac{\partial}{\partial \boldsymbol{f}} \mathfrak{v}(\boldsymbol{y}, \boldsymbol{f}) \Big|_{\boldsymbol{f} = \hat{\boldsymbol{f}}^{[m-1]}(\boldsymbol{X})},$$

which can be seen as a measure of missing adjustment of the current model fit $\hat{\boldsymbol{f}}^{[m-1]}$ to the data with respect to a specified loss function $\mathfrak{v}(\cdot, \cdot)$. Hence, instead of fitting $g(\cdot)$ to a weighted dataset, their proposal is to fit the base procedure to the *working response* or *pseudo residuals* $\boldsymbol{u}^{[m]}$ and update the model incrementally in the direction of the negative functional gradient, thereby heralding the era of functional gradient descent (FGD) algorithms[1].

[1] However, literature does not fully agree on who should be given credit for pointing out the idea of performing "gradient descent in the function space" in the first place (see Bühlmann and Hothorn, 2007; Buja et al., 2007).

In the context of statistical boosting with squared error loss, Bühlmann and Yu (2003) first recognized the necessity to employ some kind of variable selection when boosting is applied to high-dimensional data sets. Inspired by tree base learners, they proposed component-wise smoothing splines as learners, where only one smoothing spline base learner $g_j(\boldsymbol{x}_j)$ corresponding to one explanatory variable $\boldsymbol{x}_j$ is selected in each iteration, entitling the resulting algorithm *L_2Boost*. Bühlmann (2006) adopted this idea and proposed an extension under the name L_2Boosting, which was a general framework for boosting linear models for squared error loss in high dimensions, equipped with component-wise linear least squares base learners. In addition, a justification for L_2Boosting was given by proofing asymptotic consistency of the procedure in high dimensions. By fitting the base learner $g_j(\cdot), j = 1, \ldots, J$, separately, this concept allows to (a) fit regression models for $p > n$-settings, i.e., when the number of covariates exceeds the number of observations, (b) include correlated covariates into the regression setup without any further adjustment and (c) leads to a computationally desirable scaling with respect to the number of covariates, as calculations involved in least squares minimization typically imply costs, which are quadratic and cubic in p (see, e.g., Wood, 2017). Additionally, the algorithm potentially performs variable or model selection – depending on the definition of base learners – when stopped before convergence.

1.3.3 Implementation and Extensions

A wrap-up of past discoveries together with a proposal of a comprehensive framework for statistical boosting accompanied with a modular open-source R package `mboost` (Hothorn et al., 2017) was given by Bühlmann and Hothorn (2007). The authors' implementation of the component-wise functional gradient descent algorithm allows for a flexible definition of the loss function as well as arbitrary and potentially different base learners, such as a combination of linear and smoothing spline base learners. Schmid and Hothorn (2008) proposed to use P-spline instead of smoothing spline base learners when boosting additive models, which was particularly motivated from a computational point of view. Having similar prediction performance in practice, P-spline base learners reduce computational effort notably in contrast to smoothing spline base learners and represent a standard choice in more recent literature as well as in the software package `mboost`.

In the last ten years, boosting has been applied in various forms, from machine learning and data mining challenges, where especially tree-based boosting methods such as XGBoost (Chen and Guestrin, 2016) are very successful, to statistical boosting applications covering a great variety of model classes, including models for regression with functions, survival regression or generalized additive models for location, scale and shape (GAMLSS, Rigby and Stasinopoulos, 2005). A more detailed overview on the application of statistical boosting can be found in Mayr et al. (2017).

1.4 Post-Selection Inference

This section of the methodological introduction is concerned with post-selection inference (PoSI), which plays an important role in the contributing articles in Chapter 3 and Chapter 4. First, the idea behind PoSI is explained by reviewing the recent criticism of scientific publications and some of the misconceptions therein. After a brief summary of the historical evolution of PoSI, a more extensive overview of the current literature is given, including some detailed reviews of proposed PoSI concepts. The general idea of the problems underlying the PoSI framework is exemplarily illustrated on the basis of model selection in the class of linear models.

1.4.1 The Problem with Classical Statistical Inference after Model Selection

In recent years the replicability of published research findings has been subject to substantial criticism. One of the driving publications raising awareness of this "crisis" is Ioannidis (2005), who bluntly questions the correctness of publications, claiming that "most research findings are false". Besides other more intangible problems he relates this incorrectness to a prevailing lack of study power, different types of (publication) biases and greater flexibility in study definitions. A somewhat related problem to the greater flexibility in study definitions comes from misconceptions in the use of classical statistical theory after data-driven model selection. Whereas classical inference concepts grant validity if the model of interest as well as hypothesis are known a priori, the data-driven selection of a model and corresponding hypotheses produces an additional stochastic aspect in the analysis, which classical theory does not account for. This is not only the case if inference statements and hypothesis are generated as a result of a formally specified model selection procedure, but is also problematic for ill-defined ways of model selection, such as visual inspection or retrospective adaption of models.

In the last century, many authors have noticed problems associated with inference after model selection, dating at least back to Buehler and Feddersen (1963). Although many theoretical results have been discovered in the 90's as well as in the beginning of this century, e.g., by Pötscher (1991) and Leeb and Pötscher (2003) deriving the (asymptotic) conditional and unconditional finite-sample distribution of the post-model-selection estimator, the literature indicates only minor interest in this topic until 2013. After the proposal by Berk et al. (2013) to "devise statistical inference that is valid following any type of variable selection", research on this topic flourished and a rising interest in the statistical community for methods correcting for the data-driven or adaptive nature of statistical methodology in practice can be derived from literature.

Berk et al. (2013) stipulated an inference that is simultaneously valid for all hypotheses potentially coming into question. Simultaneity in this case refers to the validity of the concept for all possible selected models. Whereas simultaneous inference can be seen as a protection against all risks of data exploitation, the principle is often very idealistic and may be difficult to translate in practice. Although Berk et al. (2013) describe their framework as an inference tool, which is valid after any kind of model selection, the statements are only valid for all models within the class of linear models considered and do, e.g., not cover the case, in which additive models have been considered in the model building process. In addition, the premise for a protection against all "risks" can result in a

rather conservative inference framework, for instance a lack of power in hypothesis tests following the model selection.

Another approach, which is related to the proposal by Berk et al. (2013), is given by performing inference conditional on the selected model. Rather than providing validity for any model selection procedure in a class of models, the idea is to build a valid inference framework for specific model selection procedures conditional on the finally selected model, which is also referred to as *selective inference*. As can be seen in the contributing article in Chapter 3 and 4, this approach allows for the combination of different selection procedures as long as each of the selection mechanisms is accounted for and thus also facilitates a model search through different model classes. The contributing articles in Chapter 3 and 4 are based on this concept. Therefore the conditional or selective inference is first described in Subsection 1.4.2 in more detail, followed by a short introduction into simultaneous inference and alternative concepts in Subsection 1.4.3. The idea of conditional inference is exemplified in the following example.

Example: Inference after AIC-based model selection

As described in the first section of this methodological introduction, the AIC can be used as a measure of prediction performance of candidate models in GLMs. Specifically, if the goal is to decide whether a covariate of interest $\boldsymbol{x}_j$ with coefficient β_j should be added to an existing model $\mathcal{M}_0$, the decision can be based on the comparison of AICs for the model $\mathcal{M}_0$ and for the model $\mathcal{M}_1$, which is the same model as $\mathcal{M}_0$ but additionally includes the covariate $\boldsymbol{x}_j$. Since model $\mathcal{M}_1$ is chosen if and only if

$$\text{AIC}(\mathcal{M}_0) > \text{AIC}(\mathcal{M}_1) \quad \Leftrightarrow \quad -2\ell(\mathcal{M}_0) > -2\ell(\mathcal{M}_1) + 2 \tag{1.7}$$

for the maximized log-likelihoods $\ell(\mathcal{M}_0)$, $\ell(\mathcal{M}_1)$ under model assumptions $\mathcal{M}_0$, $\mathcal{M}_1$, respectively, (1.7) implies that minus twice the logarithmic likelihood ratio $\Lambda := -2\log[L(\mathcal{M}_0)/L(\mathcal{M}_1)]$ must be greater than 2. Under the assumption

$$H_0 : \beta_j = 0,$$

i.e., no influence of the jth covariate in the alternative model, the hypothesis can be tested using a likelihood ratio test (LRT), where $\Lambda \overset{a}{\sim} \chi^2_1$ (see, e.g., Pawitan, 2001). However, if the model $\mathcal{M}_1$ was chosen by the AIC, i.e., $\text{AIC}(\mathcal{M}_0) > \text{AIC}(\mathcal{M}_1)$,

$$\mathbb{P}(\Lambda \leq 2 | \text{AIC}(\mathcal{M}_0) > \text{AIC}(\mathcal{M}_1)) = 0$$

holds.

The invalidity of inference after model selection can then be exemplarily explained when the number of falsely rejected null hypothesis in a number of experiments is considered. For demonstrating purposes, assume that the above model comparison and subsequent test is repeated

for B times by a) repeatedly sampling a new response vector $\boldsymbol{y}$ from the true underlying data generating process, b) conducting the AIC comparison and c) testing the jth coefficient if the larger model is preferred by the AIC. This example can be thought of as a research question, which is investigated by B independent researcher teams conducting one and the same experiment. Assume that H_0 holds for all B cases and set the significance level α to 0.05. Then, in an average of $\mathbb{P}(\Lambda \leq 2) \cdot B \approx 0.8427 \cdot B$ cases the LRT is not performed and thus H_0 cannot be falsely rejected. In $\mathbb{P}(\Lambda > 2) \cdot B \approx 0.1573 \cdot B$ expected number of cases, however, the LRT is performed and yields and expected number of

$$\mathbb{P}(\Lambda > q_{0.95}^{\Lambda} | \Lambda > 2) \cdot B = \mathbb{P}(\Lambda > q_{0.95}^{\Lambda}) / \mathbb{P}(\Lambda > 2) \approx 0.3179 \cdot B$$

false rejections, where $q_{0.95}^{\Lambda} \approx 3.8415$ is the $(1-\alpha)$-quantile of the χ_1^2 distribution. Although this results in an overall expected number of $\alpha \cdot B$ false rejections, the number of false rejections in those cases, in which the variable was selected into the model is $\approx 0.3179 \cdot B \gg \alpha \cdot B$. As in practice, significance tests are only ever performed for coefficients, for which the corresponding covariate has been selected by the model selection procedure, this example elucidates the importance of valid inference concepts after model selection.

1.4.2 Selective Inference

An important concept for valid inference after model selection can be obtained when hypothesis tests and other inference statements are conducted conditional on the model selection. This idea has already been developed some time ago, including Buehler and Feddersen (1963), Brown (1967), Olshen (1973) and Sen (1979), who focused on conditional properties of tests and the maximum-likelihood estimators. Prior to the literature, which will be introduced below, the phrase *selective inference* was also shaped by approaches, which are concerned with statistical properties after multiple testing procedures. Most notably, the work of Benjamini and Yekutieli (2005), who generalized the *false discovery rate* (Benjamini and Hochberg, 1995)

$$\text{FDR} = \mathbb{E}\left[\frac{\#\text{false discoveries}}{\#\text{discoveries}}\right]$$

to a *false coverage-statement rate* (FCR) in order to account for a selective nature of performed hypothesis tests.

Selective inference as the concept of controlling the *selective type I error* can be based on the guiding principle "*The answer must be valid, given that the question was asked*" (Fithian et al., 2014). More formally, let $\hat{\mathcal{Q}} : \mathcal{Y} \to \mathcal{Q}$ be a pre-defined selection procedure mapping the data $\boldsymbol{Y} \sim \mathcal{F}$ from some measurable space $(\mathcal{Y}, \mathfrak{H})$ to the model or "*question space*" $\mathcal{Q}$ with elements $q = (\mathcal{M}, H_0)$, i.e., a hypothesis generating probability model $\mathcal{M}$, which is believed to – but does not necessarily have

to – contain $\mathcal{F}$, and a null hypothesis $H_0 \subset \mathcal{M}$. Selective inference then considers the conditional distribution

$$\boldsymbol{Y}|q \in \hat{\mathcal{Q}}(\boldsymbol{Y}),$$

i.e., the distribution of $\boldsymbol{Y}$ conditional on the *selection event* $\mathcal{A} = \{q \in \hat{\mathcal{Q}}(\boldsymbol{Y})\}$ and seeks to control the selective type I error at level α:

$$\mathbb{P}(A_1|\mathcal{A}) \leq \alpha,$$

where A_1 denotes the rejection of the null hypothesis by some test $\phi \in \{0, 1\}$. Analogous to the classical statistical inference theory, a test ϕ controls this error at level α if

$$\mathbb{E}_{\mathcal{F}}(\phi(\boldsymbol{Y})|\mathcal{A}) \leq \alpha, \qquad \text{for all } \mathcal{F} \in H_0$$

and this statement can be used to construct a *selective (confidence) interval* by duality of tests and confidence sets (see Fithian et al., 2014).

When ignoring the selection event $\mathcal{A}$ or if $\hat{\mathcal{Q}}$ is a selection procedure independent of $\boldsymbol{Y}$, the selective type I error coincides with the conventional type I error: $\mathbb{P}(A_1|\mathcal{A}) = \mathbb{P}(A_1)$. Independence of the selection event and hypothesis testing can also be accomplished in a more synthetic way by using *data splitting* as proposed by Cox (1975). The idea is to split the data $\boldsymbol{Y}$ into two independent parts $\boldsymbol{Y}^{(1)} \in \mathbb{R}^{n_1}$ and $\boldsymbol{Y}^{(2)} \in \mathbb{R}^{n_2}$ with $n_1, n_2 < n, n_1 + n_2 = n$, define $\hat{\mathcal{Q}}$ using $\boldsymbol{Y}^{(1)}$, i.e., perform model selection using only $\boldsymbol{Y}^{(1)}$, and use $\boldsymbol{Y}^{(2)}$ to test the hypothesis H_0. Usually, if n is not very large, this split affects the performance of both the model selection, yielding a potentially decrease in "model selection quality", and the following inference, which will have a decrease in power in comparison to a test, which uses n instead of n_2 observations. By conditioning on the selection event itself, selective inference provides a more elegant way to use $\boldsymbol{Y}$ for both, model selection and inference by conditioning on the information in $\boldsymbol{Y}$, which was used for model selection. This approach is therefore also known as *data carving* (Fithian et al., 2014).

To calculate p-values for such hypothesis tests in practice, different approaches exist. In particular, methods can be divided into approaches, which derive a closed form expression of the conditional null distribution of some test statistic T (e.g., Lee et al., 2016; Tibshirani et al., 2016) and thus provide exact inference based on analytic expressions, and approaches, which sample from the conditional distribution and rely on Monte Carlo methods (e.g., Fithian et al., 2014; Tian et al., 2016; Tian and Taylor, 2018).

The following example serves as a short introduction into recent advances in conditional inference as well as an geometrical illustration of the underlying problem.See Lee et al. (2016) and Tibshirani et al. (2016) for more details.

Example: Exact Inference based on the Polyhedral Conditioning Sets

Assume a linear regression setup with a response vector $\boldsymbol{Y} \in \mathbb{R}^n$ following a normal distribution $\mathcal{N}_n(\boldsymbol{\mu}, \sigma^2 \boldsymbol{I}_n)$ with known variance σ^2. Samples are obtained from the conditional distribution

$\boldsymbol{Y}|\boldsymbol{X}$ with fixed design matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$. $\hat{\mathcal{Q}}$ here can be thought of as some sort of variable selection criterion, which selects the columns $\mathfrak{T} \in \mathcal{P}(\{1, \dots, p\}) \backslash \emptyset$ of $\boldsymbol{X}$ with power set function $\mathcal{P}(\cdot)$ and $p_{\mathfrak{T}} = |\mathfrak{T}| \leq n$. Inference is then sought for the model

$$\mathcal{M}(\boldsymbol{\beta}_{\mathfrak{T}}) = \left\{ \mathcal{N}_n(\boldsymbol{X}_{\mathfrak{T}} \boldsymbol{\beta}_{\mathfrak{T}}, \sigma^2 \boldsymbol{I}_n), \boldsymbol{\beta}_{\mathfrak{T}} \in \mathbb{R}^{p_{\mathfrak{T}}} \right\}$$

and some hypothesis

$$H_{0,j} : \beta_{\mathfrak{T}_j} = 0.$$

When $\boldsymbol{\mu} \neq \boldsymbol{X}_{\mathfrak{T}} \boldsymbol{\beta}_{\mathfrak{T}}$ for any $\mathfrak{T}$, the hypothesis $H_{0,j}$ can be regarded as testing the jth direction when projection $\boldsymbol{\mu}$ onto span$(\boldsymbol{X}_{\mathfrak{T}})$. This idea is described in more detail in Chapter 3.

Let $T := T(\boldsymbol{Y}) = \boldsymbol{\eta}^\top \boldsymbol{Y} \in \mathbb{R}$ be a test statistic linearly depending on $\boldsymbol{Y}$ via the vector $\boldsymbol{\eta} \in \mathbb{R}^n$. Further, assume σ^2 to be known and let $\boldsymbol{\mu}$ be the parameter of interest. The distribution of T is then given by $\mathcal{F} = \mathcal{N}_1(\boldsymbol{\eta}^\top \boldsymbol{\mu}, \tilde{\sigma}^2 := \sigma^2 ||\boldsymbol{\eta}||_2^2)$ with $|| \cdot ||_2^2$ the quadratic euclidean norm. In the case, in which we want to test $H_{0,j}$, we obtain a suitable test statistic T by choosing $\boldsymbol{\eta}$ as $(\boldsymbol{e}_j^\top (\boldsymbol{X}_{\mathfrak{T}}^\top \boldsymbol{X}_{\mathfrak{T}})^{-1} \boldsymbol{X}_{\mathfrak{T}}^\top)^\top$ with jth unit vector $\boldsymbol{e}_j$. When model or variable selection $\hat{\mathcal{Q}}$ is regarded as a function of $\boldsymbol{Y}$, valid inference can be based on the conditional distribution of $\boldsymbol{Y}$, conditional on the selection event $q \in \hat{\mathcal{Q}}(\boldsymbol{Y})$. For approaches, which provide exact inference such as Lee et al. (2016), the model selection event $q \in \hat{\mathcal{Q}}(\boldsymbol{Y})$ can be equivalently written as a restriction $\mathfrak{G} \subset \mathbb{R}^n$ on the space, in which $\boldsymbol{Y}$ resides. Such a restriction can, for example, be a *hyperplane*

$$\mathfrak{G} = \{\boldsymbol{Y} : \mathfrak{a}^\top \boldsymbol{Y} = \mathfrak{b}\},$$

with $\mathfrak{a} \in \mathbb{R}^n$, $\mathfrak{a} \neq \boldsymbol{0}$, $\mathfrak{b} \in \mathbb{R}$; a *polyhedron*

$$\mathfrak{G} = \{\boldsymbol{Y} : \mathfrak{a}_j^\top \boldsymbol{Y} \leq \mathfrak{b}_j, j = 1, \dots, e, \, \mathfrak{c}_j^\top \boldsymbol{Y} \leq \mathfrak{d}_j, j = 1, \dots, f\} \tag{1.8}$$

or other subspaces of $\mathbb{R}^n$ such as an intersection or union of polyhedra (see Boyd and Vandenberghe, 2004, for formal definitions and further examples). When the Lasso (Tibshirani, 1996) or the L_2Boosting algorithm is used for model selection, the space of $\boldsymbol{Y}$ is restricted to a union of polyhedra (see Lee et al., 2016, and the contributing article in Chapter 4).

If the distribution $\mathcal{E}$ of $T(\boldsymbol{Y})|\boldsymbol{Y} \in \mathfrak{G}$ is sought and the space restriction is only a single polyhedron, $\mathcal{E}$ can be obtained by rewriting $\boldsymbol{Y} \in \mathfrak{G}$ in terms of T. For simplicity, assume that $\mathfrak{G}$ in (1.8) is only defined by inequalities associated with $\mathfrak{a}_j, j = 1, \dots, e$ and note that $\boldsymbol{Y}$ can be decomposed in $\boldsymbol{P}_{\boldsymbol{\eta}} \boldsymbol{Y}$, with projection matrix $\boldsymbol{P}_{\boldsymbol{\eta}} = \boldsymbol{\eta}\boldsymbol{\eta}^\top / ||\boldsymbol{\eta}||_2^2$ projecting $\boldsymbol{Y}$ onto $\boldsymbol{\eta}$ and the residual $\boldsymbol{\mathfrak{Z}} := (\boldsymbol{I}_n - \boldsymbol{P}_{\boldsymbol{\eta}})\boldsymbol{Y}$. Plugging in $\boldsymbol{Y} = \boldsymbol{P}_{\boldsymbol{\eta}} \boldsymbol{Y} + \boldsymbol{\mathfrak{Z}} = T \cdot \boldsymbol{\eta} / ||\boldsymbol{\eta}||_2^2 + \boldsymbol{\mathfrak{Z}}$ in the definition of $\mathfrak{G}$ yields the conditioning set $\tilde{\mathfrak{G}}$ with respect to T:

$$\tilde{\mathfrak{G}} := \{T : T \cdot \mathfrak{a}_j^\top \boldsymbol{\eta} / ||\boldsymbol{\eta}||_2^2 + \mathfrak{a}_j^\top \boldsymbol{\mathfrak{Z}} \leq \mathfrak{b}_j, j = 1, \dots, e\}. \tag{1.9}$$

By solving the inequalities for T, the conditional distribution of T can be written as

$$T \mid \boldsymbol{Y} \in \mathfrak{G} \quad =^d \quad T \mid \mathcal{V}^-(\mathfrak{Z}) \leq T \leq \mathcal{V}^+(\mathfrak{Z}), \mathcal{V}^0(\mathfrak{Z}) \geq 0,$$

where

$$\mathcal{V}^-(\mathfrak{Z}) := \max_{j:\mathfrak{a}_j^\top \boldsymbol{\eta}<0} \frac{\mathfrak{b}_j - \mathfrak{a}_j^\top \mathfrak{Z}}{\mathfrak{a}_j^\top \boldsymbol{\eta}} \cdot ||\boldsymbol{\eta}||_2^2, \; \mathcal{V}^+(\mathfrak{Z}) := \min_{j:\mathfrak{a}_j^\top \boldsymbol{\eta}>0} \frac{\mathfrak{b}_j - \mathfrak{a}_j^\top \mathfrak{Z}}{\mathfrak{a}_j^\top \boldsymbol{\eta}} \cdot ||\boldsymbol{\eta}||_2^2, \; \mathcal{V}^0(\mathfrak{Z}) := \min_{j:\mathfrak{a}_j^\top \boldsymbol{\eta}=0} \mathfrak{b}_j - \mathfrak{a}_j^\top \mathfrak{Z}.$$

For every realization $\mathfrak{z}$ of $\mathfrak{Z}$, T then follows a normal distribution with fixed truncations $\mathcal{V}^-(\mathfrak{z})$ and $\mathcal{V}^+(\mathfrak{z})$, which by construction are independent of T. The truncation limits being theoretically random as functions of $\mathfrak{Z}$, selective inference proceeds by conditioning on the realized value of $\boldsymbol{Z}$ for a given observation $\boldsymbol{Y} = \boldsymbol{y}$. Doing so facilitates the derivation of an explicit distribution at the cost of conditioning on more information and thereby a potentially loss of power for the subsequent inference. When conditioning on $\mathfrak{Z} = \mathfrak{z}$, a pivotal quantity can be defined as

$$F_{\boldsymbol{\eta}^\top \boldsymbol{\mu}, \tilde{\sigma}^2}^{[\mathcal{V}^-(\mathfrak{z}), \mathcal{V}^+(\mathfrak{z})]}(T) \mid \mathcal{V}^-(\mathfrak{z}) \leq T \leq \mathcal{V}^+(\mathfrak{z}), \mathcal{V}^0(\mathfrak{z}) \geq 0, \mathfrak{Z} = \mathfrak{z} \quad \sim \quad \text{Unif}(0,1),$$

with

$$F_{\psi,\omega^2}^{[a,b]}(x) = \frac{\Phi((x-\psi)/\omega) - \Phi((a-\psi)/\omega)}{\Phi((b-\psi)/\omega) - \Phi((a-\psi)/\omega)}$$

denoting the cumulative distribution function (CDF) of a truncated normal distribution with truncation limits a, b, expectation ψ and variance ω^2. Φ denotes the CDF of a standard normal random variable. Finally, the pivotal quantity can be used to test the hypothesis $H_{0,j}$ based on T and can also be used to construct confidence intervals by inverting the test.

Further Developments in Selective Inference

The following tries to give a snapshot of current developments and the many facets of selective inference. Due to the high topicality, this includes many preprints, which may be work in progress and does not guarantee completeness given the speed of growth of this research field. The focus of this summary lies on selective inference. Many other potentially relevant methods including concepts for simultaneous inference are listed afterwards in order to provide a bigger picture on the topic of valid inference post-model selection.

Many other explicit inference approaches are motivated by the exact post-selection inference framework of Lee et al. (2016), proposing a general approach for valid inference after model selection with particular focus on the Lasso. This work has been extended in several ways, e.g., by tests for groups of variables initially proposed by Loftus and Taylor (2015). Yang et al. (2016) further extended this idea in order to calculate p-values beyond the null hypothesis $H_{0,j} : \beta_{\mathfrak{T}_j} = 0$ and thereby allow for the construction of confidence intervals. Both of these publications represent an important founda-

tion of the work in Chapter 3 and 4 and will be discussed in greater detail in those contributing articles. Different approaches also allow the regularization parameters of the Lasso to be chosen via cross-validation by extending existing inference concepts for fixed regularization parameter (see, e.g., Loftus, 2015; Markovic et al., 2017). An explicit framework for selective inference has also been established for sequential selection procedures (G'Sell et al., 2016; Tibshirani et al., 2016) such as forward stagewise regression, for which an explicit conditional distribution can be derived in a similar manner as for the Lasso. The second and third contributing article will elaborate on this framework in more detail.

An important milestone in the evolution of selective inference is given by Fithian et al. (2014) and Tian and Taylor (2018), introducing the concept of randomization to obtain more power when conducting selective inference. The principle idea is to introduce a **known** randomization distribution Ω as well as a randomization map $\pi : \mathcal{Y} \times \mathcal{R} \to \mathcal{Y}^*$, where $\mathcal{R}$ is an auxiliary probability space, and to fit the regression model using a randomized response $\boldsymbol{Y}^* = \pi(\boldsymbol{Y}, \boldsymbol{\omega})$, where $\boldsymbol{\omega} \sim \Omega$. A possible choice for π is $\pi(\boldsymbol{y}, \boldsymbol{\omega}) = \boldsymbol{y} + \boldsymbol{\omega}$, i.e., defining the randomization as an additive noise or by defining the response for model selection as a random subsample of $\boldsymbol{y}$, which then coincides with data splitting. For randomized selective inference, the model selection procedure can be defined by $\hat{\mathcal{Q}}^* : \mathcal{Y} \times \mathcal{R} \to \mathcal{Q}$ and inference is based on the conditional distribution of $\boldsymbol{Y}|q \in \hat{\mathcal{Q}}^*(\boldsymbol{Y}^*)$. For the linear regression setup and additive randomization noise, for example, the condition in (1.9) is replaced by

$$\tilde{\mathfrak{G}}^* := \{T : T(\boldsymbol{Y}^*) \cdot \mathfrak{a}_j^\top \boldsymbol{\eta}/||\boldsymbol{\eta}||_2^2 + \mathfrak{a}_j^\top \mathfrak{Z} \leq \mathfrak{b}_j, j = 1, \ldots, e\}$$

with $T(\boldsymbol{Y}^*) = T(\boldsymbol{Y} + \boldsymbol{\omega})$ being distributed according to the *selective distribution* or *selective law*: $T(\boldsymbol{Y} + \boldsymbol{\omega}) \sim \mathcal{F}^*$. Although an explicit derivation of the conditional set is not possible in this case due to the randomization, the restricted space, again, is a polyhedron, which can be explored using a sampling algorithm such as a hit-and-run Gibbs sampler or a Hamiltonian Monte Carlo algorithm (Fithian et al., 2014; Tian and Taylor, 2018). The randomization concept cannot only be used to derive weak convergence results for selective inference procedures, but has the advantage of increasing the power of statistical inference while only slightly affecting the model selection quality at the same time for appropriate randomization schemes. The increase in power is due to more leftover information after model selection for the inferential procedure, which can be quantified by the so-called *leftover Fisher information* (see Tian and Taylor, 2018, Section 4.2 for more details).

Tian Harris et al. (2016) consider the general problem of selective inference after solving a convex optimization problem stemming from a regularized as well as constrained loss function and propose a projected Langevin sampler to sample from the selective distribution. This sampling procedure requires knowledge of the exact or asymptotic distribution of the data generating process, wherefore Markovic and Taylor (2016) extend this idea in order to use the bootstrap distribution instead. Panigrahi et al. (2017) extended the framework of Tian Harris et al. (2016) to an Monte Carlo free approach based on a pseudo selective law. The application of these approaches require the optimization problem to have a closed form expression. Despite the fact, that some special cases of model-based boosting can be related to known optimization problems, namely the Lasso, LARS or

forward stagewise regression (Efron et al., 2004; Bühlmann and Yu, 2003), the (penalty terms of the) target function in model-based boosting are in general unknown (Hothorn et al., 2014; Mayr et al., 2017).

Although many approaches focus on linear regression, several frameworks provide methods or theory for a broader class of models, such as Fithian et al. (2014) considering distributions from the exponential family or Tibshirani et al. (2015) and Tian and Taylor (2017) removing the Gaussian assumption of most previous works on selective inference by considering the large sample properties and convergence behavior of the pivot established in Lee et al. (2016).

Furthermore, selective inference frameworks for several other methods have been proposed recently, including selective inference for the use in model approaches with internal predictors (Gross et al., 2015), for the Lasso in specific causal inference problems (Zhao et al., 2017), selective inference to adjust for outlier removal (Chen and Bien, 2017) as well as selective inference for change point detection (Umezu and Takeuchi, 2017).

Selective inference has also been applied in Bayesian analysis by Panigrahi et al. (2016). Based on an idea of Yekutieli (2012), the authors first investigate the two possible schemes of space truncation of $\boldsymbol{Y}$ induced by model selection. From a Bayesian point of view, the truncation $\boldsymbol{Y} \in \mathfrak{G}$ may come from a "random" parameter setting or a "fixed" parameter setting. In the random parameter setting, the truncation $\boldsymbol{Y} \in \mathfrak{G}$ is given for some $\boldsymbol{Y}$ sampled alongside with a parameter of interest $\boldsymbol{\theta}$, i.e., when sampling pairs $(\boldsymbol{Y}, \boldsymbol{\theta})$. In the fixed parameter setup, some fixed $\boldsymbol{\theta} \sim \Theta$ is realized and afterwards $\boldsymbol{Y}$ is sampled based on this realization of $\boldsymbol{\theta}$. In both cases, the truncated joint distribution of $(\boldsymbol{Y}, \boldsymbol{\theta})$ with density f_S can be derived by

$$f_S(\boldsymbol{\theta}, \boldsymbol{y}) = \frac{f_{\mathfrak{G}}(\boldsymbol{\theta}) f(\boldsymbol{y}|\boldsymbol{\theta})}{\mathbb{P}_{\mathfrak{G}}} I(\boldsymbol{y} \in \mathfrak{G}).$$

In the "random" parameter setting, $\mathbb{P}_{\mathfrak{G}} = \mathbb{P}(\boldsymbol{Y} \in \mathfrak{G})$ and $f_{\mathfrak{G}}(\boldsymbol{\theta})$ is equal to the prior $f(\boldsymbol{\theta})$, whereas in the "fixed" parameter case $\mathbb{P}_{\mathfrak{G}} = \mathbb{P}(\boldsymbol{Y} \in \mathfrak{G}|\boldsymbol{\theta})$ and $f_{\mathfrak{G}}(\boldsymbol{\theta})$ is a prior $f(\boldsymbol{\theta}|\boldsymbol{Y} \in \mathfrak{G})$ based on the model selection. This makes clear, that for the first assumption, the posterior distribution of $\boldsymbol{\theta}$ does not change, as for $\boldsymbol{Y} \in \mathfrak{G}$ the posterior $f(\boldsymbol{\theta}|\boldsymbol{y}) \propto f_S(\boldsymbol{\theta}, \boldsymbol{y}) \propto f(\boldsymbol{\theta}) f(\boldsymbol{y}|\boldsymbol{\theta})$. Panigrahi et al. (2016) take the position of the fixed parameter view, which disagrees with some earlier work in this field but can be justified when the distribution assumption of $\boldsymbol{\theta}$ is viewed as summary of available prior information. Panigrahi and Taylor (2017) adopt this methodology and derive an optimization problem to approximate the posterior as well as proposing a sampling technique to reduce the computational cost of the problem.

1.4.3 Simultaneous Inference and Alternative Concepts

As described before, the principle idea of Berk et al. (2013), which initially unleashed the wave of valid inference concepts in the beginning of the early 2010s, is not to condition on the selection event but rather provide inference simultaneously for all possible model selection outcomes. Bachoc et al. (2014) extended the simultaneous inference framework of Berk et al. (2013) to valid confidence intervals for

predicted values post model selection. Whereas Berk et al. (2013) presented the PoSI idea for linear models with homoskedastic Gaussian error assumption, Bachoc et al. (2016) extended the framework to construct valid confidence intervals post model selection, allowing for different regression setups (e.g., logistic regression) and without the requirement of the existence of an unbiased or uniformly consistent estimator for σ^2. These confidence intervals (CI) guarantee coverage for any data-driven selection procedure and, similar to Berk et al. (2013), the CI is a function of K, the so-called *PoSI constant* (Berk et al., 2013). K as the most crucial component in those frameworks in turn depends on the question space $\mathcal{Q}$, i.e., the space of all considered models, where larger values of K yield to more conservative statements. Although quite general and attractive due to the validity for any selection mechanism, calculations may often not be feasible if the question space is large as the following example will briefly illustrate by giving an explicit construction for linear regression with homoskedastic Gaussian errors.

Example: Uniformly Valid Inference in Linear Regression

As in the previous example assume $|\mathfrak{T}| \leq n$ and let the target of inference be $\boldsymbol{\beta}_{\mathfrak{T}_j}$, the jth direction of the projection of $\boldsymbol{\mu}$ onto the span defined by $\boldsymbol{X}_{\mathfrak{T}}$. Following Berk et al. (2013), a valid post-selection confidence interval $\text{CI}_{\mathfrak{T}_j}(K)$ has the following guarantee:

$$\mathbb{P}(\forall j \in \mathfrak{T} : \boldsymbol{\beta}_{\mathfrak{T}_j} \in \text{CI}_{\mathfrak{T}_j}(K)) \geq 1 - \alpha. \tag{1.10}$$

This definition can be thought of as a family-wise guarantee for all $\boldsymbol{\beta}_{\mathfrak{T}_j}$ for which $j \in \mathfrak{T}$, but provides no statement about coefficients for which $j \notin \mathfrak{T}$. "*Universal validity for all selection procedures*" (Berk et al., 2013) additionally requires (1.10) to hold for all possible model selection procedures $\hat{\mathcal{Q}}$. A uniformly valid confidence interval Bachoc et al. (2016), which builds on this premise, is then given by

$$\text{CI}_{\mathfrak{T}_j}(K) = \hat{\boldsymbol{\beta}}_{\mathfrak{T}_j} \pm \sqrt{\hat{\sigma}^2 \left[(\boldsymbol{X}_{\mathfrak{T}}^\top \boldsymbol{X}_{\mathfrak{T}})^{-1}\right]_{[j,j]}} \cdot K(\mathfrak{M}),$$

where the notation $A_{[j,j]}$ denotes the jth diagonal element of a matrix A, $\hat{\sigma}^2$ is the empirical residual variance based on OLS estimation of the linear model with covariates in $\mathfrak{T}$. K, in this case, is a function of the block-matrix $\mathfrak{M}$, which is in turn defined by the blocks

$$\mathfrak{M}_{\mathfrak{T}^{(i)},\mathfrak{T}^{(j)}} = \boldsymbol{\eta}_{\mathfrak{T}^{(i)}}^\top \boldsymbol{\eta}_{\mathfrak{T}^{(j)}} \tag{1.11}$$

with pseudo-inverse matrices $\boldsymbol{\eta}_{\mathfrak{T}^{(i)}}, \boldsymbol{\eta}_{\mathfrak{T}^{(j)}}$ of two models $\mathcal{M}_i, \mathcal{M}_j$ searched in the model space. In linear models, for which the number of columns p in $\boldsymbol{X}$ corresponds to the number of covariates and for which all possible submodels of $\boldsymbol{X}$ can be selected by $\hat{\mathcal{Q}}$, $\mathfrak{M}$ consists of $2^p - 1$ blocks defined by (1.11) for all combinations of submodels $\mathcal{M}_i, \mathcal{M}_j, i, j \in \{1, \ldots, 2^p - 1\}$.

Many other approaches, which have been primarily presented in combination with Lasso-based model selection, can be summarized under the key word *high-dimensional linear model inference*. Rather than to correct inference, the idea in this case is the derivation of an (approximate) distribution of the parameter estimator. These high-dimensional inference concepts are to a large extent driven by publications from Zhang and Zhang (2014); van de Geer et al. (2014); Javanmard and Montanari (2014) and are based on a corrected, *desparsified* or *de-biased* version of the Lasso estimator for applications in which the number of columns p exceeds the number of observations ("$p > n$-case"). After correcting the Lasso or an alternative initial estimator, these approaches derive an asymptotic distribution based on an estimator with a "relaxed form" of the inverse of the empirical covariance $\hat{\boldsymbol{\Sigma}} = (\boldsymbol{X}^\top \boldsymbol{X}/n)$, which in turn can be used to provide p-values and confidence intervals. The work of Chen et al. (2016) is closely related, using the proposed inference concepts to derive a consistent estimator of the empirical covariance when using stochastic gradient boosting, for which iterative stochastic properties of the algorithm are exploited. Another approach based on the Lasso is given by Lu et al. (2017), who derive confidence intervals and regions for the Lasso estimator using stochastic variational inequality techniques. Meir and Drton (2017) provide inference for the Lasso by creating a noisy post-selection score function. Extensions to other model classes include, e.g., results on the likelihood ratio test in high-dimensional logistic regression (Sur et al., 2017) or valid PoSI in quantile regression models (Belloni et al., 2018). Ewald and Schneider (2015) derive confidence sets for the parameter vector based on the Lasso estimator. In the economic research field a line of post-selection inference frameworks for the Lasso but also for other machine learning techniques and, in particular, for L_2-Boosting have been proposed by Chernozhukov et al. (2015), Belloni et al. (2016), Chernozhukov et al. (2016) and Luo and Spindler (2017). The aim is to estimate causal and treatment effects in models that assume some form of endogeneity. Apart from methods, which mainly focus on the Lasso, PoSI concepts for many other methods have been proposed, e.g., by Yamada et al. (2018) proposing a kernel based PoSI algorithm.

Closely related to the problem of correct assessment of uncertainty after model selection is an appropriate performance measure in machine learning. Here model selection can be the reason for over-fitting and should be used as an integral part of the fitting procedure when estimating the generalization performance (Cawley and Talbot, 2010). The work of Hong et al. (2018), which is interesting from a predictive as well as from an inferential point of view, explicitly proved this phenomenon by showing that the estimated variance in a linear model that is selected via the AIC is strictly smaller than the oracle estimate.

1.5 Functional Data Analysis

Part II of this thesis addresses the estimation of functional regression models and their application as well as extension to studies, in which study settings vary between different observation units and additionally yield subject specific measurements. As a methodological basis this section therefore briefly introduces functional data analysis, functional regression models in general and the framework

of Brockhaus et al. (2017), which serves as a basis for Chapter 5.

The second part of this thesis is concerned with functional data analysis (FDA), nowadays one of the fastest growing fields in statistics, which has especially become popular due to the publication of Ramsay and Silverman (2005) describing different aspects of FDA in a "seminal textbook" (Morris, 2015). Strongly inspired by longitudinal and time series data, the analysis of "first generation functional data" or "curve data" (Wang et al., 2016) is concerned with samples $x_1(t), \ldots, x_n(t), t \in \mathcal{T} \subset \mathbb{R}$, representing real-valued functions on the domain $\mathcal{T}$. In applications, the domain $\mathcal{T}$ is often thought of as a certain time interval. These intrinsically infinite dimensional functions are viewed as realizations of a stochastic process $X : \mathcal{T} \to \mathbb{R}$, defined on a Hilbert space. In the the following and in both chapters in part II of this thesis this space is defined by $L^2(\mathcal{T}, \mu)$, the space of square integrable functions with Lebesgue measure μ, satisfying $\mathbb{E}(\int_{\mathcal{T}} X_i^2(t)\, d\mu(t)) < \infty$. In practice, the realized functions $x_i(\cdot)$ of samples $X_i(\cdot), i = 1, \ldots, n$ are observed on a grid of ordered time points $t_{1,i}, \ldots, t_{G_i,i} \in \mathcal{T}$ and are often summarized in a vector $\boldsymbol{x}_i = (x_i(t_{1,i}), \ldots, x_i(t_{G_i,i}))$. The time grid can be sparse or dense, depending on the observation mechanism and may have missing values. Thus an appropriate analysis of such data requires different methods, depending on the question of interest, the nature of the data generating process and on how the functions are observed.

Two further challenges are given when analyzing multivariate data of p stochastic processes $X_i^{(j)}, j = 1, \ldots, p$, each potentially with a different domain $\mathcal{T}_j$ or when the analysis is concerned with "next-generation functional data" (Wang et al., 2016) such as neuroimaging data, time-space data or shapes. In the latter case, the methodology of functional data is extended to stochastic processes, which are defined on higher dimensional domains $\mathcal{T} \subset \mathbb{R}^{\mathfrak{d}}, \mathfrak{d} \in \mathbb{N}$.

Analysis of functional data is done in various ways, including (functional) principal component analysis, clustering and classification of functional data, discriminant analysis or functional regression (see, e.g., Ramsay and Silverman, 2005; Wang et al., 2016). In the following, functional regression models are described in more detail.

1.5.1 Functional Regression Models

The second part of this thesis focuses on functional regression models, an area, that has received great attention with respect to the application of functional data as well as with respect to the development of new methodology (Morris, 2015). In the FDA literature, three types of functional regression models are usually addressed: (a) scalar-on-function regression (SOFR) with scalar response and functional covariate(s), (b) function-on-scalar regression (FOSR) with functional response and scalar covariate(s) and (c) function-on-function regression (FOFR), where both response and covariate(s) are considered as functions. Whereas Chapter 5 is only concerned with function-on-function regression, chapter 6 describes methods for all three model classes and illustrates how (a) and (b) can be represented as special cases of (c). For introductory purposes, a short introduction into function-on-function regression is given in the following.

1.5.2 Function-on-Function Regression

Consider a functional response Y and a functional covariate X from the product space $\mathcal{Y} \times \mathcal{X}$, where $\mathcal{Y}$ and $\mathcal{X}$ are elements of $L^2(\mathcal{T}, \mu)$ and $L^2(\mathcal{S}, \mu)$, respectively, with intervals $\mathcal{T} = [T_1, T_2]$ and $\mathcal{S} = [S_1, S_2]$ defined by $T_1, T_2, S_1, S_2 \in \mathbb{R}$, $T_1 \leq T_2$, $S_1 \leq S_2$. For realizations $(Y, X)_i = (Y_i, X_i), i = 1, \ldots, n$ of (Y, X) a simple FOFR with unconstrained surface effect $\beta(s, t)$ can be defined as

$$Y_i(t) = \beta_0(t) + \int_{\mathcal{S}} X_i(s)\beta_1(s, t)\, ds + \varepsilon_i(t), i = 1, \ldots, n, \tag{1.12}$$

where ε_i are independent and identically distributed Gaussian white noise processes, i.e., stochastic processes with zero mean and constant variance σ^2 across $\mathcal{T}$, $\beta_0(\cdot)$ is a smooth function in t and $\beta_1(\cdot, \cdot)$ a smooth two-dimensional function defined on $\mathcal{T} \times \mathcal{S}$. Using (1.12) to relate the functional covariate and the functional response, we assume a linear relationship of $X_i(s)$ and $Y_i(t)$ for all time points $s \in \mathcal{S}$ and $t \in \mathcal{T}$. Linear models such as (1.12) were first presented in Ramsay and Dalzell (1991) and adopted by many others, e.g., Yao et al. (2005). In particular, the model can be extended to include several functional covariates. In the case of J functional predictors, $\mathcal{X}$ itself can be defined as J-dimensional product space $\bigotimes_{j=1}^{J} \mathcal{X}_j$ and each $\mathcal{X}_j$ is defined on $L^2(\mathcal{S}_j, \mu)$. The work of Scheipl et al. (2015) is particularly notable in this respect, presenting a framework, which does not only allow for multiple linear functional covariates but also for (potentially time-varying) random effects, smooth functional effects, fixed scalar covariates as well as functional varying coefficients and interaction effects. This is also the case for the framework by Brockhaus et al. (2015), which additionally allows to model different characteristics of the conditional distribution of Y and which is described in more detail in the following subsection as well as in Chapter 6. Both approaches can be embedded in a comprehensive view based on GAMs, which was presented by Greven and Scheipl (2017a) describing "an impressively general framework for functional regression" (Morris, 2017) and which will be explained in more detail in the following.

Both SOFR and FOSR can be seen as special cases of the function-on-function regression. For SOFR, for which the response Y_i is assumed to be scalar, this can be achieved by defining the time domain $\mathcal{T}$ as single point interval $[t, t]$ and μ as Dirac measure (see, e.g., Brockhaus et al., 2015). The FOSR can be derived from the FOFR when no functional covariates are present. Although SOFR and FOSR represent special cases of the FOFR, the FOFR has received comparatively little attention in the past literature (Morris, 2015).

Representation and Estimation

In Greven and Scheipl (2017a) with corresponding rejoinder (Greven and Scheipl, 2017b) present and discuss FDA with particular focus on FOFR and give an extensive comparison of different available methods with their practical applicability. The authors identify five general approaches which deal with functional responses and four particularly developed software solutions, implementing fitting routines for FOFR to some extent. The presented approaches and software solutions stem from

three different representation of FOFR. The idea of Ramsay and Silverman (2005), implemented in the R package `fda` (Ramsay et al., 2016), works with pre-smoothed observations. Although this is convenient from a mathematical point of view, the practical applicability is limited due to the neglect of measurement error and due to the less general setup of this framework. The two remaining frameworks excel in their broad scope of applicability and are thus briefly described in the following.

The functional mixed model (FMM) framework, firstly introduced by Morris and Carroll (2006), is a Bayesian model setup, which represents a very flexible and historically early representation and implementation of FOSR. The approach is based on a representation of curves in their (wavelet) basis space, which reduces the regression of infinitely many observations to a finite number of basis coefficients and estimates the resulting model in the basis space using MCMC sampling. After the estimation results can be retransformed into the original space. This approach allows for a general assumption on residual errors and different random effect structures with potential extensions to parametrically specified correlation structures of the functional residuals (Zhu et al., 2016; Zhang et al., 2016). The extension of this framework to FOFR was presented by Meyer et al. (2015), which is applicable if functions are observed on a common grid.

The second approach stems from the representation of functional regression as an additive mixed model and has several roots, with some ideas dating back to Hastie and Mallows (1993) as well as Marx and Eilers (1999) and Marx and Eilers (2005), who initially proposed this concept for SOFR. The extension to functional responses by transformation to scalar data is related to varying coefficient models (Hastie and Mallows, 1993) and the idea of Reiss et al. (2010). These ideas have then been extended to FOFR by Ivanescu et al. (2015) and Scheipl et al. (2015) in the functional additive mixed model (FAMM) framework. Instead of estimating a surrogate model in the basis space as done in the FMM framework, additive model terms in the FAMM framework are represented using tensor or row-wise tensor products of two marginal basis functions for parameterizing (a) the covariate effect and (b) the (functional) form of the effect over $\mathcal{T}$. A notable extension to the FAMM framework is given by Scheipl et al. (2016), presenting generalized functional additive mixed models (GFAMM), which allows for the estimation of the conditional expectation $\mathbb{E}(Y(t)|\boldsymbol{X})$ for $Y(t)$, which is assumed to follow a distribution from the exponential family, but can be also be a random variable from a less commonly used distribution, such as the Tweedie or the Negative Binomial distribution (see Scheipl et al., 2016, Section 2 for a complete list of supported distributions). In the case of FAMM and GFAMM, estimation is done by the initial use of a Laplace-approximate marginal likelihood to estimate involved smoothing parameters, followed by the estimation of model coefficients for given smoothing parameters based on a penalized likelihood. The model framework is implemented in the R package `refund` (Goldsmith et al., 2016), which in turn uses the R package `mgcv` (Wood, 2016) as fitting engine. As presented by Greven and Scheipl (2017a), the framework can be even more generalized, including the extension to GAMLSS, a general basis representation and an alternative way of model estimation, namely boosting, which is presented in Chapter 6.

Whereas most of the initial ideas have been proposed for an unconstrained functional effect $\int_{\mathcal{S}} X(s)\beta(s,t)\,ds$, further functional predictors can be defined by softening the linear assumption,

by allowing for interaction with scalar covariates or by restricting the integration of $X(s)\beta(s,t)$ to an interval $[l(t), u(t)] \subset \mathcal{S}$. A special case, which can be derived from this restriction, is the so-called *historical effect*. Historical effects are extensively studied and generalized in Chapter 5 and are therefore described briefly in the following.

1.5.3 Historical Models

In the second part of this thesis, functional historical models will play an important role in the application of FOFR to bioelectrical signals. In contrast to previously described frameworks for FOFR, both functional response and functional covariates are therefore assumed to be observed over the same (time) interval, i.e., $\mathcal{T} = \mathcal{S}$, allowing for a meaningful association between the response and a covariate with respect to the (time) domain. The name *historical* stems from the fact, that only the history $s < t$ of covariates modeled via *historical effects* are assumed to influence the response at time point t, $s, t \in \mathcal{T}$, but not "future values". In more detail, a historical effect for a functional covariate X can be defined via

$$\int_{T_1}^{t} X(s)\beta(s,t)\, ds, \quad s,t \in \mathcal{T}$$

or more general via

$$\int_{l(t)}^{u(t)} X(s)\beta(s,t)\, ds, \quad s,t \in \mathcal{T},$$

where $l(\cdot)$ and $u(\cdot)$ define the integration limits as function of the time t and the first definition of a historical effect can be obtained by setting $l(t) = T_1$ and $u(t) = t$. FOFR models with one historical effect were introduced by Malfait and Ramsay (2003), Harezlak et al. (2007) and Gervini (2015), whereas the FAMM framework and the framework by Brockhaus et al. (2017) allow for a variety and multitude of functional and, in particular, historical effects. The FAMM approach, on the one hand, relies on the estimation via mixed models and is thus based on a well established and thoroughly studied framework but limited to estimation of models for the conditional mean of the response. On the other hand, Brockhaus et al. (2017) use the CFGD algorithm as fitting procedure, which is not accompanied with a ready to use inference toolbox, but allows for model estimation of any transformation function of the conditional response and is computational advantageous. This is due to the component-wise nature of the estimation procedure, highlighted in more detail in Section 1.3 and studied in Part II of this thesis. The used software package, which is presented in Chapter 6, additionally allows for a very modular specification of historical and other FOFR models.

References

Akaike, H. (1973). *Information Theory and an Extension of the Maximum Likelihood Principle.* Ed. B. N. Petrov and F. Csaki.

Bachoc, F., Leeb, H., and Pötscher, B. (2014). Valid confidence intervals for post-model-selection predictors. *arXiv preprint arXiv:1412.4605*.

Bachoc, F., Preinerstorfer, D., and Steinberger, L. (2016). Uniformly valid confidence intervals post-model-selection. *arXiv e-prints arXiv:1611.01043*.

Belloni, A., Chernozhukov, V., Hansen, C., and Kozbur, D. (2016). Inference in high-dimensional panel models with an application to gun control. *Journal of Business & Economic Statistics*, 34(4):590–605.

Belloni, A., Chernozhukov, V., and Kato, K. (2018). Valid post-selection inference in high-dimensional approximately sparse quantile regression models. *Journal of the American Statistical Association*, pages 1–33.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 57(1):289–300.

Benjamini, Y. and Yekutieli, D. (2005). False discovery rate - adjusted multiple confidence intervals for selected parameters. *Journal of the American Statistical Association*, 100(469):71–81.

Berk, R., Brown, L., Buja, A., Zhang, K., Zhao, L., et al. (2013). Valid post-selection inference. *The Annals of Statistics*, 41(2):802–837.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Bozdogan, H. (1987). Model selection and Akaike's Information Criterion (AIC): The general theory and its analytical extensions. *Psychometrika*, 52(3):345–370.

Brockhaus, S., Melcher, M., Leisch, F., and Greven, S. (2017). Boosting flexible functional regression models with a high number of functional historical effects. *Statistics and Computing*, 27(4):913–926.

Brockhaus, S., Rügamer, D., and Greven, S. (2017). Boosting Functional Regression Models with FDboost. *ArXiv e-prints arXiv:1705.10662*.

Brockhaus, S., Scheipl, F., Hothorn, T., and Greven, S. (2015). The functional linear array model. *Statistical Modelling*, 15(3):279–300.

Brown, L. (1967). The conditional level of Student's t test. *Annals of Mathematical Statistics*, 38:1068–1071.

Buehler, R. J. and Feddersen, A. P. (1963). Note on a conditional property of Student's t. *Annals of Mathematical Statistics*, 34:1098–1100.

Bühlmann, P. (2006). Boosting for high-dimensional linear models. *The Annals of Statistics*, pages 559–583.

Bühlmann, P. and Hothorn, T. (2007). Boosting algorithms: Regularization, prediction and model fitting (with discussion). *Statistical Science*, 22(4):477–505.

Bühlmann, P. and Yu, B. (2003). Boosting with the L_2 loss: regression and classification. *Journal of the American Statistical Association*, 98(462):324–339.

Buja, A., Mease, D., and Wyner, A. J. (2007). Comment: Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science*, 22(4):506–512.

Cawley, G. C. and Talbot, N. L. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(Jul):2079–2107.

Chen, S. and Bien, J. (2017). Valid Inference Corrected for Outlier Removal. *arXiv e-prints arXiv:1711.10635*.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM.

Chen, X., Lee, J. D., Tong, X. T., and Zhang, Y. (2016). Statistical inference for model parameters in stochastic gradient descent. *arXiv preprint arXiv:1610.08637*.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., et al. (2016). Double machine learning for treatment and causal parameters. *arXiv preprint arXiv:1608.00060*.

Chernozhukov, V., Hansen, C., and Spindler, M. (2015). Post-selection and post-regularization inference in linear models with many controls and instruments. *American Economic Review*, 105(5):486–90.

Cox, D. R. (1975). A note on data-splitting for the evaluation of significance levels. *Biometrika*, 62(2):441–444.

de Boor, C. (1972). On calculating with B-splines. *Journal of Approximation Theory*, 6(1):50 – 62.

de Boor, C. (2001). A practical guide to splines, vol. 27 of Springer Series in Applied Mathematics.

deLeeuw, J. (1992). Introduction to akaike (1973) information theory and an extension of the maximum likelihood principle. In *Breakthroughs in Statistics*, Springer Series in Statistics, pages 599–609. Springer New York.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, 32(2):407–499.

Eilers, P. H. C. and Marx, B. D. (1996). Flexible smoothing with B-splines and penalties. *Statistical Science*, 11(2):89–121.

Ewald, K. and Schneider, U. (2015). Confidence sets based on the lasso estimator. *arXiv preprint arXiv:1507.05315*.

Fahrmeir, L., Kneib, T., Lang, S., and Marx, B. D. (2013). *Regression Models, Methods and Applications*. Springer, 2nd edition.

Fithian, W., Sun, D., and Taylor, J. (2014). Optimal Inference After Model Selection. *arXiv e-prints arXiv:1410.2597*.

Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232.

Friedman, J. H., Hastie, T., and Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics*, 28(2):337–407.

Gervini, D. (2015). Dynamic retrospective regression for functional data. *Technometrics*, 57(1):26–34.

Goldsmith, J., Scheipl, F., Huang, L., Wrobel, J., Gellar, J., Harezlak, J., McLean, M. W., Swihart, B., Xiao, L., Crainiceanu, C., and Reiss, P. T. (2016). *refund: Regression with Functional Data*. R package version 0.1-16.

Greven, S. and Kneib, T. (2010). On the behaviour of marginal and conditional aic in linear mixed models. *Biometrika*, 97(4):773–789.

Greven, S. and Scheipl, F. (2017a). A general framework for functional regression modelling. *Statistical Modelling*, 17(1-2):1–35.

Greven, S. and Scheipl, F. (2017b). Rejoinder. *Statistical Modelling*, 17(1-2):100–115.

Gross, S. M., Taylor, J., and Tibshirani, R. (2015). A selective approach to internal inference. *arXiv preprint arXiv:1510.00486*.

G'Sell, M. G., Wager, S., Chouldechova, A., and Tibshirani, R. (2016). Sequential selection procedures and false discovery rate control. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(2):423–444.

Harezlak, J., Coull, B. A., Laird, N. M., Magari, S. R., and Christiani, D. C. (2007). Penalized solutions to functional regression problems. *Computational Statistics & Data Analysis*, 51(10):4911–4925.

Hastie, T. and Mallows, C. (1993). Discussion. *Technometrics*, 35(2):140–143.

Hong, L., Kuffner, T. A., and Martin, R. (2018). On overfitting and post-selection uncertainty assessments. *Biometrika*, 105(1):221–224.

Hothorn, T., Buehlmann, P., Kneib, T., Schmid, M., and Hofner, B. (2017). *mboost: Model-Based Boosting*. R package version 2.8-1.

Hothorn, T. et al. (2014). Boosting–an unusual yet attractive optimiser. *Methods of Information in Medicine*, 53(6):417–418.

Ioannidis, J. P. A. (2005). Why most published research findings are false. *PLOS Medicine*, 2(8).

Ivanescu, A. E., Staicu, A.-M., Scheipl, F., and Greven, S. (2015). Penalized function-on-function regression. *Computational Statistics*, 30(2):539–568.

Javanmard, A. and Montanari, A. (2014). Confidence intervals and hypothesis testing for high-dimensional regression. *The Journal of Machine Learning Research*, 15(1):2869–2909.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Lee, J. D., Sun, D. L., Sun, Y., and Taylor, J. E. (2016). Exact post-selection inference, with application to the lasso. *The Annals of Statistics*, 44(3):907–927.

Leeb, H. and Pötscher, B. M. (2003). The finite-sample distribution of post-model-selection estimators and uniform versus nonuniform approximations. *Econometric Theory*, 19(1):100–142.

Loftus, J. R. (2015). Selective inference after cross-validation. *arXiv e-prints arXiv:1511.08866*.

Loftus, J. R. and Taylor, J. E. (2015). Selective inference in regression models with groups of variables. *arXiv e-prints arXiv:1511.01478*.

Lu, S., Liu, Y., Yin, L., and Zhang, K. (2017). Confidence intervals and regions for the lasso by using stochastic variational inequality techniques in optimization. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(2):589–611.

Luo, Y. and Spindler, M. (2017). Estimation and inference of treatment effects with L_2-boosting in high-dimensional settings. *arXiv preprint arXiv:1801.00364*.

Malfait, N. and Ramsay, J. O. (2003). The historical functional linear model. *Canadian Journal of Statistics*, 31(2):115–128.

Markovic, J. and Taylor, J. (2016). Bootstrap inference after using multiple queries for model selection. *arXiv preprint arXiv:1612.07811*.

Markovic, J., Xia, L., and Taylor, J. (2017). Adaptive p-values after cross-validation. *arXiv preprint arXiv:1703.06559*.

Marx, B. D. and Eilers, P. H. (1999). Generalized Linear Regression on Sampled Signals and Curves: A P-Spline Approach. *Technometrics*, 41(1):1–13.

Marx, B. D. and Eilers, P. H. (2005). Multidimensional penalized signal regression. *Technometrics*, 47(1):13–22.

Mayr, A., Hofner, B., Waldmann, E., Hepp, T., Meyer, S., and Gefeller, O. (2017). An update on statistical boosting in biomedicine. *Computational and Mathematical Methods in Medicine*, 2017.

McCullagh, P. (1984). Generalized linear models. *European Journal of Operational Research*, 16(3):285–292.

Meir, A. and Drton, M. (2017). Tractable post-selection maximum likelihood inference for the lasso. *arXiv preprint arXiv:1705.09417*.

Meyer, M. J., Coull, B. A., Versace, F., Cinciripini, P., and Morris, J. S. (2015). Bayesian function-on-function regression for multilevel functional data. *Biometrics*, 71(3):563–574.

Morris, J. S. (2015). Functional regression. *Annual Review of Statistics and Its Application*, 2(1):321–359.

Morris, J. S. (2017). Comparison and contrast of two general functional regression modelling frameworks. *Statistical Modelling*, 17(1-2):59–85.

Morris, J. S. and Carroll, R. J. (2006). Wavelet-based functional mixed models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(2):179–199.

Olshen, R. A. (1973). The conditional level of the F-test. *Journal of the American Statistical Association*, 68:692–698.

Panigrahi, S., Markovic, J., and Taylor, J. (2017). An MCMC free approach to post-selective inference. *arXiv preprint arXiv:1703.06154*.

Panigrahi, S. and Taylor, J. (2017). Sampling from a pseudo selective posterior using a primal-dual approach. *arXiv preprint arXiv:1703.06176*.

Panigrahi, S., Taylor, J., and Weinstein, A. (2016). Bayesian post-selection inference in the linear model. *arXiv preprint arXiv:1605.08824*.

Pawitan, Y. (2001). *In All Likelihood: Statistical Modelling and Inference Using Likelihood*. Oxford science publications. OUP Oxford.

Pötscher, B. M. (1991). Effects of model selection on inference. *Econometric Theory*, 7(2):163–185.

Ramsay, J. O. and Dalzell, C. (1991). Some tools for functional data analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, pages 539–572.

Ramsay, J. O. and Silverman, B. W. (2005). *Functional Data Analysis*. Springer-Verlag, New York.

Ramsay, J. O., Wickham, H., Graves, S., and Hooker, G. (2016). *fda: Functional Data Analysis*. R package version 2.4.5.

Reiss, P. T., Huang, L., and Mennes, M. (2010). Fast function-on-scalar regression with penalized basis expansions. *The international journal of biostatistics*, 6(1).

Ridgeway, G. (1999). The state of boosting. *Computing Science and Statistics*, pages 172–181.

Rigby, R. A. and Stasinopoulos, D. M. (2005). Generalized additive models for location, scale and shape. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 54(3):507–554.

Rügamer, D., Brockhaus, S., Gentsch, K., Scherer, K., and Greven, S. (2018). Boosting factor-specific functional historical models for the detection of synchronization in bioelectrical signals. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 67(3):621–642.

Rügamer, D. and Greven, S. (2018a). Selective inference after likelihood- or test-based model selection in linear models. *Statistics & Probability Letters*, 140:7 – 12.

Rügamer, D. and Greven, S. (2018b). Valid inference for L_2-boosting. *arXiv e-prints arXiv:1805.01852*.

Ruppert, D., Wand, M. P., and Carroll, R. J. (2003). *Semiparametric regression*. Cambridge series in statistical and probabilistic mathematics. Cambridge University Press, Cambridge and New York.

Säfken, B., Rügamer, D., Kneib, T., and Greven, S. (2018). Conditional Model Selection in Mixed-Effects Models with cAIC4. *arXiv e-prints arXiv:1803.05664*.

Schapire, R. E. (1990). The strength of weak learnability. *Machine learning*, 5(2):197–227.

Scheipl, F., Gertheiss, J., and Greven, S. (2016). Generalized functional additive mixed models. *Electronic Journal of Statistics*, 10(1):1455–1492.

Scheipl, F., Staicu, A.-M., and Greven, S. (2015). Functional additive mixed models. *Journal of Computational and Graphical Statistics*, 24(2):477–501.

Schmid, M. and Hothorn, T. (2008). Boosting additive models using component-wise p-splines. *Computational Statistics & Data Analysis*, 53(2):298 – 311.

Schoenberg, I. J. (1946a). Contributions to the problem of approximation of equidistant data by analytic functions: Part a. – on the problem of smoothing or graduation. a first class of analytic approximation formulae. *Quarterly of Applied Mathematics*, 4(1):45–99.

Schoenberg, I. J. (1946b). Contributions to the problem of approximation of equidistant data by analytic functions: Part b. – on the problem of osculatory interpolation. a second class of analytic approximation formulae. *Quarterly of Applied Mathematics*, 4(2):112–141.

Sen, P. K. (1979). Asymptotic properties of maximum likelihood estimators based on conditional specification. *The Annals of Statistics*, 7(5):1019–1033.

Sur, P., Chen, Y., and Candès, E. J. (2017). The likelihood ratio test in high-dimensional logistic regression is asymptotically a rescaled chi-square. *arXiv preprint arXiv:1706.01191*.

Tian, X., Bi, N., and Taylor, J. (2016). MAGIC: a general, powerful and tractable method for selective inference. *arXiv preprint arXiv:1607.02630*.

Tian, X. and Taylor, J. (2017). Asymptotics of selective inference. *Scandinavian Journal of Statistics*, 44(2):480–499.

Tian, X. and Taylor, J. (2018). Selective inference with a randomized response. *The Annals of Statistics*, 46(2):679–710.

Tian Harris, X., Panigrahi, S., Markovic, J., Bi, N., and Taylor, J. (2016). Selective sampling after solving a convex problem. *arXiv e-prints arXiv:1609.05609*.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 58(1):267–288.

Tibshirani, R. J., Rinaldo, A., Tibshirani, R., and Wasserman, L. (2015). Uniform Asymptotic Inference and the Bootstrap After Model Selection. *arXiv e-prints arXiv:1506.06266*.

Tibshirani, R. J., Taylor, J., Lockhart, R., and Tibshirani, R. (2016). Exact post-selection inference for sequential regression procedures. *Journal of the American Statistical Association*, 111(514):600–620.

Umezu, Y. and Takeuchi, I. (2017). Selective Inference for Change Point Detection in Multi-dimensional Sequences. *arXiv e-prints arXiv:1706.00514*.

Vaida, F. and Blanchard, S. (2005). Conditional Akaike information for mixed-effects models. *Biometrika*, 92(2):351–370.

van de Geer, S., Bühlmann, P., Ritov, Y., and Dezeure, R. (2014). On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics*, 42(3):1166–1202.

Wang, J.-L., Chiou, J.-M., and Müller, H.-G. (2016). Functional data analysis. *Annual Review of Statistics and Its Application*, 3(1):257–295.

Wood, S. (2016). mgcv: mixed GAM computation vehicle with GCV/AIC/REML smoothness estimation. R package version 1.8-15.

Wood, S. N. (2017). *Generalized additive models: an introduction with R*. CRC press, 2nd edition.

Yamada, M., Umezu, Y., Fukumizu, K., and Takeuchi, I. (2018). Post selection inference with kernels. In Storkey, A. and Perez-Cruz, F., editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 152–160, Playa Blanca, Lanzarote, Canary Islands. PMLR.

Yang, F., Barber, R. F., Jain, P., and Lafferty, J. (2016). Selective inference for group-sparse linear models. In *Advances in Neural Information Processing Systems*, pages 2469–2477.

Yao, F., Müller, H.-G., and Wang, J.-L. (2005). Functional linear regression analysis for longitudinal data. *The Annals of Statistics*, 33(6):2873–2903.

Yekutieli, D. (2012). Adjusted bayesian inference for selected parameters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(3):515–541.

Zhang, C.-H. and Zhang, S. S. (2014). Confidence intervals for low dimensional parameters in high dimensional linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(1):217–242.

Zhang, L., Baladandayuthapani, V., Zhu, H., Baggerly, K. A., Majewski, T., Czerniak, B. A., and Morris, J. S. (2016). Functional car models for large spatially correlated functional datasets. *Journal of the American Statistical Association*, 111(514):772–786.

Zhao, Q., Small, D. S., and Ertefaie, A. (2017). Selective inference for effect modification via the lasso. *arXiv preprint arXiv:1705.08020*.

Zhu, H., Versace, F., Cinciripini, P., and Morris, J. S. (2016). Robust functional mixed models for spatially correlated functional regression, with application to event-related potentials for nicotine-addicted individuals. *Under revision*.

Part I

Model Selection and Valid Inference after Model Selection

Chapter 2

Conditional Model Selection in Mixed-Effects Models with cAIC4

Chapter 2 gives an introduction to model selection with particular focus on the Akaike Information Criterion (AIC) as well as an extensions of the AIC to the class of mixed and additive regression models, which is referred to as conditional AIC. Efficient computation of the conditional AIC for certain classes of mixed models is presented and its implementation in the R package `cAIC4` is described.

Contributing article:

Säfken, B., Rügamer, D., Kneib, T., and Greven, S. (2018). Conditional Model Selection in Mixed-Effects Models with cAIC4. *arXiv e-prints arXiv:1803.05664*.

Author contributions:

The manuscript was written by Benjamin Säfken and David Rügamer. Contributions were divided by sections. The first draft of the general introduction as well as the introduction to the `cAIC4` package were authored by Benjamin Säfken. The first draft for the sections to mixed models, the conditional AIC, the stepwise conditional variable selection as well as the conclusion were authored by David Rügamer. Benjamin Säfken additionally provided the draft of the conditional AIC for non-Gaussian responses. Sonja Greven and Thomas Kneib added valuable input, suggested several noteable modifications and proofread the manuscript.

Conditional Model Selection in Mixed-Effects Models with cAIC4

Benjamin Säfken
Georg-August Universität Göttingen

David Rügamer
Ludwig-Maximilans-Universität München

Thomas Kneib
Georg-August Universität Göttingen

Sonja Greven
Ludwig-Maximilans-Universität München

Abstract

Model selection in mixed models based on the conditional distribution is appropriate for many practical applications and has been a focus of recent statistical research. In this paper we introduce the R-package **cAIC4** that allows for the computation of the conditional Akaike Information Criterion (cAIC). Computation of the conditional AIC needs to take into account the uncertainty of the random effects variance and is therefore not straightforward. We introduce a fast and stable implementation for the calculation of the cAIC for linear mixed models estimated with **lme4** and additive mixed models estimated with **gamm4** . Furthermore, **cAIC4** offers a stepwise function that allows for a fully automated stepwise selection scheme for mixed models based on the conditional AIC. Examples of many possible applications are presented to illustrate the practical impact and easy handling of the package.

Keywords: conditional AIC, **lme4**, Mixed Effects Models, Penalized Splines.

1. Introduction

The linear mixed model is a flexible and broadly applicable statistical model. It is naturally used for analysing longitudinal or clustered data. Furthermore, any regularized regression model incorporating a quadratic penalty can be written in terms of a mixed model. This incorporates smoothing spline models, spatial models and more general additive models (Wood 2017). Thus efficient and reliable estimation of such models is of major interest for applied statisticians. The package **lme4** for the statistical computing software R (R Core Team 2016) offers such an exceptionally fast and generic implementation for mixed models (see Bates, Mächler, Bolker, and Walker 2015). The package has a modular framework allowing for the profile restricted maximum likelihood (REML) criterion as a function of the model parameters to be optimized using any constrained optimization function in R and uses rapid techniques for solving penalized least squares problems based on sparse matrix methods.
The fact that mixed models are widely used popular statistical tools make model selection an indispensable necessity. Consequently research regarding model choice, variable selection and hypothesis testing in mixed models has flourished in recent years.
Hypothesis testing on random effects is well established, although for likelihood ratio tests

boundary issues arise (Crainiceanu and Ruppert 2004; Greven, Crainiceanu, Küchenhoff, and Peters 2008; Wood 2013). In model selection for mixed models using the Akaike information criterion (AIC Akaike 1973), Vaida and Blanchard (2005) suggest to use different criteria depending on the focus of the underlying research question. They make a distinction between questions with a focus on the population and on clusters, respectively. For the latter, they introduce a conditional AIC accounting for the shrinkage in the random effects. Based on this conditional AIC, Liang, Wu, and Zou (2008) propose a criterion that corrects for the estimation uncertainty of the random effects variance parameters based on a numerical approximation. Greven and Kneib (2010) show that ignoring this estimation uncertainty induces a bias and derive an analytical representation for the conditional AIC.

For certain generalized mixed models, analytical representations of the conditional AIC exist, for instance for Poisson responses (see Lian 2012). Although there is no general unbiased criterion in analytical form for all exponential family distributions as argued in Säfken, Kneib, van Waveren, and Greven (2014), bootstrap-based methods can often be applied as we will show for those in presented in Efron (2004). An asymptotic criterion for a wider class of distributions is described in Wood, Pya, and Säfken (2016).

In this paper, we describe an add-on package to **lme4** that facilitates model selection based on the conditional AIC and illustrates it with several examples. For the conditional AIC proposed by Greven and Kneib (2010) for linear mxied models, the computation of the criterion is not as simple as it is for other common AIC criteria. This article focuses on techniques for fast and stable computation of the conditional AIC in mixed models estimated with **lme4**, as they are implemented in the R-package **cAIC4**. The amount of possible models increases substantially with the R-package **gamm4** (see Wood and Scheipl 2016) allowing for the estimation of a wide class of models with quadratic penalty such as spline smoothing and additive models. The presented conditional AIC applies to any of these models.

In addition to translating the findings of Greven and Kneib (2010) to the model formulations used in Bates *et al.* (2015), we present the implementation of conditional AICs proposed for non-Gaussian settings in Säfken *et al.* (2014) and as we propose based on Efron (2004). With these results, a new scheme for stepwise conditional variable selection in mixed models is introduced. This allows for fully automatic choice of fixed and random effects based on the optimal conditional AIC. All methods are accompanied by examples, mainly taken from **lme4**, see Bates *et al.* (2015). The rest of this paper is structured as follows:

In Section 2 the mixed model formulations are introduced based on one example with random intercepts and random slopes and a second example on penalised spline smoothing. The conditional AIC for Gaussian, Poisson and Bernoulli responses is introduced in Section 3. Section 4 gives a hands-on introduction to **cAIC4** with specific examples for the `sleepstudy` and the `grouseticks` data from **lme4**. The new scheme for stepwise conditional variable selection in mixed models is presented in Section 5 and applied to the `Pastes` data set. After the conclusion in Section 6, part A of the appendix describes how **cAIC4** automatically deals with boundary issues. Furthermore the underlying code for the rapid computation of the conditional AIC is presented in part B of the appendix.

2. The mixed model

In a linear mixed model, the conditional distribution of the response $\boldsymbol{y}$ given the random

effects $\boldsymbol{u}$ has the form

$$\boldsymbol{y}|\boldsymbol{u} \sim \mathcal{N}\left(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u}, \sigma^2\boldsymbol{I}_n\right), \tag{1}$$

where $\boldsymbol{y}$ is the n-dimensional vector of responses, $\boldsymbol{\beta}$ is the p-dimensional vector of fixed effects and $\boldsymbol{u}$ is the q-dimensional vector of random effects. The matrices $\boldsymbol{X}$ and $\boldsymbol{Z}$ are the $(n \times p)$ and $(n \times q)$ design matrices for fixed and random effects, respectively, and σ^2 refers to the variance of the error terms.
The unconditional distribution of the random effects $\boldsymbol{u}$ is assumed to be a multivariate Gaussian with mean $\mathbf{0}$ and positive semidefinite $(q \times q)$ covariance matrix $\boldsymbol{D}_{\boldsymbol{\theta}}$, i.e.,

$$\boldsymbol{u} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{D}_{\boldsymbol{\theta}}\right).$$

The symmetric covariance matrix $\boldsymbol{D}_{\boldsymbol{\theta}}$ depends on the covariance parameters $\boldsymbol{\theta}$ and may be decomposed as

$$\boldsymbol{D}_{\boldsymbol{\theta}} = \sigma^2\boldsymbol{\Lambda}_{\boldsymbol{\theta}}\boldsymbol{\Lambda}_{\boldsymbol{\theta}}^t, \tag{2}$$

with the lower triangular covariance factor $\boldsymbol{\Lambda}_{\boldsymbol{\theta}}$ and the variance parameter σ^2 of the conditional response distribution. In analogy to generalized linear models, the generalized linear mixed model extends the distributional assumption in (1) to a distribution $\mathcal{F}$ from the exponential family,

$$\boldsymbol{y}|\boldsymbol{u} \sim \mathcal{F}(\boldsymbol{\mu}, \phi)$$

where ϕ is a scale parameter and the mean has the form

$$\boldsymbol{\mu} = \mathbb{E}(\boldsymbol{y}|\boldsymbol{u}) = h\left(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u}\right), \tag{3}$$

with h being the response function applied componentwise and natural parameter $\boldsymbol{\eta} = h^{-1}\left(\boldsymbol{\mu}\right)$. As the hereinafter presented results are limited to the Poisson and binomial distributions we can assume $\phi = 1$. The symmetric covariance matrix in (2) then is the same as for Gaussian responses except that σ^2 is omitted, i.e., $\boldsymbol{D}_{\boldsymbol{\theta}} = \boldsymbol{\Lambda}_{\boldsymbol{\theta}}\boldsymbol{\Lambda}_{\boldsymbol{\theta}}^t$.

The given conditional formulations of (generalized) linear mixed models imply marginal models, which can (conceptually) be obtained by integrating the random effects out of the joint distribution of $\boldsymbol{y}$ and $\boldsymbol{u}$, i.e.,

$$f(\boldsymbol{y}) = \int f(\boldsymbol{y} \mid \boldsymbol{u}) f(\boldsymbol{u}) d\boldsymbol{u}.$$

However, there is typically no closed form solution for this integral. While the marginal model formulation is usually used for estimation, an analytic representation of $f(\boldsymbol{y})$ is only available for the linear mixed model (1). The marginal distribution $f(\boldsymbol{y})$ for Gaussian responses $\boldsymbol{y}$ is given by

$$\boldsymbol{y} \sim \mathcal{N}\left(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2\left(\boldsymbol{I}_n + \boldsymbol{Z}\boldsymbol{\Lambda}_{\boldsymbol{\theta}}\boldsymbol{\Lambda}_{\boldsymbol{\theta}}^t\boldsymbol{Z}^t\right)\right).$$

Further extensions of linear mixed models can be obtained by, for example, relaxing the assumption $\text{Cov}(\boldsymbol{y}|\boldsymbol{u}) = \sigma^2 \boldsymbol{I}_n$.

Example I: Random intercepts and random slopes

Some special cases of mixed models are commonly used in applications, including the random intercept model and the random slope model. In the random intercept model, the responses differ in an individual- or cluster-specific intercept for m individuals or clusters. In this case the individual-specific intercept is modeled as random effect $\boldsymbol{u} = (u_{1,1}, u_{1,2}, \ldots, u_{1,m})$, yielding the (generalized) linear mixed model

$$\mathbb{E}(y_{ij}|u_{1,i}) = h(\boldsymbol{x}_{ij}\boldsymbol{\beta} + u_{1,i}), \quad u_{1,i} \overset{iid}{\sim} \mathcal{N}(0, \tau_0^2 \boldsymbol{I}_m)$$

for the j-th observation from an individual or cluster i.

Whereas for the random intercept model all covariates modeled with fixed effects are assumed to have the same influence on the response variable across individuals, the random slope model is suitable when an independent variable x_s is assumed to have an individual-specific effect on the dependent variable. The random intercept model is extended to

$$\mathbb{E}(y_{ij}|\boldsymbol{u}_i) = h(\boldsymbol{x}_{ij}\boldsymbol{\beta} + u_{1,i} + x_{s,ij}u_{2,i}),$$

where $u_{2,i}$ is the individual-specific slope, which can be regarded as the deviation from the population slope β_s corresponding to the s-th covariate $x_{s,ij}$ in $\boldsymbol{x}_{ij}$. In most cases, there is no reason to suppose $u_{1,i}$ and $u_{2,i}$ to be uncorrelated and the distributional assumption thus is

$$\begin{pmatrix} u_{1,i} \\ u_{2,i} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_1^2 & \tau_{12} \\ \tau_{21} & \tau_2^2 \end{pmatrix} \right). \tag{4}$$

Example II: Penalised spline smoothing

In addition to many possibilities to extend these simple random effect models, linear mixed models can also be utilized to fit semi-parametric regression models (see, e.g., Ruppert, Wand, and Carroll 2003). For univariate smoothing, consider the model

$$\mathbb{E}(y_i) = f(x_i), \tag{5}$$

for $i = 1, \ldots, n$, where $f(\cdot)$ is a deterministic function of the covariate x_i, which shall be approximated using splines. For illustrative purposes, we consider the truncated polynomial basis representation

$$f(x) = \sum_{j=0}^{g} \beta_j x^j + \sum_{j=1}^{k} u_j (x - \kappa_j)_+^g, \tag{6}$$

in the following, where $\kappa_1 < \ldots < \kappa_k$ are $k \in \mathbb{N}$ knots, partitioning the domain of x, $g \in \mathbb{N}$ and

$$(z)_+^g = z^g \cdot I(z > 0) = \begin{cases} z^g & \text{if } z > 0 \\ 0 & \text{if } z \leq 0 \end{cases}. \tag{7}$$

As the truncated part $u_j(x - \kappa_j)_+^g$ is non-zero for $x > \kappa_j$, u_j can be seen as a gradient change of the two consecutive function segments defined on $(\kappa_{j-1}, \kappa_j]$ and $(\kappa_j, \kappa_{j+1}]$. In order to estimate $\beta_j, j = 0, \ldots, g$ and $u_j, j = 1, \ldots, k$, the method of ordinary least squares (OLS) could in principle be applied. In most cases, however, this yields a rather rough estimate of f for suitably large k as the gradient changes of functions segments have a large impact. Therefore estimation methods for linear mixed models can be utilized in order to obtain a smooth function. Representing the untruncated polynomial part in (6) as the fixed effects and $\sum_{j=1}^k u_j(x - \kappa_j)_+^g$ as the random effects part, the well known shrinkage effect of mixed models is transferred to the estimation of the u_js, shrinking the changes in the gradient of the fitted polynomials. The random effects assumption corresponds to a quadratic penalty on the u_j, with the smoothing parameter estimated from the data.

This approach also works analogously for various other basis functions including the frequently used B-spline basis (see, e.g., Fahrmeir, Kneib, Lang, and Marx 2013). Moreover, a rich variety of models that can be represented as reduced rank basis smoothers with quadratic penalties allow for this kind of representation. The estimation via **lme4** can be employed by the use of **gamm4**. For an overview of possible model components see Wood (2017). An example is also given in Section 5.

3. The conditional AIC

The Akaike Information Criterion

Originally proposed by Hirotogu Akaike (Akaike 1973) as An Information Criterion (AIC), the AIC was one of the first model selection approaches to attract special attention among users of statistics. In some way, the AIC extends the maximum likelihood paradigm by making available a framework, in which both parameter estimation and model selection can be accomplished. The principle idea of the AIC can be traced back to the Kullback-Leibler distance (KLD Kullback and Leibler 1951), which can be used to measure the distance between a true (but normally unknown) density $g(\boldsymbol{y})$ and a parametric model $f(\boldsymbol{y} \mid \boldsymbol{\nu})$. The unknown parameters $\boldsymbol{\nu}$ are commonly estimated by their maximum likelihood estimator $\hat{\boldsymbol{\nu}}(\boldsymbol{y})$. As minimizing the expected Kullback-Leibler distance is equivalent to minimizing the so called Akaike Information

$$\text{AI} = -2\, \mathbb{E}_{g(\boldsymbol{y})} \mathbb{E}_{g(\tilde{\boldsymbol{y}})} \log f(\tilde{\boldsymbol{y}} \mid \hat{\boldsymbol{\nu}}(\boldsymbol{y})), \tag{8}$$

with $\tilde{\boldsymbol{y}}$ a set of independent new observations from g, minus twice the maximized log-likelihood $\log f(\boldsymbol{y} \mid \hat{\boldsymbol{\nu}}(\boldsymbol{y}))$ as a natural measure of goodness-of-fit is an obvious estimator of the AI. However, this approach induces a bias as the maximized log-likelihood only depends on $\boldsymbol{y}$

whereas (8) is defined as a predictive measure of two independent replications $\tilde{\boldsymbol{y}}$ and $\boldsymbol{y}$ from the same underlying distribution. Therefore the bias correction is defined by

$$\mathrm{BC} = 2\left(\mathbb{E}_{g(\boldsymbol{y})} \log f(\boldsymbol{y} \mid \hat{\boldsymbol{\nu}}(\boldsymbol{y})) - \mathbb{E}_{g(\boldsymbol{y})}\mathbb{E}_{g(\tilde{\boldsymbol{y}})} \log f(\tilde{\boldsymbol{y}} \mid \hat{\boldsymbol{\nu}}(\boldsymbol{y}))\right). \tag{9}$$

Akaike derived the bias correction, which under certain regularity conditions can be estimated asymptotically by two times the dimension of $\boldsymbol{\nu}$. This yields the well-known AI estimator

$$\mathrm{AIC}(\boldsymbol{y}) = -2 \log f(\boldsymbol{y} \mid \hat{\boldsymbol{\nu}}(\boldsymbol{y})) + 2 \dim(\boldsymbol{\nu}).$$

Hence, as the statistical model $f(\cdot|\boldsymbol{\nu})$ with the smallest AI aims at finding the model which is closest to the true model, the AIC can be seen as a relative measure of goodness-of-fit for different models of one model class. Notice that the bias correction is equivalent to the (effective) degrees of freedom and the covariance penalty, see Efron (2004).

The marginal and the conditional perspective on the AIC

Adopting this principle for the class of mixed models to select amongst different random effects is not straightforward. First of all, the question arises on the basis of which likelihood to define this AIC. For the class of mixed models, two common criteria exist, namely the marginal AIC (mAIC) based on the marginal log-likelihood and the conditional AIC (cAIC) based on the conditional log-likelihood. The justification of both approaches therefore corresponds to the purpose of the marginal and the conditional mixed model perspective, respectively. Depending on the question of interest, the intention of both perspectives differs, as for example described in Vaida and Blanchard (2005) or Greven and Kneib (2010).

The marginal perspective of mixed models is suitable when the main interest is to model fixed population effects with a reasonable correlation structure. The conditional perspective, by contrast, can be used to make statements based on the fit of the predicted random effects. In longitudinal studies, for example, the latter point of view seems to be more appropriate if the focus is on subject- or cluster-specific random effects. Another crucial difference in both approaches lies in the model's use for prediction. On the one hand, the marginal model seems to be more plausible if the outcome for new observations comes from new individuals or clusters, i.e., observations having new random effects. The conditional model on the other hand is recommended if predictions are based on the same individuals or clusters, thereby predicting on the basis of already modeled random effects.

The corresponding AI criteria have closely related intentions. The conditional AIC estimates the optimism of the estimated log-likelihood for a new data set $\tilde{\boldsymbol{y}}$ by leaving the random effects unchanged. This can be understood as a predictive measure based on a new data set originating from the same clusters or individuals as $\boldsymbol{y}$. On the contrary, the marginal approach evaluates the log-likelihood using a new predictive data set $\tilde{\boldsymbol{y}}$, which is not necessarily associated with the cluster(s) or individual(s) of $\boldsymbol{y}$.

In particular for the use of mixed models in penalized spline smoothing, the cAIC usually represents a more plausible choice. As demonstrated in Example II of Section 2, the representation of penalized spline smoothing via mixed models divides certain parts of the spline basis into fixed and random effects. Using the marginal perspective in Example II, predictions would therefore be based only on the polynomial coefficients of f. If the fitted non-linear

function is believed to represent a general relationship of x and y, predictions as well as the predictive measure in terms of the Akaike Information, however, make more sense if the truncated parts of the basis are also taken into account.

Vaida and Blanchard (2005) proposed the cAIC, an estimator of the conditional Akaike Information

$$\text{cAI} = -2\,\mathbb{E}_{g(\boldsymbol{y},\boldsymbol{u})}\mathbb{E}_{g(\tilde{\boldsymbol{y}}|\boldsymbol{u})}\log f(\tilde{\boldsymbol{y}} \mid \hat{\boldsymbol{\nu}}(\boldsymbol{y}), \hat{\boldsymbol{u}}(\boldsymbol{y})) \tag{10}$$

as an alternative to the mAIC, where $\boldsymbol{\nu}$ includes the fixed effects and covariance parameters $\boldsymbol{\theta}$. The cAIC may be more appropriate when the AIC is used for the selection of random effects. In addition, Greven and Kneib (2010) investigated the difference of both criteria from a mathematical point of view. Since the mAIC is intended for the use in settings where the observations are independent and the k-dimensional parameter space $\boldsymbol{\mathcal{V}}_k$ can be transformed to $\mathbb{R}^k$, the corresponding bias correction $2\dim(\boldsymbol{\nu})$ is biased for mixed models for which these conditions do not apply. In particular, Greven and Kneib showed that the mAIC leads to a preference for the selection of smaller models without random effects.

Conditional AIC for Gaussian responses

Depending on the distribution of $\boldsymbol{y}$, different bias corrections of the maximized conditional log-likelihood exist to obtain the cAIC. For the Gaussian case, Liang *et al.* (2008) derive a corrected version of the initially proposed cAIC by Vaida and Blanchard (2005) for known error variance, taking into account the estimation of the covariance parameters $\boldsymbol{\theta}$:

$$\text{cAIC}(\boldsymbol{y}) = -2\log f(\boldsymbol{y} \mid \hat{\boldsymbol{\nu}}(\boldsymbol{y}), \hat{\boldsymbol{u}}(\boldsymbol{y})) + 2\,\text{tr}\left(\frac{\partial \hat{\boldsymbol{y}}}{\partial \boldsymbol{y}}\right). \tag{11}$$

Evaluating the bias correction $\text{BC} = 2\,\text{tr}(\frac{\partial \hat{\boldsymbol{y}}}{\partial \boldsymbol{y}})$ in expression (11) via numerical approximation, or a similar formula for unknown error variance, is however computationally expensive. Greven and Kneib (2010) develop an analytic version of the corrected cAIC making the calculation of the corrected cAIC feasible. We adapt their efficient implementation originally written for `lme`-objects (returned by the `nlme` package) and reimplement their algorithm for `lmerMod`-objects (returned by `lme4`). A more detailed description on the calculation of several terms in the proposed formula of Greven and Kneib (2010) is given in Appendix B. Furthermore, a partition of the parameter space is needed in order to account for potential parameters on the boundary of the parameter space, as presented in Theorem 3 in Greven and Kneib (2010). This process can be very unwieldy. Therefore, a fully automated correction algorithm is implemented in **cAIC4** and presented in Appendix A.

Conditional AIC for Poisson responses

As for the Gaussian case, note that for the Poisson and the binomial distribution the bias

correction (9) can be rewritten as twice the sum of the covariances between $\widehat{\eta}_i$ and y_i,

$$BC = 2\sum_{i=1}^{n} \mathbb{E}\left(\widehat{\eta}_i \left(y_i - \mu_i\right)\right), \tag{12}$$

with true but unobserved mean μ_i and the estimator of the natural parameter $\widehat{\boldsymbol{\eta}}$ depending on $\boldsymbol{y}$. For the Poisson distribution an analytic reformulation of the bias correction term (12) has to be utilized to make it analytically accessible as in Säfken *et al.* (2014). Using results from Hudson (1978) and an identity due to Chen (1975), the bias correction (12) for Poisson distributed responses can be reformulated to

$$BC = 2\sum_{i=1}^{n} \mathbb{E}\left(y_i \left(\log \hat{\mu}_i(\boldsymbol{y}) - \log \hat{\mu}_i(\boldsymbol{y}_{-i}, y_i - 1)\right)\right), \tag{13}$$

for observations $i = 1, \ldots, n$ and mean estimator $\hat{\mu}_i$. The i-th component of $\boldsymbol{y}$ in $(\boldsymbol{y}_{-i}, y_i - 1)$ is substituted by $y_i - 1$ along with the convention $y_i \log \hat{\mu}_i(\boldsymbol{y}_{-i}, y_i - 1) = 0$ if $y_i = 0$. The computational implementation of the cAIC in this case requires $n - d$ model fits, where d corresponds to the number of Poisson responses being equal to zero (see Section 4 for details). The resulting cAIC was first derived by Lian (2012).

Conditional AIC for Bernoulli responses

For binary responses there is no analytical representation for the bias correction (12), see Säfken *et al.* (2014). Nevertheless a bootstrap estimate for the bias correction can be based on Efron (2004). The bias correction is equal to the sum over the covariances of the estimators of the natural parameter $\widehat{\eta}_i$ and the data y_i. To estimate this quantity, we could in principle draw a parametric bootstrap sample $\boldsymbol{z}_i$ of size B for the i-th data point - keeping all other observations fixed at their observed values - to estimate the i-th component $\mathbb{E}\left(\widehat{\eta}_i \left(y_i - \mu_i\right)\right)$ of the bias correction (12) for binary responses by

$$\frac{1}{B-1}\sum_{j=1}^{B} \hat{\eta}_i(z_{ij}) \left(z_{ij} - \overline{\boldsymbol{z}}_{i\cdot}\right) = \frac{B_1}{B-1}\hat{\eta}_i(1) \left(1 - \overline{\boldsymbol{z}}_{i\cdot}\right) + \frac{B_0}{B-1}\hat{\eta}_i(0) \left(-\overline{\boldsymbol{z}}_{i\cdot}\right),$$

where B_0 is the number of zeros in the bootstrap sample, B_1 is the number of ones in the bootstrap sample, $\hat{\eta}_i(1) = \log\left(\frac{\hat{\mu}_i(1)}{1-\hat{\mu}_i(1)}\right)$ is the estimated logit (the natural parameter) with $z_{ij} = 1$, $\hat{\eta}_i(0) = \log\left(\frac{\hat{\mu}_i(0)}{1-\hat{\mu}_i(0)}\right)$ is the estimated logit with $z_{ij} = 0$ and $\overline{\boldsymbol{z}}_{i\cdot}$ is the mean of the bootstrap sample $\boldsymbol{z}_i$. Letting the number of bootstrap samples tend to infinity, i.e., $B \to \infty$ the mean of the bootstrap sample $\overline{\boldsymbol{z}}_{i\cdot} = \frac{1}{B}\sum_{j=1}^{B} z_{ij} = B_1/B$ (as well as $B_1/(B-1)$) converges to the estimate from the data, which corresponds to the true mean in the bootstrap, $\hat{\mu}_i$ and therefore

$$\begin{aligned}\frac{B_1}{B-1}\hat{\eta}_i(1) \left(1 - \overline{\boldsymbol{z}}_{i\cdot}\right) - \frac{B_0}{B-1}\hat{\eta}_i(0) \left(\overline{\boldsymbol{z}}_{i\cdot}\right) &\to \hat{\mu}_i\hat{\eta}_i(1) \left(1 - \hat{\mu}_i\right) - \left(1 - \hat{\mu}_i\right)\hat{\eta}_i(0) \left(\hat{\mu}_i\right) \\ &= \hat{\mu}_i \left(1 - \hat{\mu}_i\right) \left(\hat{\eta}_i(1) - \hat{\eta}_i(0)\right) \text{ for } B \to \infty.\end{aligned}$$

Since the bootstrap estimates are optimal if the number of bootstrap samples B tends to infinity, this estimator can be seen as the optimal bootstrap estimator. The resulting estimator of the bias correction

$$\widehat{BC} = 2\sum_{i=1}^{n} \hat{\mu}_i \left(1 - \hat{\mu}_i\right) \left(\hat{\eta}_i(1) - \hat{\eta}_i(0)\right) \tag{14}$$

, which we use in the following, avoids a full bootstrap but requires n model refits.

4. Introduction to cAIC4

Example for linear mixed models

An example that is often used in connection with the R-package **lme4** is the `sleepstudy` data from a study on the daytime performance changes of the reaction time during chronic sleep restriction, see Belenky, Wesensten, Thorne, Thomas, Sing, Redmond, Russo, and Balkin (2003). Eighteen volunteers were only allowed to spend three hours of their daily time in bed for one week. The speed (mean and fastest 10% of responses) and lapses (reaction times greater than 500 ms) on a psychomotor vigilance task where measured several times. The averages of the reaction times are saved as response variable `Reaction` in the data set. Each volunteer has an identifier `Subject`. Additionally the number of days of sleep restriction at each measurement is listed in the covariate `Days`.

An example of how the `sleepstudy` data looks can be derived by the first 13 of the 180 measurements it contains:

```
R> sleepstudy[1:13,]
```

```
   Reaction Days Subject
1  249.5600    0     308
2  258.7047    1     308
3  250.8006    2     308
4  321.4398    3     308
5  356.8519    4     308
6  414.6901    5     308
7  382.2038    6     308
8  290.1486    7     308
9  430.5853    8     308
10 466.3535    9     308
11 222.7339    0     309
12 205.2658    1     309
13 202.9778    2     309
```

Further insight into the data can be gained by a lattice plot, as presented in Bates *et al.* (2015). The average reaction times of each volunteer are plotted against the days of sleep restriction with the corresponding linear regression line. Such a plot can be found in Figure 1.

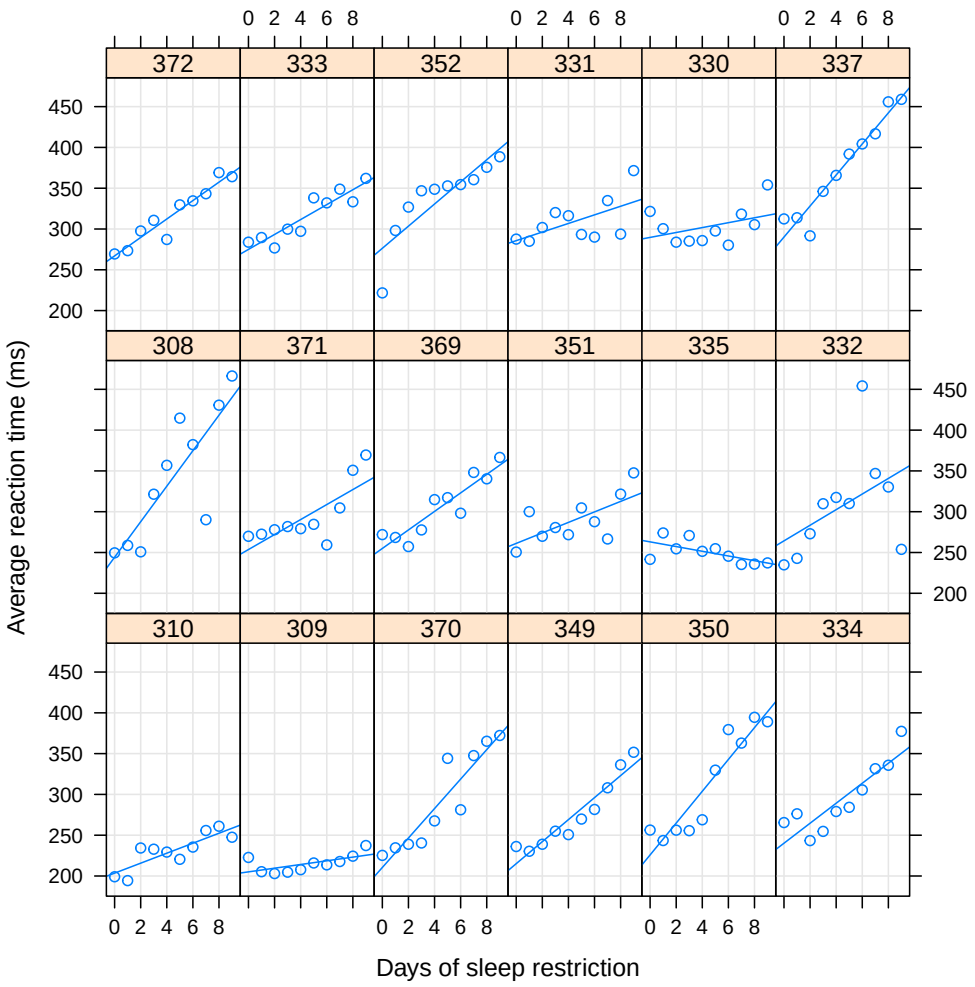

Figure 1: Lattice plot of the sleepstudy data. For each volunteer there is one panel. The identification number of each volunteer is in the heading of the panels. In the panels the reaction time is plotted against the days of sleep restriction and a regression line is added for each volunteer/panel.

The conditional AIC can be used to find the model that best predicts future observations, assuming that future observations share the same random effects as the ones used for the model fitting. In case of this data set, using the cAIC for model choice corresponds to finding the model that best predicts future reaction times of the volunteers that took part in the study.
After looking at the lattice plot, a first model that could be applied is a model with a random intercept and a random slope for `Days` within each volunteer (`Subject`):

$$y_{ij} = \beta_0 + \beta_1 \cdot \text{day}_{ij} + u_{j0} + u_{j1} \cdot \text{day}_{ij} + \epsilon_{ij} \tag{15}$$

for $i = 1, \ldots, 18$ and $j = 1, \ldots 10$, with

$$\begin{pmatrix} u_{j0} \\ u_{j1} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_1^2 & \tau_{12}^2 \\ \tau_{12}^2 & \tau_2^2 \end{pmatrix} \right).$$

In the preceding notation $\tau_1^2 = \theta_1$, $\tau_2^2 = \theta_2$ and $\tau_{12}^2 = \theta_3$. That τ_{12}^2 is not necessarily zero indicates, that the random intercept and the random slope are allowed to be correlated.

```
R> (m1 <- lmer(Reaction ~ 1 + Days + (1 + Days|Subject), sleepstudy))

Linear mixed model fit by REML ['lmerMod']
Formula: Reaction ~ 1 + Days + (1 + Days | Subject)
   Data: sleepstudy
REML criterion at convergence: 1743.628
Random effects:
 Groups   Name        Std.Dev. Corr
 Subject  (Intercept) 24.740
          Days         5.922   0.07
 Residual             25.592
Number of obs: 180, groups:  Subject, 18
Fixed Effects:
(Intercept)         Days
     251.41        10.47
```

The output shows that the within-subject correlation between the random intercepts u_{j0} and the random slopes u_{j1} is low, being estimated as 0.07. Hence there seems to be no evidence that the initial reaction time of the volunteers has systematic impact on the pace of increasing reaction time following the sleep restriction.
Consequently a suitable model might be one in which the correlation structure between both is omitted. The model for the response therefore stays the same as in (15), but the random effects covariance structure is predefined as

$$\begin{pmatrix} u_{j0} \\ u_{j1} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_0^2 & 0 \\ 0 & \tau_1^2 \end{pmatrix} \right).$$

Such a model without within-subject correlation is called by

```
R> (m2 <- lmer(Reaction ~ 1 + Days + (1|Subject) + (0 + Days|Subject),
+ sleepstudy))

Linear mixed model fit by REML ['lmerMod']
Formula: Reaction ~ 1 + Days + (1 | Subject) + (0 + Days | Subject)
   Data: sleepstudy
REML criterion at convergence: 1743.669
Random effects:
 Groups    Name        Std.Dev.
 Subject   (Intercept) 25.051
 Subject.1 Days         5.988
 Residual              25.565
Number of obs: 180, groups:  Subject, 18
Fixed Effects:
(Intercept)         Days
     251.41        10.47
```

Notice that the estimates of standard deviations of the random effects do not differ much between the first and the second model. To decide which model is more appropriate in terms of subject specific prediction the conditional AIC can be used. Calling the `cAIC`-function from the **cAIC4**-package gives the output:

```
R> cAIC(m1)

$loglikelihood
[1] -824.507

$df
[1] 31.30192

$reducedModel
NULL

$new
[1] FALSE

$caic
[1] 1711.618
```

The conditional log-likelihood and the corrected degrees of freedom, i.e., the bias correction, are the first two elements of the resulting list. The third element is called `reducedModel` and is the model without the random effects covariance parameters that were estimated to lie on the boundary of the parameter space, see Appendix A and Greven and Kneib (2010), and `NULL` if there were none on the boundary. The fourth element says if such a new model was fitted because of the boundary issue, which was not the case here. The last element is the conditional AIC as proposed in Greven and Kneib (2010).
The cAIC of the second model `m2` is:

```
R> cAIC(m2)$caic

[1] 1710.426
```

From a conditional perspective, the second model is thus preferred to the first one. This confirms the assertion that the within-subject correlation can be omitted in the model.
There are several further possible models for these data. For instance the random slope could be excluded from the model. In this model the pace of increasing reaction time does not systematically vary between the volunteers. This model is estimated by

```
R> m3 <- lmer(Reaction ~ 1 + Days + (1|Subject), sleepstudy)
```

The conditional AIC of this model is

```
R> cAIC(m3)$caic

[1] 1767.118
```

This is by far larger than the cAIC for the two preceding models. The lattice plot in Figure 1 already indicated that there is strong evidence of subject-specific (random) slopes. This is also reflected by the cAIC.
The conditional AIC is also appropriate for choosing between a simple null model without any random effects and a complex model incorporating random effects, as has been noticed by Greven and Kneib (2010). Thus it is possible to compare the cAIC of the three previous mixed models with the standard AIC for a linear model, here including three parameters (intercept, linear effect for `Days` and error variance)

```
R> -2 * logLik(lm(Reaction ~ 1 + Days, sleepstudy), REML = TRUE)[1] + 2 * 3

[1] 1899.664
```

In this case, however, the mixed model structure is evident, reflected by the large AIC for the linear model.

Example for generalized linear mixed models

The **cAIC4**-package additionally offers a conditional AIC for conditionally Poisson distributed responses and an approximate conditional AIC for binary data. The Poisson cAIC uses the bias correction (13) and the bias correction term for the binary data is (14).
Making use of the fast `refit()` function of the **lme4**-package, both cAICs can be computed moderately fast, since $n - d$ and n model refits are required, respectively, with n being the number of observations and d the number of responses that are zero for the Poisson responses.
In the following, the cAIC for Poisson response is computed for the `grouseticks` data set from the **lme4**-package as an illustration.
The `grouseticks` data set was originally published in Elston, Moss, Boulinier, Arrowsmith, and Lambin (2001). It contains information about the aggregation of parasites, so-called

Variable	Description
INDEX	identifier of the chick
TICKS	the number of ticks sampled
BROOD	the brood number
HEIGHT	height above sea level in meters
YEAR	the year as 95, 96 or 97
LOCATION	the geographic location code

Table 1: The variables and response of the grouseticks data set.

sheep ticks, on red grouse chicks. The variables in the data set are given in Table 1. Every chick, identified by INDEX, is of a certain BROOD and every BROOD, in turn, corresponds to a specific YEAR.

The number of ticks is the response variable. Following the authors in a first model the expected number of ticks λ_l with INDEX (l) is modelled depending on the year and the height as fixed effects and for each of the grouping variables BROOD (i), INDEX (j) and LOCATION (k) a random intercept is incorporated. The full model is

$$\log\left(\mathbb{E}\left(\texttt{TICKS}_l\right)\right) = \log\left(\lambda_l\right) = \beta_0 + \beta_1 \cdot \texttt{YEAR}_l + \beta_2 \cdot \texttt{HEIGHT}_l + u_{1,i} + u_{2,j} + u_{3,k} \tag{16}$$

with random effects distribution

$$\begin{pmatrix} u_{1,i} \\ u_{2,j} \\ u_{3,k} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_1^2 & 0 & 0 \\ 0 & \tau_2^2 & 0 \\ 0 & 0 & \tau_3^2 \end{pmatrix}\right).$$

Before fitting the model the covariates HEIGHT and YEAR are centred for numerical reasons and stored in the data set grouseticks_cen.

```
R> formula <- TICKS ~ YEAR + HEIGHT + (1|BROOD) + (1|INDEX) + (1|LOCATION)
R> p1  <- glmer(formula, family = "poisson", data = grouseticks_cen)
```

A summary of the estimated model is given below. Notice that the reported AIC in the automated summary of **lme4** is not appropriate for conditional model selection.

```
Generalized linear mixed model fit by maximum likelihood
(Laplace Approximation) ['glmerMod']
 Family: poisson  ( log )
Formula: TICKS ~ YEAR + HEIGHT + (1 | BROOD) + (1 | INDEX) + (1 | LOCATION)
   Data: grouseticks_cen

     AIC      BIC   logLik deviance df.resid
  1845.5   1869.5   -916.7   1833.5      397

Scaled residuals:
    Min      1Q  Median      3Q     Max
```

```
-1.6507 -0.5609 -0.1348  0.2895  1.8518

Random effects:
 Groups   Name        Variance  Std.Dev.
 INDEX    (Intercept) 2.979e-01 5.458e-01
 BROOD    (Intercept) 1.466e+00 1.211e+00
 LOCATION (Intercept) 5.411e-10 2.326e-05
Number of obs: 403, groups:  INDEX, 403; BROOD, 118; LOCATION, 63

Fixed effects:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.472353   0.134712   3.506 0.000454 ***
YEAR        -0.480261   0.166128  -2.891 0.003841 **
HEIGHT      -0.025715   0.003772  -6.817 9.32e-12 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 .

Correlation of Fixed Effects:
       (Intr) YEAR
YEAR   0.089
HEIGHT 0.096  0.061
```

The conditional log-likelihood and the degrees of freedom for the conditional AIC with conditionally Poisson distributed responses as in (13) for model (16) are obtained by the call of the `cAIC`-function:

```
R> set.seed(42)
R> cAIC(p1)

$loglikelihood
[1] -572.0133

$df
[1] 205.5786

$reducedModel
NULL

$new
[1] FALSE

$caic
[1] 1555.184
```

The output is the same as for Gaussian linear mixed models. It becomes apparent that there is a substantial difference between the conditional and the marginal AIC: In the output of the model the marginal AIC is reported to be 1845.48. Note that the marginal AIC is biased,

see Greven and Kneib (2010), and based on a different likelihood .
In the full model, the standard deviations of the random effects are rather low. It thus may be possible to exclude one of the grouping variables from the model, only maintaining two random effects. There are three possible models with one of the random effects terms excluded.
If the random intercept associated with `LOCATION` is excluded the model is

```
R> formel <- TICKS ~ YEAR + HEIGHT + (1|BROOD) + (1|INDEX)
R> p2  <- glmer(formel, family = "poisson", data = grouseticks_cen)
R> cAIC(p2)$caic

[1] 1555.214
```

The conditional AIC is almost the same as for the full model. It may thus make sense to choose the reduced model and for the prediction of the number of ticks not to make use of the random intercept associated with the `LOCATION` grouping.
Another possible model can be obtained by omitting the random intercepts for the `INDEX` grouping structure instead of those associated with `LOCATION`. This would make the model considerably simpler, since each chick has an `INDEX` and hence a random intercept is estimated for each observation in order to deal with overdispersion in the data.

```
R> formel <- TICKS ~ YEAR + HEIGHT + (1|BROOD) + (1|LOCATION)
R> p3  <- glmer(formel, family = "poisson", data = grouseticks_cen)
R> cAIC(p3)$caic

[1] 1842.205
```

The large cAIC in comparison with the two preceding models documents that the subject-specific random intercept for each observation should be included.
The final model for the comparison omits random intercepts associated with the `BROOD` grouping. This is equivalent to setting the associated random intercepts variance to zero, i.e., $\tau_2^2 = 0$.

```
R> formel <- TICKS ~ YEAR + HEIGHT + (1|INDEX) + (1|LOCATION)
R> p4  <- glmer(formel, family = "poisson", data = grouseticks_cen)
R> cAIC(p4)$caic

[1] 1594.424
```

The cAIC is higher than the cAICs for the full model and the model without the `LOCATION` grouping structure. Consequently either the full model or the model without the `LOCATION` grouping structure is favoured by the cAIC. The authors favour the latter.

5. A scheme for stepwise conditional variable selection

Now having the possibility to compare different (generalized) linear mixed models via the conditional AIC, we introduce a model selection procedure in this section, searching the

space of possible model candidates in a stepwise manner. Inspired by commonly used `step`-functions as for example given by the `stepAIC` function in the **MASS**-package (Venables and Ripley 2002), our `stepcAIC`-function provides an automatic model selection applicable to all models of the class `merMod` (produced by `[g]lmer`) or objects resulting from a **gamm4**-call.

For example, consider the sleepstudy model

```
R> fm1 <- lmer(Reaction ~ Days + (Days | Subject), sleepstudy)
```

which implicitly fits the random effects structure `(1 + Days | Subject)` (correlated random intercept and slope). In order to perform a data-driven search for the best model, a backward step procedure needs to fit and evaluate the following three nested models (uncorrelated random intercept and slope, only random slope, only random intercept).

```
R> fm1a <- lmer(Reaction ~ Days + (1 | Subject) + (0 + Days | Subject),
+   sleepstudy)
R> fm1b <- lmer(Reaction ~ Days + (0 + Days | Subject), sleepstudy)
R> fm1c <- lmer(Reaction ~ Days + (1 | Subject), sleepstudy)
```

Choosing the model `fm1a` in the first step, further model comparisons may be performed by for example reducing the model once again or adding another random effect. For this purpose, the `stepcAIC`-function provides the argument `direction`, having the options `backward`, `forward` and `both`. Whereas the `backward`- and `forward`-direction procedures fit and evaluate all nested or extended models step-by-step, the `both`-direction procedure alternates between forward- and backward-steps as long as any of both steps lead to an improvement in the cAIC. During model modifications in each step, the function allows to search through different types of model classes.

For fixed effects selection, the step procedure furthermore can be used to successively extend or reduce the model in order to check whether a fixed effect has a constant, linear or non-linear impact. For example, we specify a generalized additive mixed model (GAMM) as follows (cf. Gu and Wahba 1991)

$$y_{ij} = \beta_0 + x_{1,i,j}\beta_1 + f(x_{3,i,j}) + b_i + \varepsilon_{ij}, \quad i = 1, \ldots, 20, j = 1, \ldots, J_i,$$

with metric variables x_1 and x_3 in the `guWahbaData` supplied in the **cAIC4** package with continuous covariates x_0, x_1, x_2 and x_3. .

The corresponding model fit in R using **gamm4** is given by

```
R> set.seed(42)
R> guWahbaData$fac <- fac <- as.factor(sample(1:20, 400, replace =TRUE))
R> guWahbaData$y <- guWahbaData$y + model.matrix(~ fac - 1) %*% rnorm(20) * 0.5
R> br <- gamm4(y ~ x1 + s(x3, bs = "ps"), data = guWahbaData, random = ~ (1|fac))
```

resulting in the following non-linear estimate of $f(x_{3,i,j})$ (Figure 2).

Applying the backward stepwise procedure to the model `br` via

```
R> stepcAIC(br, trace = TRUE, direction = "backward", data = guWahbaData)
```

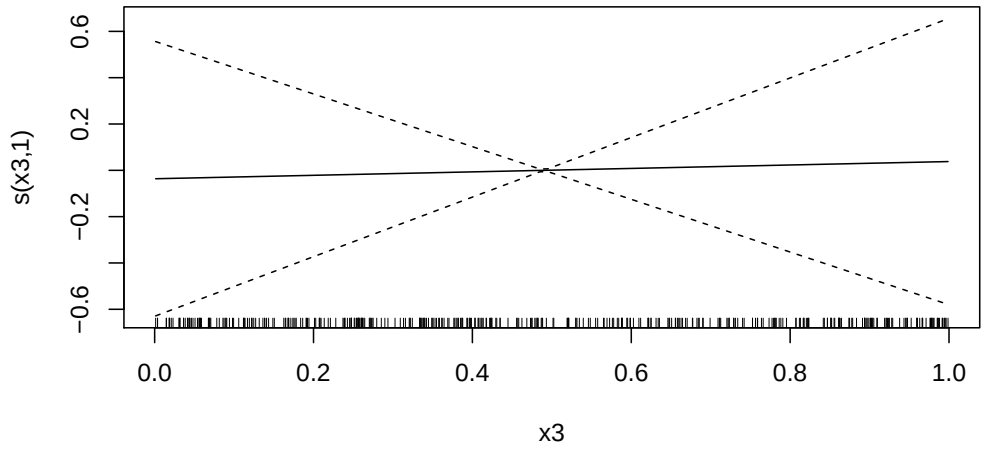

Figure 2: Plot of non-linear effect estimate for covariate x_3.

the procedure stops after one step with a warning, saying that the model contains zero variance components and the corresponding terms must be removed manually. This is due to the fact that the `stepcAIC` function can not reduce non-linear effects such as $f(x_{3,i,j})$ automatically, as the type of additive effect depends on the specification of the `s`-term and its arguments. Modifying the term manually, a GLMM is fitted and passed to the `stepcAIC` function.

```
R> br0 <- gamm4(y ~ x1 + x3, data = guWahbaData, random = ~ (1|fac))
R> stepcAIC(br0, trace = TRUE, direction = "backward", data = guWahbaData)
```

In the next steps `stepcAIC` removes `x3` completely from the model and also checks whether a GLM with no random effects at all might be the best possible model, hence having searched for the smallest cAIC in three different model classes in the end.

Whereas the backward procedure has straightforward mechanism and does not need any further mandatory arguments as shown in the previous example, the `stepcAIC`-function provides several optional and obligatory arguments for the `forward`- and `both` procedure in order to limit the possibly large number of model extensions. Regarding the required parameters, the user must specify the variables, which may be added with fixed or random effects as they are referred to in the `data.frame` given by the argument `data`. For the fixed effects, this is done by specifying the `fixEf` argument, which expects a character vector with the names of the covariates, e.g., `fixEf=c("x1","x2")`. Variables listed in the `fixEf`-argument are firstly included in the model as linear terms and, if the linear effect leads to an improvement of the cAIC, checked for their non-linearity by evaluating the cAIC of the corresponding model(s). Model extensions resulting from additional random effects are created in two different ways. A new model may, on the one hand, include a random intercept for a variable forming a

grouping structure (in the `sleepstudy` example for `Subject`) or, on the other hand, a random slope for a variable (`Days` in this case). These two types are specified using the arguments `groupCandidates` for grouping variables candidates or `slopeCandidates` for candidates for variables with random slope, again by referring to the variable names in `data` as string.

Further optional arguments can determine the way random effects are treated in the step procedure:

- `allowUseAcross`: logical value whether slope variables, which are already in use with a grouping variable can also be used with other grouping variables,
- `maxSlopes`: maximum number of slopes for one grouping variable.

Following the `stepAIC`-function, the `stepcAIC`-function also provides an argument for printing interim results (`trace`) and allows for the remaining terms of the initial model to be unaffected by the procedure (`keep`: list with entries `fixed` and `random`, each either `NULL` or a `formula`). In addition, the user may choose whether the cAIC is calculated for models, for which the fitting procedure in (`g`)`lmer` could not find an optimum (`calcNonOptimMod, default = FALSE`) and might choose the type of smoothing terms added in forward steps (`bsType`).

If the step-function is used for large datasets or in the presence of highly complex models the fitting procedures as well as the calculations of the cAIC can be parallelized by defining the number of cores (`numCores`) being used if more than one model has to be fitted and evaluated in any step (therefore passing the `numCores`-argument to a `mclapply`-function implemented in the `parallel`-package (R Core Team 2016)).

Due to the variety of additive model definitions in **gamm4**, the `stepcAIC` is however limited in its generic step-functionality for GAMMs. On the one hand, extensions with non-linear effects are restricted to one smooth class given by `bsType`, on the other hand, the step-procedure is not able to deal with further arguments passed in smooth terms. The latter point is a current limitation, since the default basis dimension of the smooth term (i.e., the number of knots and the order of the penalty) is essentially arbitrary.

An additional current limitation of the `stepcAIC`-function in its applications with GAMMs is the handling of zero variance components occurring during the function call. As a meaningful handling of zero variance smoothing terms would depend on the exact specification of the non-linear term, the stepwise procedure is stopped and returns the result of the previous step. After removing the zero variance term manually the user may call the step-function again.

Examples

In order to demonstrate some functionalities of the `stepcAIC`-function, various examples are given in the following using the `Pastes` data set (Davies and Goldsmith 1972), which is available in the **lme4**-package. The data set consists of 60 observations including one metric variable `strength`, which is the strength of a chemical paste product and the categorical variables `batch` (the delivery batch), the `cask` within the delivery batch and `sample`, which is an identifier from what cask in what batch the paste sample was taken.

Starting with a random effects backward selection, the model `fm3`

```
R> fm3 <- lmer(strength ~ 1 + (1|sample) + (1|batch), Pastes)
```

may be automatically reduced using

```
R> fm3_step <- stepcAIC(fm3, direction = "backward", trace = TRUE, data = Pastes)

Starting stepwise procedure...
-----------------------------------------------
-----------------------------------------------

Step 1 (backward):  cAIC=178.2809
Best model so far: ~ (1 | sample) + (1 | batch)
New Candidates:

Calculating cAIC for 2 model(s) ...

        models loglikelihood        df     caic
  ~(1 | batch)    -141.49709  9.157892 301.3100
 ~(1 | sample)     -58.95458 30.144477 178.1981

-----------------------------------------------
-----------------------------------------------

Step 2 (backward):  cAIC=178.1981
Best model so far: ~ (1 | sample)
New Candidates:

Calculating cAIC for 1 model(s) ...

 models loglikelihood df     caic
     ~1     -155.1363  2 312.2727

-----------------------------------------------
-----------------------------------------------

Best model:  ~ (1 | sample) , cAIC: 178.1981
-----------------------------------------------
```

where in a first step, the random intercept of `batch` is dropped. Afterwards, the procedure compares the cAICs of the models `lmer(strength ~ 1 + (1|sample), Pastes)` and `lm(strength ~ 1, Pastes)`, keeping the second random effect due to a smaller cAIC of the linear mixed model.

Using the step function the other way round, a forward stepwise selection can be initialized by a simple linear model

```
R> fm3_min <- lm(strength ~ 1, data = Pastes)
```

followed by a `stepcAIC`-call

```
R> fm3_min_step <- stepcAIC(fm3_min,
+   groupCandidates = c("batch", "sample"),
+   direction = "forward", trace = TRUE,
+   data = Pastes, analytic = TRUE)
```

where possible new candidates for grouping variables are specified using the `groupCandidates`-argument. Again, the random intercept model with group `sample` is finally selected.

To illustrate the use of the `stepcAIC`-function in the context of GAMM selection, two examples are generated following the **gamm4**-help page on the basis of the `guWahbaData` data set. First, the GAMM

$$y_{ij} = \beta_0 + f(x_{0,i,j}) + x_{1,i,j}\beta_1 + f(x_{2,i,j}) + b_i, \quad i = 1, \ldots, 20, j = 1, \ldots, J_i$$

is fitted to the `guWahbaData` including a nonlinear term for the covariate `x0` using a thin-plate regression spline, a P-spline (Eilers and Marx 1996) for the covariate `x2` as well as a random effect for the grouping variable `fac`.

```
R> br <- gamm4(y ~ s(x0) + x1 + s(x2, bs = "ps"),
+   data = guWahbaData, random = ~ (1|fac))
```

In order to check for linear or non-linear effects of the two other covariates `x1` and `x3`, the `stepcAIC`-function is employed.

```
R> br_step <- stepcAIC(br, fixEf = c("x1", "x3"),
+   direction = "both",
+   data = guWahbaData)
```

After changing the linear effect `x1` to a non-linear effect, i.e., `s(x1, bs = "tp")`, and therefore improving the model's cAIC in a first forward step, the function stops due to zero variance components.

The final model `br_step` to this point is thus given by `y ~ s(x0, bs = "tp") + s(x2, bs = "ps") + s(x1, bs = "tp") + (1 | fac)`. In contrast to the effect of covariate `x2` modeled as P-spline, the effects of covariates `x0` and `x1` are modeled as thin plate regression splines (Wood 2017). For `x0`, this is due to the initial model definition, as `s(x0)` is internally equal to `s(x0, bs = "tp")`, whereas for `x1`, the definition of the spline is set by the argument `bsType` of the `stepcAIC`-function. As the `bsType`-argument is not specified in the call, the default `"tp"` is used.

Finally, a demonstration of the `keep`-statement is given for the model

```
R> br2 <- gamm4(y ~ s(x0, bs = "ps") + x2, data = guWahbaData,
+   random = ~ (1|fac))
```

where the aim is to prevent the step procedure changing the linear effect of the covariate `x2`, the non-linear effect of `x0` as well as the random effect given by `~ (1|fac)`.

```
R> br2_step <- stepcAIC(br2, trace = TRUE, direction = "both",
+   fixEf = c("x1", "x3"), bsType = "cs",
+   keep = list(fixed = ~ s(x0, bs = "ps") + x2,
+   random= ~ (1|fac)), data = guWahbaData)
```

After successively adding a linear effect of `x1` to the model, neither the following backward step nor another forward step do improve the cAIC. The final model is given by `y ~ s(x0, bs = "ps") + x1 + x2` and random effect `(1|fac)`.

6. Conclusion

This paper gives a hands-on introduction to the R-package **cAIC4** allowing for model selection in mixed models based on the conditional AIC. The package and the paper offer a possibility for users from the empirical sciences to use the conditional AIC without having to worry about lengthy and complex calculations or mathematically sophisticated boundary issues of the parameter space. The applications presented in this paper go far beyond model selection for mixed models and extend to penalized spline smoothing and other structured additive regression models. Furthermore a stepwise algorithm for these models is introduced that allows for fast model selection.
Often statistical modelling is not about finding one 'true model'. In such cases it is of interest to define weighted sums of plausible models. This approach called model averaging is presented in Zhang, Zou, and Liang (2014) for weights chosen by the cAIC. We plan to implement this approach in **cAIC4**. Another future research path is to implement an appropriate version of the Bayesian information criterion (BIC) for conditional model selection.

Acknowledgements
The research by Thomas Kneib and Benajmin Säfken was supported by the RTG 1644 - Scaling Problems in Statistics and the Centre for Statistics at Georg-August-Universität Göttingen. Sonja Greven and David Rügamer acknowledge funding by Emmy Noether grant GR 3793/1-1 from the German Research Foundation.

A. Dealing with the boundary issues

A major issue in obtaining the conditional AIC in linear mixed models is to account for potential parameters of $\boldsymbol{\theta}$ on the boundary of the parameter space (see Greven and Kneib 2010). This needs to be done in order to ensure positive definiteness of the covariance matrix $\boldsymbol{D}_{\boldsymbol{\theta}}$.
The restructuring of the model in order to obtain the cAIC is done automatically by **cAIC4**. To gain insight into the restructuring, an understanding of the mixed model formulas used in **lme4** is essential. For an in depth explanation on how the formula module of **lme4** works, see Bates *et al.* (2015), Section 2.1.
Suppose we want to fit a mixed model with two grouping factors `g1` and `g2`. Within the first grouping factor `g1`, there are three continuous variables `v1`, `v2` and `v3` and within the second grouping factor there is only one variable `x`. Thus there are not only random intercepts but also random slopes that are possibly correlated within the groups. Such a model with response `y` would be called in **lme4** by

```
R> m <- lmer(y ~ (v1 + v2 + v3|g1) + (x|g2), exampledata)
```

In mixed models fitted with **lme4**, the random effects covariance matrix $\boldsymbol{D}_{\boldsymbol{\theta}}$ always has block-diagonal structure. For instance in the example from above the Cholesky factorized blocks of the estimated $\boldsymbol{D}_{\boldsymbol{\theta}}$ associated with each random effects term are

```
R> getME(m, "ST")

$g2
            [,1] [,2]
[1,]  1.18830353  NaN
[2,] -0.01488359    0

$g1
              [,1]        [,2] [,3] [,4]
[1,]  1.0184626697  0.00000000  NaN  NaN
[2,] -0.1438761295  0.05495809  NaN  NaN
[3,] -0.0007341796  0.19904339    0  NaN
[4,] -0.0883652598 -1.36463267 -Inf    0
```

If any of the diagonal elements of the blocks are zero the corresponding random effects terms are deleted from the formula. In **lme4** this is done conveniently by the component names list

```
R> m@cnms

$g2
[1] "(Intercept)" "x"

$g1
[1] "(Intercept)" "v1"          "v2"          "v3"
```

Thus a new model formula can be obtained by designing a new components names list:

```
R> varBlockMatrices <- getME(m, "ST")
R> cnms <- m@cnms
R> for(i in 1:length(varBlockMatrices)){
+   cnms[[i]] <- cnms[[i]][which(diag(varBlockMatrices[[i]]) != 0)]
+ }
R> cnms

$g2
[1] "(Intercept)"

$g1
[1] "(Intercept)" "v1"
```

The `cnms2formula` function from the **cAIC4**-package forms a new formula from the `cnms` object above. Hence the new formula can be computed by

```
R> rhs  <- cAIC4:::cnms2formula(cnms)
R> lhs  <- formula(m)[[2]]
R> reformulate(rhs, lhs)

y ~ (1 | g2) + (1 + v1 | g1)
```

This code is called from the `deleteZeroComponents` function in the **cAIC4**-package. This function automatically deletes all zero components from the model. The `deleteZeroComponents` function is called recursively, so the new model is checked again for zero components. In the example above only the random intercepts are non-zero. Hence the formula of the reduced model from which the conditional AIC is calculated is

```
R> formula(cAIC4:::deleteZeroComponents(m))

y ~ (1 | g2) + (1 | g1)
```

With the new model the conditional AIC is computed. If there are no random effect terms left in the formula, a linear model and the conventional AIC is returned. The `deleteZeroComponents` function additionally accounts for several special cases that may occur.
Notice however that in case of using smoothing terms from **gamm4** no automated check for boundary issues can be applied and zero components have to be manually deleted.

B. Computational matters

Gaussian responses

The corrected conditional AIC proposed in Greven and Kneib (2010) accounts for the uncertainty induced by the estimation of the random effects covariance parameters $\boldsymbol{\theta}$. In order to adapt the findings of Greven and Kneib (2010), a number of quantities from the `lmer` model fit need to be extracted and transformed. In the following these computations are presented. They are designed to minimize the computational burden and maximize the numerical stability. Parts of the calculations needed, for instance the Hessian of the ML/REML criterion, can also be found in Bates *et al.* (2015). Notice however, that **lme4** does not explicitly calculate these quantities but uses derivative free optimizers for the profile likelihoods.
A core ingredient of mixed models is the covariance matrix of the marginal responses $\boldsymbol{y}$. The inverse of the scaled covariance matrix $\boldsymbol{V}_0$ will be used in the following calculations:

$$\boldsymbol{V} = \operatorname{cov}(\boldsymbol{y}) = \sigma^2 \left(\boldsymbol{I}_n + \boldsymbol{Z}\boldsymbol{\Lambda}_{\boldsymbol{\theta}}\boldsymbol{\Lambda}_{\boldsymbol{\theta}}^t\boldsymbol{Z}^t \right) = \sigma^2 \boldsymbol{V}_0.$$

Large parts of the computational methods in **lme4** rely on a sparse Cholesky factor that satisfies

$$\boldsymbol{L}_{\boldsymbol{\theta}}\boldsymbol{L}_{\boldsymbol{\theta}}^t = \boldsymbol{\Lambda}_{\boldsymbol{\theta}}^t\boldsymbol{Z}^t\boldsymbol{Z}\boldsymbol{\Lambda}_{\boldsymbol{\theta}} + \boldsymbol{I}_q. \tag{17}$$

From this equation and keeping in mind that $\boldsymbol{I} - \boldsymbol{V}_0^{-1} = \boldsymbol{Z}\left(\boldsymbol{Z}^t\boldsymbol{Z} + \left(\boldsymbol{\Lambda}_{\boldsymbol{\theta}}^t\right)^{-1}\boldsymbol{\Lambda}_{\boldsymbol{\theta}}^{-1}\right)^{-1}\boldsymbol{Z}^t$, see Greven and Kneib (2010), it follows that

$$\begin{aligned}\boldsymbol{\Lambda}_{\boldsymbol{\theta}}\left(\boldsymbol{L}_{\boldsymbol{\theta}}^t\right)^{-1}\boldsymbol{L}_{\boldsymbol{\theta}}^{-1}\boldsymbol{\Lambda}_{\boldsymbol{\theta}}^t &= \left(\boldsymbol{Z}^t\boldsymbol{Z} + \left(\boldsymbol{\Lambda}_{\boldsymbol{\theta}}^t\right)^{-1}\boldsymbol{\Lambda}_{\boldsymbol{\theta}}^{-1}\right)^{-1} \\ \Rightarrow \qquad \boldsymbol{I} - \boldsymbol{V}_0^{-1} &= \left(\boldsymbol{L}_{\boldsymbol{\theta}}^{-1}\boldsymbol{\Lambda}_{\boldsymbol{\theta}}^t\boldsymbol{Z}^t\right)^t\left(\boldsymbol{L}_{\boldsymbol{\theta}}^{-1}\boldsymbol{\Lambda}_{\boldsymbol{\theta}}^t\boldsymbol{Z}^t\right).\end{aligned}$$

Hence the inverse of the scaled variance matrix $\boldsymbol{V}_0^{-1}$ can be efficiently computed with the help of the R-package **Matrix** (see Bates and Maechler 2017) that provides methods specifically for sparse matrices:

```
R> Lambdat <- getME(m, "Lambdat")
R> V0inv <- diag(rep(1, n)) -
+   crossprod(solve(getME(m, "L"), system = "L") %*%
+   solve(getME(m, "L"), Lambdat, system = "P") %*% t(Z))
```

Notice that `solve(getME(m, "L"), Lambdat, system = "P")` accounts for a fill-reducing permutation matrix $\boldsymbol{P}$ associated (and stored) with $\boldsymbol{L}_{\boldsymbol{\theta}}$, see Bates *et al.* (2015), and is thus equivalent to

```
R> P %*% Lambdat
```

Another quantity needed for the calculation of the corrected degrees of freedom in the conditional AIC are the derivatives of the scaled covariance matrix of the responses $\boldsymbol{V}_0$ with respect to the j-th element of the parameter vector $\boldsymbol{\theta}$:

$$\boldsymbol{W}_j = \frac{\partial}{\partial\theta_j}\boldsymbol{V}_0 = \boldsymbol{Z}\boldsymbol{D}_{\boldsymbol{\theta}}^{(j)}\boldsymbol{Z}^t,$$

where the derivative of the scaled covariance matrix of the random effects with respect to the j-th variance parameter is defined by

$$\boldsymbol{D}_{\boldsymbol{\theta}}^{(j)} = \frac{1}{\sigma^2}\frac{\partial}{\partial\theta_j}\boldsymbol{D}_{\boldsymbol{\theta}}.$$

Notice that $\boldsymbol{D}_{\boldsymbol{\theta}} = [d_{st}]_{s,t=1,\ldots,q}$ is symmetric and block-diagonal and its scaled elements are stored in $\boldsymbol{\theta}$, hence $d_{st} = d_{ts} = \theta_j\sigma^2$, for certain t, s and j. Thus the matrix $\boldsymbol{D}_{\boldsymbol{\theta}}^{(j)} = \left[d_{st}^{(j)}\right]_{s,t=1,\ldots,q}$ is sparse with

$$d_{st}^{(j)} = \begin{cases}1 & , \text{ if } d_{st} = d_{ts} = \theta_j\sigma^2 \\ 0 & , \text{ else.}\end{cases}$$

The derivative matrices $\boldsymbol{W}_j$ can be derived as follows:

```
R> Lambda <- getME(m, "Lambda")
R> ind    <- getME(m, "Lind")
R> len    <- rep(0, length(Lambda@x))
R>
R> for(j in 1:length(theta)) {
+   LambdaS                    <- Lambda
+   LambdaSt                   <- Lambdat
+   LambdaS@x                  <- LambdaSt@x                    <- len
+   LambdaS@x[which(ind == j)] <- LambdaSt@x[which(ind == j)]   <- 1
+   diagonal                   <- diag(LambdaS)
+   diag(LambdaS)              <- diag(LambdaSt)                <- 0
+   Dj                         <- LambdaS + LambdaSt
+   diag(Dj)                   <- diagonal
+   Wlist[[j]]                 <- Z %*% Dj %*% t(Z)
+ }
```

The following matrix is essential to derive the corrected AIC of Theorem 3 in Greven and Kneib (2010). Adapting their notation, the matrix is

$$\boldsymbol{A} = \boldsymbol{V}_0^{-1} - \boldsymbol{V}_0^{-1}\boldsymbol{X}\left(\boldsymbol{X}^t\boldsymbol{V}_0^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^t\boldsymbol{V}_0^{-1}.$$

Considering that the cross-product of the fixed effects Cholesky factor is

$$\boldsymbol{X}^t\boldsymbol{V}_0^{-1}\boldsymbol{X} = \boldsymbol{R}_{\boldsymbol{X}}^t\boldsymbol{R}_{\boldsymbol{X}},$$

the matrix $\boldsymbol{A}$ can be rewritten

$$\boldsymbol{A} = \boldsymbol{V}_0^{-1} - \left(\boldsymbol{X}\boldsymbol{R}_{\boldsymbol{X}}^{-1}\boldsymbol{V}_0^{-1}\right)\left(\boldsymbol{X}\boldsymbol{R}_{\boldsymbol{X}}^{-1}\boldsymbol{V}_0^{-1}\right)^t.$$

Accordingly the computation in R can be done as follows:

```
R> A <- V0inv - crossprod(crossprod(X %*% solve(getME(m, "RX")), V0inv))
```

With these components, the Hessian matrix

$$\boldsymbol{B} = \frac{\partial^2 \mathrm{REML}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^t} \text{ or } \boldsymbol{B} = \frac{\partial^2 \mathrm{ML}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^t}$$

and the matrix

$$\boldsymbol{G} = \frac{\partial^2 \mathrm{REML}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{y}^t} \text{ or } \boldsymbol{G} = \frac{\partial^2 \mathrm{ML}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{y}^t},$$

depending on whether the restricted or the marginal profile log-likelihood $\mathrm{REML}(\boldsymbol{\theta})$ or $\mathrm{ML}(\boldsymbol{\theta})$ is used, can be computed straightforward as in Greven and Kneib (2010). Depending on the optimization, it may not even be necessary to compute the matrix $\boldsymbol{B}$. Considering that $\boldsymbol{B}$ is the Hessian of the profile (restricted) log-likelihood, the matrix can also be taken from the model fit, although this is only a numerical approximation. If the Hessian is computed it is stored in:

```
R> B <- m@optinfo$derivs$Hessian
```

The inverse of $\boldsymbol{B}$ does not need to be calculated – instead, if $\boldsymbol{B}$ is positive definite, a Cholesky decomposition and two backward solves are sufficient:

```
R> Rchol   <- chol(B)
R> L1      <- backsolve(Rchol, G, transpose = TRUE)
R> Gammay  <- backsolve(Rchol, L1)
```

The trace of the hat matrix, the first part of the effective degrees of freedom needed for the cAIC, can also easily be computed with the help of the residual matrix $\boldsymbol{A}$

```
R> df <- n - sum(diag(A))
```

The correction needed to account for the uncertainty induced by the estimation of the variance parameters can be added for each random effects variance parameter separately by calculating

```
R> for (j in 1:length(theta)) {
+   df <- df + sum(Gammay[j,] %*% A %*% Wlist[[j]] %*% A %*% y)
+ }
```

Poisson responses

The computation of the bias correction for Poisson distributed responses is obtained differently. In a first step the non-zero responses need to be identified and a matrix with the responses in each column is created. Consider the `grouseticks` example in Section 4 with the model `p1` fitted by `glmer`.

```
R> y              <- p1@resp$y
R> ind            <- which(y != 0)
R> workingMatrix  <- matrix(rep(y, length(y)), ncol = length(y))
```

The diagonal values of the matrix are reduced by one and only those columns of the matrix with non-zero responses are kept.

```
R> diag(workingMatrix) <- diag(workingMatrix) - 1
R> workingMatrix       <- workingMatrix[, ind]
```

Now the `refit()` function can be applied to the columns of the matrix in order to obtain the estimates $\log \hat{\mu}_i(\boldsymbol{y}_{-i}, y_i - 1)$ in (13) from the reduced data.

```
R> workingEta  <- diag(apply(workingMatrix, 2, function(x)
+   refit(p1, newresp = x)@resp$eta)[ind,])
```

The computation of the bias correction is then straightforward:

```
R> sum(y[ind] * (p1@resp$eta[ind] - workingEta))

[1] 205.5785
```

and corresponds to the bias correction obtained in Section 4.

Bernoulli

The computation of an estimator of the bias correction for Bernoulli distributed responses as in Equation (14) is similar to the implementation for Poisson distributed responses above. Therefore consider any Bernoulli model `b1` fitted by the `glmer` function in **lem4**. For the calculation of the bias correction for each observed response variable the model needs to be refitted with corresponding other value, i.e., 0 for 1 and vice versa. This is done best by use of the `refit()` function from **lme4**.

```
R> muHat              <- b1@resp$mu
R> workingEta          <- numeric(length(muHat))
R> for(i in 1:length(muHat)){
+   workingData    <- b1$y
+   workingData[i] <- 1 - workingData[i]
+   workingModel   <- refit(b1, nresp = workingData)
+   workingEta[i]  <- log(workingModel@resp$mu[i] /
+         (1 - workingModel@resp$mu[i])) -
+         log(muHat[i] / (1 - muHat[i]))
+ }
```

The sign of the re-estimated logit (the natural parameter) in (14) which is stored in the vector `workingEta` needs to be taken into account, i.e., $\hat{\eta}_i(1)$ is positive and $\hat{\eta}_i(0)$ negative. With a simple sign correction

```
R> signCor <- - 2 * b1@resp$y + 1
```

the following returns the bias correction:

```
R> sum(muHat * (1 - muHat) * signCor * workingEta)
```

It should be pointed out that for the conditional AIC it is essential to use the conditional log-likelihood with the appropriate bias correction. Notice that the log-likelihood that by default is calculated by the S3-method `logLik` for class `merMod` (the class of a mixed model fitted by a `lmer` call) is the marginal log-likelihood.

References

Akaike H (1973). *Information Theory and an Extension of the Maximum Likelihood Principle.* Ed. B. N. Petrov and F. Csaki.

Bates D, Mächler M, Bolker B, Walker S (2015). "Fitting Linear Mixed-Effects Models Using lme4." *Journal of Statistical Software*, **67**(1), 1–48. doi:10.18637/jss.v067.i01.

Bates D, Maechler M (2017). *Matrix: Sparse and Dense Matrix Classes and Methods*. R package version 1.2-9, URL https://CRAN.R-project.org/package=Matrix.

Belenky G, Wesensten NJ, Thorne DR, Thomas ML, Sing HC, Redmond DP, Russo MB, Balkin TJ (2003). "Patterns of Performance Degradation and Restoration during Sleep Restriction and Subsequent Recovery: a Sleep Dose-response Study." *Journal of sleep research*, **12**(1), 1–12. ISSN 0962-1105.

Chen LHY (1975). "Poisson Approximation for Dependent Trials." *The Annals of Probability*, **3**(3), 534–545. ISSN 0091-1798. doi:10.1214/aop/1176996359.

Crainiceanu CM, Ruppert D (2004). "Likelihood Ratio Tests in Linear Mixed Models with One Variance Component." *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, **66**(1), 165–185. ISSN 1369-7412. doi:10.1111/j.1467-9868.2004.00438.x.

Davies OL, Goldsmith PLE (1972). *Statistical Methods in Research and Production*. 4 edition. Oliver and Boyd.

Efron B (2004). "The Estimation of Prediction Error." *Journal of the American Statistical Association*, **99**(467), 619–632. doi:10.1198/016214504000000692.

Eilers PHC, Marx BD (1996). "Flexible Smoothing with B-splines and Penalties." *Statistical Science*, **11**(2), 89–121. ISSN 0883-4237. doi:10.1214/ss/1038425655.

Elston DA, Moss R, Boulinier T, Arrowsmith C, Lambin X (2001). "Analysis of Aggregation, a Worked Example: Numbers of Ticks on Red Grouse Chicks." *Parasitology*, **122**(05), 563–569. ISSN 1469-8161.

Fahrmeir L, Kneib T, Lang S, Marx BD (2013). *Regression Models, Methods and Applications*. 2 edition. Springer-Verlag.

Greven S, Crainiceanu CM, Küchenhoff H, Peters A (2008). "Restricted Likelihood Ratio Testing for Zero Variance Components in Linear Mixed Models." *Journal of Computational and Graphical Statistics*, **17**(4), 870–891. doi:10.1198/106186008X386599.

Greven S, Kneib T (2010). "On the Behaviour of Marginal and Conditional AIC in Linear Mixed Models." *Biometrika*, **97**(4), 773–789. doi:10.1093/biomet/asq042.

Gu C, Wahba G (1991). "Minimizing GCV/GML Scores with Multiple Smoothing Parameters via the Newton Method." *SIAM Journal on Scientific and Statistical Computing*, **12**(2), 383–398. ISSN 0196-5204. doi:10.1137/0912021.

Hudson HM (1978). "A Natural Identity for Exponential Families with Applications in Multiparameter Estimation." *The Annals of Statistics*, **6**(3), 473–484. ISSN 0090-5364. doi:10.1214/aos/1176344194.

Kullback S, Leibler RA (1951). "On Information and Sufficiency." *Ann. Math. Statist.*, **22**(1), 79–86. URL 10.1214/aoms/1177729694.

Lian H (2012). "A Note on Conditional Akaike Information for Poisson Regression with Random Effects." *Electronic Journal of Statistics*, **6**(0), 1–9. ISSN 1935-7524. doi:10.1214/12-EJS665.

Liang H, Wu H, Zou G (2008). "A Note on Conditional AIC for Linear Mixed-effects Models." *Biometrika*, **95**(3), 773–778. doi:10.1093/biomet/asn023.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Ruppert D, Wand MP, Carroll RJ (2003). *Semiparametric Regression*. Cambridge series in statistical and probabilistic mathematics. Cambridge University Press, Cambridge and New York. ISBN 9780521785167.

Säfken B, Kneib T, van Waveren CS, Greven S (2014). "A Unifying Approach to the Estimation of the Conditional Akaike Information in Generalized Linear Mixed Models." *Electronic Journal of Statistics*, **8**(0), 201–225. ISSN 1935-7524. doi:10.1214/14-EJS881.

Vaida F, Blanchard S (2005). "Conditional Akaike Information for Mixed-effects Models." *Biometrika*, **92**(2), 351–370. doi:10.1093/biomet/92.2.351.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. Fourth edition. Springer-Verlag, New York. ISBN 0-387-95457-0, URL http://www.stats.ox.ac.uk/pub/MASS4.

Wood S, Scheipl F (2016). *gamm4: Generalized Additive Mixed Models using 'mgcv' and 'lme4'*. R package version 0.2-4, URL https://CRAN.R-project.org/package=gamm4.

Wood SN (2013). "A Simple Test for Random Effects in Regression Models." *Biometrika*. doi:10.1093/biomet/ast038.

Wood SN (2017). *Generalized Additive Models: An Introduction with R*. CRC press.

Wood SN, Pya N, Säfken B (2016). "Smoothing Parameter and Model Selection for General Smooth Models." *Journal of the American Statistical Association*, **111**(516), 1548–1563. doi:10.1080/01621459.2016.1180986.

Zhang X, Zou G, Liang H (2014). "Model averaging and weight choice in linear mixed-effects models." *Biometrika*, **101**(1), 205. doi:10.1093/biomet/ast052.

Chapter 3

Selective inference after likelihood- or test-based model selection in linear models

Chapter 3 is concerned with the non-trivial task of performing valid inference after model selection. The selective inference framework is presented, which provides valid inference statements conditional on the selection event. In this context, analytic expressions for valid inference after likelihood- and test-based model selection are derived. The validity of the proposed expressions are demonstrated using simulation studies.

Contributing article:

Rügamer, D. and Greven, S. (2018a). Selective inference after likelihood- or test-based model selection in linear models. *Statistics & Probability Letters*, 140:7 – 12.

Author contributions:

The manuscript was written by David Rügamer. Sonja Greven added valuable input and proofread the manuscript.

Supplementary material available at:

`https://www.sciencedirect.com/science/article/pii/S0167715218301640`

Statistics and Probability Letters 140 (2018) 7–12

Contents lists available at ScienceDirect

Statistics and Probability Letters

journal homepage: www.elsevier.com/locate/stapro

Selective inference after likelihood- or test-based model selection in linear models

David Rügamer *, Sonja Greven

Department of Statistics, LMU Munich, Ludwigstraße 33, 80539, Munich, Germany

ARTICLE INFO

Article history:
Received 22 September 2017
Received in revised form 22 March 2018
Accepted 10 April 2018
Available online 24 April 2018

Keywords:
AIC
Likelihood-based model selection
Linear models
Selective inference
Test-based model selection

ABSTRACT

Statistical inference after model selection requires an inference framework that takes the selection into account in order to be valid. Following recent work on selective inference, we derive analytical expressions for inference after likelihood- or test-based model selection for linear models.

1. Introduction

The invalidity of standard inference after model selection has been mentioned by many authors throughout the last decades, including Buehler and Feddersen (1963) and Leeb and Pötscher (2005). Following these publications different approaches for inference in (high-dimensional) regression models after some sort of model selection have emerged over the past years. Initiated by the proposal for valid statistical inference after arbitrary selection procedures by Berk et al. (2013), many new findings and adoptions of post-selection inference (PoSI) to existing statistical methods have been published. Particularly notable is the general framework of Fithian et al. (2014) transferring the classical theory of Lehmann and Scheffé (1955) in exponential family models to *selective inference*. This post-selection inference concept is based on the conditional distribution of parameter estimators, conditional on the given selection event. Apart from general theory, several authors derive explicit representations of the space to which inference is restricted by well-known selection methods. Initially motivated by the application to the Lasso (see, e.g., Lee et al., 2016) several recent publications aim for valid selective inference in forward stepwise regression or any forward stagewise algorithms. In this context, substantial work was done by Tibshirani et al. (2016) as well as by Loftus and Taylor (2014, 2015) for linear models with known error variance σ^2. Tibshirani et al. (2016) build a framework for any sequential regression technique resulting in a limitation to the space for inference, where the limitation can be characterized by a polyhedral set. Loftus and Taylor (2014, 2015) extend the idea to a more general framework, for which the limitation of the inference space is given by quadratic inequalities, which coincides with the polyhedral approach in special cases.

Despite the popularity of the Lasso and similar selection techniques in statistical applications, likelihood-based model selection such as stepwise Akaike Information Criterion (AIC, Akaike, 1973) selection is still used in an extremely vast number of statistical applications and diverse scientific fields (see, e.g., Zhang, 2016). However, authors usually do not adjust their inference for model selection, although consequences may be grave (see, e.g., Mundry and Nunn, 2009). Selective inference

* Corresponding author.
E-mail address: david.ruegamer@stat.uni-muenchen.de (D. Rügamer).

https://doi.org/10.1016/j.spl.2018.04.010
0167-7152/

allows to adjust inference after model selection, but an explicit representation of the required conditional distribution for likelihood-based model selection or similar selection procedures has not been derived so far.

We close this gap by explicitly deriving the necessary distribution in linear models with unknown σ^2 after likelihood- or test-based model selection, which comprises (iterative) model selection based on the AIC or Bayesian Information Criterion (BIC, Schwarz, 1978), model selection via likelihood-based tests, F-tests, and p-value selection ("significance hunting", Berk et al., 2013) based on t-tests. We derive an analytical solution for inference in linear models after these model selection procedures and make available an R package for selective inference in such settings in practice (Rügamer, 2017). In addition, we provide inference for multiple and arbitrarily combined selection events, such as stepwise AIC selection followed by significance hunting. We thereby close an important gap in the application of selective inference to model selection approaches that are ubiquitous in statistical applications across all scientific areas.

Section 2 presents the theory on selective testing for linear models and explicitly derives the necessary conditional distributions for several commonly used model selection approaches. In Section 3 we present simulation results for the proposed methods and apply our method to the prostate cancer data. We summarize our concept in Section 4. Derivations of our results and visualizations of additional simulation settings can be found in the supplementary material online.

2. Selective inference in linear models

After outlining the model framework and existing theoretical foundations on selective tests for linear models in Section 2.1, we present the new results on selective tests after various particular selection techniques in Sections 2.2–2.3. We further show how to extend existing theory for the construction of conditional confidence intervals in Section 2.4 and outline tests of grouped variables in this framework in Section 2.5.

2.1. Setup and theoretical foundation

Given n independent variables $\boldsymbol{Y} = (Y_1, \dots, Y_n)^\top$ with true underlying distribution $\mathcal{N}_n(\boldsymbol{\mu}, \sigma^2\boldsymbol{I}_n)$, we consider as possible models submodels of the maximal linear model $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, $\boldsymbol{\varepsilon} \sim \mathcal{N}_n(\boldsymbol{0}, \sigma^2\boldsymbol{I}_n)$, for the given data $(\boldsymbol{y}, \boldsymbol{X})$, where $\boldsymbol{y} = (y_1, \dots, y_n)^\top$ are the observed values of $\boldsymbol{Y}$ and $\boldsymbol{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_p) \in \mathbb{R}^{n\times p}$ is a fixed design matrix. In particular, we allow the considered models to be misspecified if $\boldsymbol{\mu}$ does not lie in the column space of the design matrix, in which case the corresponding model aims at estimating the linear projection of $\boldsymbol{\mu}$ onto the column space of the design matrix. We then compare two or more linear models based on different column subsets $\boldsymbol{X}_{\mathcal{T}}$ of $\boldsymbol{X}$ by using a likelihood-based model selection criterion, as for example the AIC. For the compared subsets, we let $\mathcal{T} \in \mathcal{P}(\{1, \dots, p\}) \setminus \emptyset$ with power set function $\mathcal{P}(\cdot)$. After selection of the "best fitting" model with design matrix $\boldsymbol{X}_{\mathcal{T}^*}$ with $|\mathcal{T}^*| = p_{\mathcal{T}^*}$, we would ideally like to test the jth regression coefficient in the set of corresponding coefficients $\boldsymbol{\beta}_{\mathcal{T}^*}$, i.e.

$$H_0 : \beta_{\mathcal{T}^* j} = \theta. \tag{1}$$

However, taking into account that the true mean $\boldsymbol{\mu}$ is potentially non-linear in the selected covariates or the selection is not correct, we instead test the jth component of the projection of $\boldsymbol{\mu}$ into the linear space spanned by the selected covariates $\boldsymbol{X}_{\mathcal{T}^*}$:

$$H_0 : \tilde{\beta}_{\mathcal{T}^* j} = \boldsymbol{v}^\top\boldsymbol{\mu} := \boldsymbol{e}_j^\top(\boldsymbol{X}_{\mathcal{T}^*}{}^\top\boldsymbol{X}_{\mathcal{T}^*})^{-1}\boldsymbol{X}_{\mathcal{T}^*}{}^\top\boldsymbol{\mu} = \theta, \tag{2}$$

where $\boldsymbol{e}_j$ is the jth unit vector and $\boldsymbol{v}$ is the so-called test vector. This coincides with (1) if we select the correct model and $\boldsymbol{\mu}$ is actually linear in $\boldsymbol{X}_{\mathcal{T}^*}$. Testing the linear approximation instead of (1) is a more realistic scenario in practice and is in line with the approach of several recent publications including Berk et al. (2013).

We consider the following quadratic inequality introduced in a similar form by Loftus and Taylor (2015), on the basis of which a model is chosen:

$$\boldsymbol{Y}^\top\boldsymbol{A}\boldsymbol{Y} + c \geq 0, \tag{3}$$

before showing that several common model selection approaches lead to restrictions on $\boldsymbol{Y}$ that can be written in this form. In most practical situations $c \equiv 0$. We are interested in the null distribution of $\hat{\beta}_{\mathcal{T}^* j} = \boldsymbol{v}^\top\boldsymbol{Y}$, which we use as a test statistic to test the null hypothesis (2). Since $\boldsymbol{Y} \sim \mathcal{N}_n(\boldsymbol{\mu}, \sigma^2\boldsymbol{I}_n)$, $\boldsymbol{v}^\top\boldsymbol{Y} \sim \mathcal{N}_1(\boldsymbol{v}^\top\boldsymbol{\mu}, \sigma^2\boldsymbol{v}^\top\boldsymbol{v})$ with $\boldsymbol{v}^\top\boldsymbol{\mu} = \theta$ under H_0. After model selection of the form (3), $\boldsymbol{v}^\top\boldsymbol{Y}$ conditional on $\boldsymbol{Y}^\top\boldsymbol{A}\boldsymbol{Y} + c \geq 0$, and also conditional on $\boldsymbol{P}_{\boldsymbol{v}}^\perp\boldsymbol{Y} = \boldsymbol{P}_{\boldsymbol{v}}^\perp\boldsymbol{y}$ with $\boldsymbol{P}_{\boldsymbol{v}}^\perp\boldsymbol{y}$ the projection of $\boldsymbol{y}$ into the space orthogonal to $\boldsymbol{v}$, follows a truncated normal distribution (Loftus and Taylor, 2015) with truncation limits based on $\tau_{1/2} = \frac{1}{2}\delta^{-1}(-\zeta \pm \sqrt{\zeta^2 - 4\delta\xi})$, where $\delta = \boldsymbol{y}^\top\boldsymbol{P}_v\boldsymbol{A}\boldsymbol{P}_v\boldsymbol{y}$, $\zeta = 2\boldsymbol{y}^\top\boldsymbol{P}_v\boldsymbol{A}\boldsymbol{P}_v^\perp\boldsymbol{y}$ and $\xi = \boldsymbol{y}^\top\boldsymbol{P}_v^\perp\boldsymbol{A}\boldsymbol{P}_v^\perp\boldsymbol{y} + c$. In this case, additionally conditioning on $\boldsymbol{P}_v^\perp\boldsymbol{y}$ is necessary to derive the truncation limits of the truncated normal distribution of $\boldsymbol{v}^\top\boldsymbol{Y}$, which otherwise would be random themselves (see, e.g., Lee et al., 2016, Section 5.1). Due to the form of (3), the two solutions $\tau_1 \leq \tau_2$ imply that the distribution of our test statistic is truncated to $(-\infty, \tau_1 \cdot \boldsymbol{v}^\top\boldsymbol{y}] \cup [\tau_2 \cdot \boldsymbol{v}^\top\boldsymbol{y}, \infty)$ in the case in which δ is positive, and to $[\tau_1 \cdot \boldsymbol{v}^\top\boldsymbol{y}, \tau_2 \cdot \boldsymbol{v}^\top\boldsymbol{y}]$ if δ is negative.

2.2. Explicit derivations

We now show that several commonly used model selection approaches can be written as in (3) and explicitly derive the corresponding truncation limits to the normal distribution of the test statistic. For all derivations, please see the supplementary material. We always consider the comparison of two models 1 and 2 in which model 1 is preferred over model 2. Let $\mathcal{T}_k$, $k = 1, 2$ be the corresponding covariate subsets of the two considered models k and let $\boldsymbol{X}_k := \boldsymbol{X}_{\mathcal{T}_k}$ denote the corresponding design matrix.

Model selection based on log-likelihood comparison plus optional penalty term. We start with conventional model selection procedures that are based on a log-likelihood comparison plus optional penalty term (as, for example, used in the AIC or BIC). Let ℓ_k be the log-likelihood of model k and pen_k the penalty term for this model, which is assumed not dependent on $\boldsymbol{Y}$. For example, if p_k denotes the number of regression coefficients for model k and the unknown σ^2 is estimated, $\text{pen}_k = 2(p_k + 1)$ for the AIC and $\text{pen}_k = \log(n)(p_k + 1)$ for the BIC. Furthermore, let $\hat{\sigma}_k^2$ be the scale parameter estimator and $\hat{\boldsymbol{\mu}}_k = \boldsymbol{P}_{X_k}\boldsymbol{Y}$ the mean vector estimator of model $k = 1, 2$ with $\boldsymbol{P}_{X_k} = \boldsymbol{X}_k(\boldsymbol{X}_k^\top \boldsymbol{X}_k)^{-1}\boldsymbol{X}_k^\top$ and $\boldsymbol{X}_k \in \mathbb{R}^{n\times p_k}$. Then the model 1 is selected iff

$$\begin{aligned} & -2\ell_1(\boldsymbol{Y}) + \text{pen}_1 \leq -2\ell_2(\boldsymbol{Y}) + \text{pen}_2 \\ \Leftrightarrow \quad & \boldsymbol{Y}^\top\{(n-p_1)\exp(-\gamma/n)(\boldsymbol{I} - \boldsymbol{P}_{X_2}) - (n-p_2)(\boldsymbol{I} - \boldsymbol{P}_{X_1})\}\boldsymbol{Y} \geq 0 \end{aligned} \tag{4}$$

with $\gamma = (p_2 - p_1 + \text{pen}_1 - \text{pen}_2)$. We therefore define $\boldsymbol{A} := \{(n-p_1)\exp(-\gamma/n)(\boldsymbol{I} - \boldsymbol{P}_{X_2}) - (n-p_2)(\boldsymbol{I} - \boldsymbol{P}_{X_1})\}$ as well as $c := 0$. In the supplementary material we additionally derive the matrix $\boldsymbol{A}$ and c when treating σ^2 as known and plugging in $\hat{\sigma}_1, \hat{\sigma}_2$ as estimators, i.e. when ignoring the fact that $\hat{\sigma}_k^2$, $k = 1, 2$ are also functions of $\boldsymbol{Y}$, to show the difference.

Model selection on the basis of tests. We first consider the likelihood-ratio test (LRT). For model 1 being nested in model 2, the derivation is analogous to the AIC comparison by defining $\text{pen}_2 - \text{pen}_1 := q_{\chi^2_{1-\alpha}(p_2-p_1)}$, where $q_{\chi^2_\alpha(df)}$ is the α-quantile of the χ^2-distribution with df degrees of freedom.

The F-Test is not strictly likelihood-based, but falls into the same framework. Let $\text{RSS}_k = \|\boldsymbol{Y} - \hat{\boldsymbol{\mu}}_k\|^2$ be the residual sum of squares of model k. If we choose model 1, which is nested in model 2, and denote by $F(\phi_1, \phi_2)$ the critical value of the F-distribution with ϕ_1 and ϕ_2 degrees of freedom, then:

$$\frac{\frac{\text{RSS}_1 - \text{RSS}_2}{p_2 - p_1}}{\frac{\text{RSS}_2}{n-p_2}} \leq F(p_2 - p_1, n - p_2) \quad \Leftrightarrow \quad \boldsymbol{Y}^\top\{\boldsymbol{P}_{X_1} + \kappa(\boldsymbol{I} - \boldsymbol{P}_{X_2}) - \boldsymbol{P}_{X_2}\}\boldsymbol{Y} \geq 0, \tag{5}$$

where $\kappa = F(p_2 - p_1, n - p_2) \cdot \frac{p_2 - p_1}{n - p_2}$ and therefore $\boldsymbol{A} = \{\boldsymbol{P}_{X_1} + \kappa(\boldsymbol{I} - \boldsymbol{P}_{X_2}) - \boldsymbol{P}_{X_2}\}$ and $c = 0$. Similarly, if we select the larger model 2 for either LRT or F-test, we simply have to invert the previous inequalities and define $\boldsymbol{A}$ as the negative of the respective matrices $\boldsymbol{A}$ defined above.

"Significance hunting". As described in Berk et al. (2013), variable deselection or backward selection on the basis of the size of t-test p-values reduces to deselecting the smallest t-value among several candidates. For the comparison of two variables j^* and j and deselection of j^* in the model k, it therefore holds that

$$|t_{j^*}| := \frac{|\hat{\beta}_{k,j^*}|}{\text{se}(\hat{\beta}_{k,j^*})} = \left|\frac{\boldsymbol{v}_{j^*}^\top \boldsymbol{Y}}{\sqrt{\hat{\sigma}_k^2 \boldsymbol{v}_{j^*}^\top \boldsymbol{v}_{j^*}}}\right| \leq \left|\frac{\boldsymbol{v}_j^\top \boldsymbol{Y}}{\sqrt{\hat{\sigma}_k^2 \boldsymbol{v}_j^\top \boldsymbol{v}_j}}\right|, \tag{6}$$

where $\boldsymbol{v}_j^\top = \boldsymbol{e}_j^\top(\boldsymbol{X}_k{}^\top \boldsymbol{X}_k)^{-1}\boldsymbol{X}_k{}^\top$ and $\boldsymbol{v}_{j^*}^\top = \boldsymbol{e}_{j^*}^\top(\boldsymbol{X}_k{}^\top \boldsymbol{X}_k)^{-1}\boldsymbol{X}_k{}^\top$. Let $\boldsymbol{P}_{\boldsymbol{v}} = \boldsymbol{v}\boldsymbol{v}^\top/\sqrt{\|\boldsymbol{v}\|^2}$ for a given vector $\boldsymbol{v}$. Then (6) is equivalent to $\boldsymbol{Y}\left(\boldsymbol{P}_{\boldsymbol{v}_j} - \boldsymbol{P}_{\boldsymbol{v}_{j^*}}\right)\boldsymbol{Y} \geq 0$ and we can define $\boldsymbol{A} := \left(\boldsymbol{P}_{\boldsymbol{v}_j} - \boldsymbol{P}_{\boldsymbol{v}_{j^*}}\right)$, $c = 0$. If only variables which are not significant are dropped for the "significance hunting", the t-value of j^* additionally fulfills the condition $|t_{j^*}| \leq \mathcal{Q}_{T_{n-p_k}}(1 - \frac{\alpha}{2})$, where $\mathcal{Q}_{T_{n-p_k}}(\cdot)$ is the quantile function of the Student's t-distribution with $n - p_k$ degrees of freedom, which is evaluated with a prespecified significance level α to obtain the decision. Since this is equivalent to $\boldsymbol{Y}^\top \boldsymbol{P}_{\boldsymbol{v}_{j^*}} \boldsymbol{Y} \leq \hat{\sigma}_k^2 \cdot (\mathcal{Q}_{T_{n-p_k}}(1 - \frac{\alpha}{2}))^2$, we get $\boldsymbol{A} = \{(\mathcal{Q}_{T_{n-p_k}}(1 - \frac{\alpha}{2}))^2(n - p_k)^{-1}(\boldsymbol{I} - \boldsymbol{P}_{\boldsymbol{X}_k}) - \boldsymbol{P}_{\boldsymbol{v}_{j^*}}\}$ and $c = 0$.

2.3. Multiple selection events and p-value calculation

If there are m selection events of the kind as in Section 2.2, the final space restriction can be calculated by finding the two (or more) most restrictive values in all limiting selection steps. Since this may involve several inequalities with different directions and may result in two or more non-overlapping intervals, additional care is needed. In general, let the resulting truncated normal distribution have multiple truncations given by the ordered intervals $[a_1, b_1], \ldots, [a_z, b_z]$, $z \in \mathbb{N}$, where the case of no finite lower or upper truncation is given by $a_1 = -\infty$ or $b_z = \infty$ with intervals $(-\infty, b_1]$ or $[a_z, \infty)$ implied by convention, respectively. Let $\hat{\beta}_{\mathcal{T}^*,j} = \boldsymbol{v}^\top \boldsymbol{y}$ be the observed value of the test statistic, which lies in the interval $[a_l, b_l]$ for some $l \in \{1, \ldots, z\}$. Then, following Tibshirani et al. (2016), a p-value $p \sim \mathcal{U}[0, 1]$ for the two-sided significance test for (2) based on $\hat{\beta}_{\mathcal{T}^*,j}$ can be calculated via $p = 2 \cdot \min(\tilde{p}, 1 - \tilde{p})$, with $\tilde{p}$ being the p-value of the one sided test. In our setting and as we

allow for multiple disjoint truncation intervals, we can define this as $\tilde{p} = \mathbb{P}_{H_0}(\boldsymbol{v}^\top \boldsymbol{Y} > \hat{\beta}_{\mathcal{T}^*,j} \mid \text{selection event}, \boldsymbol{P}_v^\perp \boldsymbol{Y} = \boldsymbol{P}_v^\perp \boldsymbol{y}) = \Psi_{\text{nom}}/\Psi_{\text{denom}}$, where $\Psi_{\text{nom}} = \psi(b_l) - \psi(\hat{\beta}_{\mathcal{T}^*,j}) + \sum_{i=l+1}^{z} \psi(b_i) - \psi(a_i)$, $\Psi_{\text{denom}} = \sum_{i=1}^{z} \psi(b_i) - \psi(a_i)$ and $\psi(x) = \Phi(\frac{x}{\sigma\sqrt{\boldsymbol{v}^\top \boldsymbol{v}}})$ with cumulative distribution function $\Phi(\cdot)$ of the standard normal distribution. In other words, Ψ_{denom} is equal to the cumulative probability mass for all possible values $\boldsymbol{v}^\top \boldsymbol{Y}$ that comply with the conditioning event, and Ψ_{nom} is the cumulative probability mass of possible values $\boldsymbol{v}^\top \boldsymbol{Y}$ that are larger than $\hat{\beta}_{\mathcal{T}^*,j}$.

As in practice σ^2 is usually unknown, we investigate in simulations the performance and validity of our proposed p-values when plugging in the restricted maximum likelihood estimate $\hat{\sigma}^2_{\text{REML}} = \|\boldsymbol{y} - \boldsymbol{X}_{\mathcal{T}^*} \hat{\boldsymbol{\beta}}_{\mathcal{T}^*}\|^2/(n - p_{\mathcal{T}^*})$ for σ^2. We describe the corresponding results in Section 3. Note that while $\hat{\sigma}^2_{\text{REML}}$ is plugged into the truncated normal conditional distribution for $\boldsymbol{v}^\top \boldsymbol{Y}$, this distribution is exact and does account for estimation of σ^2 in the selection event.

2.4. Conditional confidence intervals

We extend the results of Tibshirani et al. (2016) to allow for the construction of selective confidence intervals if the null distribution is truncated to several intervals. We thus find the quantiles $q_{\alpha/2}$ and $q_{1-\alpha/2}$, for which

$$\mathbb{P}(q_{\alpha/2} \leq \boldsymbol{v}^\top \boldsymbol{\mu} \leq q_{1-\alpha/2} \mid \text{selection event}, \boldsymbol{P}_v^\perp \boldsymbol{Y} = \boldsymbol{P}_v^\perp \boldsymbol{y}) = 1 - \alpha.$$

Analogous to Tibshirani et al. (2016), we can make use of the fact that the truncated normal survival function with multiple truncation limits is also monotonically decreasing in its mean θ as the truncated normal distribution with multiple truncation intervals is a natural exponential family in θ (see Fithian et al., 2014; Lee et al., 2016). The corresponding quantiles can be found via a grid search, where q_α satisfies $1 - F^{\cup_l [a_l, b_l]}_{\mathcal{N}(q_\alpha, \sigma^2 \boldsymbol{v}^\top \boldsymbol{v})}(\boldsymbol{v}^\top \boldsymbol{y}) = \alpha$ with $F^{\mathcal{J}}_{\mathcal{N}(\mu,\sigma^2)}$ being the truncated cumulative normal distribution function with mean μ, variance σ^2 and truncation interval(s) $\mathcal{J} \subseteq (-\infty, \infty)$. In other words, we search for the mean values $\theta = q_{\alpha/2}$ and $\theta = q_{1-\alpha/2}$ of the truncated normal distribution $\mathcal{N}^{\mathcal{J}}(\theta, \sigma^2 \boldsymbol{v}^\top \boldsymbol{v})$, for which the observed value $\boldsymbol{v}^\top \boldsymbol{y}$ is equal to the $\alpha/2$ and $1 - \alpha/2$ quantile, respectively, and $H_0 : \beta_{\mathcal{T}^*,j} = \theta$ thus would not be rejected.

2.5. Testing groups of variables

Following Loftus and Taylor (2015), a selective χ-significance test for groups of variables can be constructed by testing the null hypothesis $H_0 : \tilde{\boldsymbol{P}}_g \boldsymbol{\mu} = \boldsymbol{0}$, where $\tilde{\boldsymbol{P}}_g = \tilde{\boldsymbol{X}}_{\mathcal{T}^*,g} (\tilde{\boldsymbol{X}}^\top_{\mathcal{T}^*,g} \tilde{\boldsymbol{X}}_{\mathcal{T}^*,g})^{-1} \tilde{\boldsymbol{X}}^\top_{\mathcal{T}^*,g}$, $\tilde{\boldsymbol{X}}_{\mathcal{T}^*,g} = (\boldsymbol{I} - \boldsymbol{P}_{\mathcal{T}^*\setminus g}) \boldsymbol{X}_{\mathcal{T}^*,g}$, $\boldsymbol{X}_{\mathcal{T}^*,g}$ are the columns of the grouped variable g in $\boldsymbol{X}_{\mathcal{T}^*}$, $\boldsymbol{P}_{\mathcal{T}^*\setminus g}$ is the projection onto the column space of $\boldsymbol{X}_{\mathcal{T}^*\setminus g}$ and $\boldsymbol{X}_{\mathcal{T}^*\setminus g}$ are the columns of $\boldsymbol{X}_{\mathcal{T}^*}$ without $\boldsymbol{X}_{\mathcal{T}^*,g}$. Without model selection, a test statistic is given by $T = \sigma^{-1} \|\tilde{\boldsymbol{P}}_g^\top \boldsymbol{Y}\|_2 \overset{H_0}{\sim} \chi_{\text{Trace}(\tilde{\boldsymbol{P}}_g)}$, i.e., T^2 follows a χ^2-distribution with $\text{Trace}(\tilde{\boldsymbol{P}}_g)$ degrees of freedom under H_0. When conditioning on $(\boldsymbol{I} - \tilde{\boldsymbol{P}}_g)\boldsymbol{Y} = (\boldsymbol{I} - \tilde{\boldsymbol{P}}_g)\boldsymbol{y} =: \boldsymbol{z}$ and the unit vector $\boldsymbol{u}$ in the direction of $\tilde{\boldsymbol{P}}_g^\top \boldsymbol{y}$, $\boldsymbol{Y}$ can be decomposed as $\boldsymbol{Y} = \boldsymbol{z} + \sigma T \boldsymbol{u}$, such that the only variation is in T. Conditional on the selection event (3), T follows a truncated χ-distribution with truncation limits $\tau_{1/2}$ now given by $\delta = \sigma^2 \boldsymbol{u}^\top \boldsymbol{A} \boldsymbol{u}$, $\zeta = 2\sigma \boldsymbol{u}^\top \boldsymbol{A} \boldsymbol{z}$ and $\xi = \boldsymbol{z}^\top \boldsymbol{A} \boldsymbol{z} + c$. Depending on the sign of δ and the number of solutions $\tau_{1/2} \geq 0$, the truncation set $\mathcal{J} \subseteq [0, \infty)$ is either a closed interval $\mathcal{J} = [\max(0, \tau_1), \tau_2]$, an open interval $\mathcal{J} = [\tau_2, \infty)$, or a union of intervals $\mathcal{J} = [0, \tau_1] \cup [\tau_2, \infty)$. The test for grouped variables with multiple selection events can be treated analogously to Section 2.3 by normalizing the truncated χ distribution analogously, replacing ψ with the cumulative distribution function of the $\chi_{\text{Trace}(\tilde{\boldsymbol{P}}_g)}$-distribution. Note that while the truncated normal distribution is replaced by a truncated χ-distribution, the types of conditioning events do not change when incorporating groups of variables. The only exception is *significance hunting*, for which model selection is then not based on t-statistics of regression coefficients but an F-test as in (5) is typically used. We also note that tests for grouped variables can be employed to test model terms in a semi-parametric regression model $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \sum_{j=p+1}^{p_f} f(\boldsymbol{x}_j) + \boldsymbol{\varepsilon}$, in which smooth terms are incorporated using a basis representation $f(\boldsymbol{x}_j) \approx \boldsymbol{B}_j(\boldsymbol{x}_j)\boldsymbol{\vartheta}_j$ with $\boldsymbol{B}_j(\boldsymbol{x}_j)$ being the vector of basis functions for the jth component evaluated at observed values $\boldsymbol{x}_j$. Testing the basis coefficient vector $\boldsymbol{\vartheta}_j$ against zero then corresponds to testing for a vanishing model term, i.e., $f(\boldsymbol{x}_j) = \boldsymbol{0}$, in the best linear projection of $\boldsymbol{\mu}$ onto the space spanned by the selected linear covariates and selected basis functions. Extensions to selective inference after model selection of penalized smooth terms in non-parametric and functional regression, such as in Aneiros and Vieu (2014), are, however, beyond the scope of this work.

3. Empirical evidence

We evaluate the proposed selective inference concepts in linear models for a forward stepwise selection procedure based on the AIC.

For the simulation study, we consider $p \in \{5, 25\}$ covariates $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p \in \mathbb{R}^n$, $n \in \{30, 150\}$ observations and use the data generating process $\boldsymbol{y} = \boldsymbol{X}^\dagger \boldsymbol{\beta}^\dagger + \boldsymbol{\varepsilon}$. $\boldsymbol{X}^\dagger = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_4)$ respectively $\boldsymbol{\beta}^\dagger = (4, -2, 1, -0.5)^\top$ correspond to the true active covariates respectively their effects and $\boldsymbol{\varepsilon}$ is Gaussian noise with zero mean and variance σ^2, which is determined by the signal-to-noise ratio SNR $\in \{0.5, 1\}$. Covariates are independently drawn from a standard normal distribution (*ind*) or exhibit a correlation of 0.4 (*cor*). For each setting, 100,000 simulation iterations are performed. We present resulting p-values in a *uniform quantiles vs. observed p-value*-plot in Fig. 1, where p-values are calculated on the basis of concepts introduced in Section 2. In the plot, p-values along the diagonal indicate uniformity, which seems to hold for all inactive

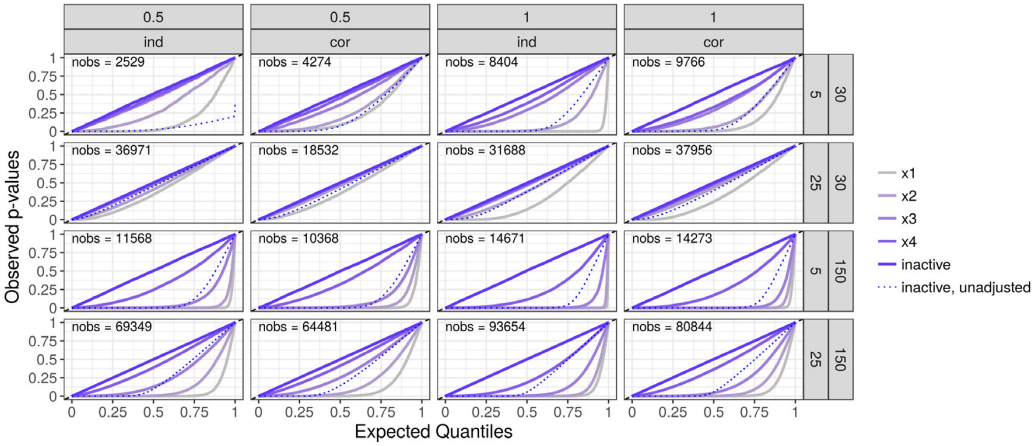

Fig. 1. Quantiles of the standard uniform distribution versus the observed p-values for different SNR and correlation settings (columns) as well as different settings for n and p (rows) in simulation iterations, in which all of the active covariates and additional inactive covariates are selected. p-values were calculated on the basis of the true variance. For each setting, the number of iterations (nobs) is noted in the left upper corner. The dotted line indicates the non-uniformity of "naive" p-values for inactive variables, which are not adjusted for model selection.

Table 1
Coverage of selective 95% confidence intervals for the simulation setting with correlation, $n = 150$, $p = 25$ and SNR $= 1$ for selection cases in which all the active (and potentially additional inactive) variables are selected after AIC stepwise forward selection. The coverage is estimated using 8725 observations for active and 31,371 observations for inactive variables.

	Inactives	x_1	x_2	x_3	x_4
Using true variance	0.9516	0.9492	0.9485	0.9532	0.9542
Using plugin estimate	0.9496	0.9485	0.9457	0.9515	0.9532

variables in all given simulation settings. For active variables, the corresponding selective test shows higher power the closer the point line of p-values runs along the axis. Results are based on those simulation iterations in which all of the active covariates and additional inactive covariates are selected. Note that in the selective inference framework, p-values of inactive variables should exhibit uniformity given any particular set of selection events, if the null hypothesis holds. Aggregating across selected models in each panel of Fig. 1 results in mixture distributions for the p-values, with a mixture of uniform $\mathcal{U}[0, 1]$ variables again being $\mathcal{U}[0, 1]$. Results for iterations without selected inactive variables (not shown) are similar in terms of power. In summary, p-values for inactive variables exhibit uniformity in every setting. p-values for active covariates indicate large power in most of the settings, with notable exceptions for those simulation settings in which p is relatively large in comparison to n.

Further results are given in the supplementary material, showing the resulting p-values for simulation iterations in which the selected model is misspecified due to missing active variables and potentially selected inactive variables. Here, p-values of inactive variables exhibit some deviation from the uniform quantiles when not all of the active variables have been selected. However, deviations mainly occur when inactive variables are correlated with unselected active variables in which case the null hypothesis (2) in fact does not exactly hold. This is due to the fact that the linear projection of $\boldsymbol{\mu}$ into the column space of the selected design matrix has a non-zero coefficient for the jth variable if a correlated variable is omitted from the model. For the setting with correlation, $n = 150$, $p = 25$ and SNR $= 1$, Table 1 additionally provides the estimated coverage for the confidence intervals constructed as in Section 2.4, averaging over all iterations where at least all the active variables are selected (and over inactive variables for the inactives column). In addition, we investigate the performance of our approach when plugging in $\hat{\sigma}^2_{\text{REML}}$ for σ^2 in the derived distribution of $\hat{\beta}_{\mathcal{T},j}$ for all simulation settings (see supplementary material). p-values for inactive variables still approximately exhibit a uniform distribution when using an estimate for σ^2. Notable deviations in comparison to p-values calculated with the true variance can occur when σ^2 is not estimated well such as for $n = 30$ and $p = 25$. Furthermore, as shown in Table 1, almost no difference in the coverage of selective confidence intervals is obtained when plugging in $\hat{\sigma}^2_{\text{REML}}$ for σ^2. In the supplementary material, we also provide results for a simulation study for the χ-test after stepwise AIC selection with a group noise variable.

We additionally apply our approach to the prostate cancer data set (Stamey et al., 1989), which has also been used in Tibshirani et al. (2016) to illustrate selective confidence intervals after forward stepwise regression (see the supplementary material). When using $\alpha = 0.05$, the significant variables match the two significant variables after forward stepwise regression in Tibshirani et al. (2016), although the selected model is different. Compared to unadjusted inference, confidence intervals become wider for all coefficients in the selective inference framework.

4. Summary

Based on the general selective inference framework derived in Loftus and Taylor (2014), Tibshirani et al. (2016) and Loftus and Taylor (2015), we address the issue of conducting valid inference in linear models after likelihood- or test-based model selection, which comprises (iterative) model selection based on the AIC or BIC, model selection via likelihood-based or F-tests and significance hunting based on t-tests. We explicitly derive the necessary conditional distributions for these selection events, which allow the application of selective inference to additional practically relevant settings compared to existing results. We extend the construction of p-values and confidence intervals to the case in which the distribution of the test statistic conditional on the selection events is truncated to multiple intervals. In simulations, we see that obtained p-values yield desirable properties even if the selected model is not correctly specified and confidence intervals show the nominal coverage. We make available an R software package (Rügamer, 2017) for selective inference to apply the proposed framework in practice.

Acknowledgments

We thank Fabian Scheipl and the two reviewers for their useful comments.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.spl.2018.04.010.

References

Akaike, H., 1973. Information theory and an extension of the maximum likelihood principle. In: Petrov, B.N., Csaki, F. (Eds.), Proceedings of the 2nd International Symposium on Information Theory. Akademiai Kiado, pp. 267–281.

Aneiros, G., Vieu, P., 2014. Variable selection in infinite-dimensional problems. Statist. Probab. Lett. 94, 12–20.

Berk, R., Brown, L., Buja, A., Zhang, K., Zhao, L., et al., 2013. Valid post-selection inference. Ann. Statist. 41 (2), 802–837. http://dx.doi.org/10.1214/12-AOS1077.

Buehler, R.J., Feddersen, A.P., 1963. Note on a conditional property of Student's t. Ann. Math. Stat. 34, 1098–1100.

Fithian, W., Sun, D., Taylor, J., 2014. Optimal inference after model selection, ArXiv e-prints arXiv:1410.2597 [math.ST].

Lee, J.D., Sun, D.L., Sun, Y., Taylor, J.E., 2016. Exact post-selection inference, with application to the lasso. Ann. Statist. 44 (3), 907–927.

Leeb, H., Pötscher, B.M., 2005. Model selection and inference: facts and fiction. Econometric Theory 21 (1), 21–59.

Lehmann, E., Scheffé, 1955. Completeness, similar regions, and unbiased estimation: Part ii. Indian J. Stat. 15 (3), 219–236.

Loftus, J.R., Taylor, J.E., 2014. A significance test for forward stepwise model selection, ArXiv e-prints arXiv:1405.3920 [stat.ME].

Loftus, J.R., Taylor, J.E., 2015. Selective inference in regression models with groups of variables, ArXiv e-prints arXiv:1511.01478 [stat.ME].

Mundry, R., Nunn, C., 2009. Stepwise model fitting and statistical inference: Turning noise into signal pollution. Amer. Nat. 173 (1), 119–123. PMID: 19049440.

Rügamer, D., 2017. coinflibs: Conditional Inference after Likelihood-based Selection. https://github.com/davidruegamer/coinflibs. R package version 0.0.0.9000.

Schwarz, G., 1978. Estimating the dimension of a model. Ann. Statist. 6 (2), 461–464.

Stamey, T., Kabalin, J., McNeal, J., Johnstone, I., Freiha, F., Redwine, E., Yang, N., 1989. Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. II. Radical prostatectomy treated patients. J. Urol. 141 (5), 1076–1083.

Tibshirani, R.J., Taylor, J., Lockhart, R., Tibshirani, R., 2016. Exact post-selection inference for sequential regression procedures. J. Amer. Statist. Assoc. 111 (514), 600–620.

Zhang, Z., 2016. Variable selection with stepwise and best subset approaches. Annals of Translational Medicine 4 (7). http://dx.doi.org/10.21037/atm.2016.03.35.

Chapter 4

Valid Inference for L_2-Boosting

Chapter 4 extends the methodology of Chapter 3 to the selection process induced by L_2-Boosting, an important special case of the component-wise functional gradient descent algorithm. In contrast to the previous chapter, the conditional distribution of commonly used test statistics, conditional on the model selection with L_2-Boosting, cannot derived analytically. A selective sampling idea is presented, on the basis of which tests and confidence intervals for linear, grouped and penalized base-learners in the boosted regression model can be constructed. The framework is verified in simulation studies and applied to the prostate cancer data set.

Contributing article:

Rügamer, D. and Greven, S. (2018b). Valid inference for L_2-boosting. *arXiv e-prints arXiv:1805.01852*.

Author contributions:

The manuscript was written by David Rügamer. Sonja Greven added valuable input and proofread the manuscript.

Supplementary material available at:

`https://github.com/davidruegamer/inference_boosting`

Valid Inference for L_2-Boosting

David Rügamer
and
Sonja Greven
Department of Statistics, LMU Munich

May 4, 2018

Abstract

We review several recently proposed post-selection inference frameworks and assess their transferability to the component-wise functional gradient descent algorithm (CFGD) under normality assumption for model errors, also known as L_2-Boosting. The CFGD is one of the most versatile toolboxes to analyze data, as it scales well to high-dimensional data sets, allows for a very flexible definition of additive regression models and incorporates inbuilt variable selection. Due to the iterative nature, which can repeatedly select the same component to update, an inference framework for component-wise boosting algorithms requires adaptations of existing approaches; we propose tests and confidence intervals for linear, grouped and penalized additive model components estimated using the L_2-boosting selection process. We apply our framework to the prostate cancer data set and investigate the properties of our concepts in simulation studies.

Keywords: Bootstrap, Functional Gradient Descent Boosting, Post-Selection Inference, Selective Inference

1 Introduction

Inference for Boosting. In this work we review and adapt recently proposed inference techniques to the component-wise functional gradient descent algorithm (CFGD; see, e.g., Hothorn et al. 2010), which emerged from the field of machine learning (c.f. Friedman 2001), but has since also become an algorithm used to estimate statistical models (see, e.g., Mayr et al. 2017, Melcher et al. 2017, Rügamer et al. 2018, Brockhaus et al. 2018). A commonly used and well studied special CFGD algorithm is L_2-*Boosting* (Bühlmann & Yu 2003). Apart from Luo & Spindler (2017), who study uncertainty for treatment effects when selecting control variables via L_2-Boosting in instrumental variable models, which require additional assumptions for all the variables in the model, no general inferential concepts in the sense of classical statistical inference have been proposed for L_2-Boosting yet, though ad-hoc solutions such as a non-parametric bootstrap are often used to quantify the uncertainty of boosting estimates (see e.g. Brockhaus et al. 2015, Rügamer et al. 2018). In many research areas such an uncertainty quantification is indispensable. We therefore propose a framework for conducting valid inference for regression coefficients in models fitted with L_2-Boosting by conditioning on the selected covariates. We adapt recent research findings on *selective inference*, which transfers classical statistical inference to algorithms that rely on a preceding selection of model terms as is the case for CFGD algorithms. Compared to existing approaches for sequential regression procedures including forward stepwise regression (Tibshirani et al. 2016) inference for L_2-Boosting carries additional challenges due to an iterative procedure that can repeatedly select the same model term.

Suitable inference concepts. The necessity for an explicit inference framework for methods with preceding selection is due to the invalidity of inference after model selection. This invalidity has been mentioned by many authors throughout the last decades (see, e.g., Berk et al. 2013). Different approaches for inference in high-dimensional regression models have emerged over the past years, including data splitting (Wasserman & Roeder 2009). Apart from these techniques, post-selection inference (PoSI; Berk et al. 2013) attracts growing interest. Initiated by the proposal for valid statistical inference after arbitrary selection procedures by Berk et al. (2013), many new findings and adoptions of post-

selection inference to known statistical methods have been published in the last years.

We here focus on *selective inference*, which provides inference statements conditional on the observed model selection. Similar to data splitting, selective inference separates the information in the data, which is used for the model selection, from the information, which is used to infer about parameters post model selection. In contrast to the original PoSI idea of providing simultaneous inference for every possible model selection, selective inference is designed to yield less conservative inference statements.

Apart from general theory described in Fithian et al. (2014), which transfers the classical theory to selective inference in exponential family models following any type of selection mechanism, different explicit frameworks for several selection methods have been derived (see e.g. Lee et al. 2016, for selective inference after Lasso selection or Rügamer & Greven 2018, for selective inference after likelihood- and test-based model selection). Recent publications, which are particularly relevant for this work, aim for valid inference in forward stepwise regression (Tibshirani et al. 2016, Loftus & Taylor 2014, 2015). Whereas Tibshirani et al. (2016) build a framework for any sequential regression technique resulting in a limitation to the space for inference, which can be characterized by a polyhedral set, Loftus & Taylor (2014, 2015) extend the idea to a more general framework, for which the inference space is given by quadratic inequalities and coincides with the polyhedral approach in special cases. A continuation of Loftus & Taylor (2015) is given by Yang et al. (2016). With the objective to build a selective inference framework for the group Lasso (Yuan & Lin 2006), Yang et al. describe an importance sampling algorithm that circumvents the problem of having to explicitly define the space, to which the inference is restricted after conditioning.

Resampling for uncertainty quantification. Uncertainty quantification by the use of resampling methods is as error-prone as classical inference when applied to models after a certain model selection procedure. We therefore will shortly address this issue by the example of bootstrap as one of the most commonly used techniques.

Let us first consider the parametric bootstrap. When generating new samples of the response from the selected model and proceeding as in unadjusted inference post model-

selection, the selected model is treated as the true model and this can incorrectly lead effects to be (non-)zero. A non-parametric bootstrap on the other hand is accompanied by its own problems. First, when drawing pairs of response and covariates, we (implicitly) assume that the underlying data model is based on a random design in contrast to many regression model settings, where the covariates are assumed to be fixed. If we ignore this issue, we still face the problem of either neglecting the uncertainty of model selection, if we refit the initially selected model for the resampled data, or the problem of having to aggregate over different models when integrating the model selection process into our resampling procedure. If estimates are aggregated over different models, uncertainty quantification of parameters is based on different selected models with different interpretations of the estimated coefficients and thus does not correspond to a meaningful single null hypothesis.

An additional difficulty arises when using the bootstrap for boosted regression models, in which the estimated coefficients exhibit a bias due to the shrinkage effect of boosting. Hence, bootstrap intervals are not centered around the true value and thus yield a quantification of variability rather than a measure of deviation from the truth.

Contribution of this work. In this work, we adapt and extend several existing approaches for selective inference, thereby addressing the following issues:

1. We explicitly derive the space restriction of the response given by the L_2-Boosting path and thereby allow for inference as proposed in Tibshirani et al. (2016).

2. We propose a new conditional inference concept for L_2-Boosting and potentially other slow learning algorithms by conditioning on a set of possible selection paths.

3. We combine the work of Tibshirani et al. (2016) and Yang et al. (2016) to allow for the computation of p-values and confidence intervals using test statistics, which lie in a union of polyhedra and therefore have a (conditional) normal distribution with potentially multiple truncation limits.

4. We explain how the proposed inference concept can easily be extended to account for cross-validation, stability selection (Shah & Samworth 2013) and similar sub-sampling methods.

5. We extend the idea of the selective inference framework to models including L_2-penalized additive effects, such as smooth effects.

In the following, we describe the L_2-Boosting algorithm in section 2 and recapitulate the concept of selective inference for sequential regression procedures in section 3. In section 4 we investigate the challenges accompanying a new inference framework for L_2-Boosting and propose several solutions. In section 5 we present simulation results and analyze the prostate cancer data using our new approach in section 6. We discuss limitations and further extensions of the approach in section 7. An add-on R-package to the model-based boosting R package `mboost` is available at `https://github.com/davidruegamer/iboost`, which can be used to conduct inference for boosted models and to reproduce the results of section 5 and 6. Supplementary materials including further simulation results are available at `https://github.com/davidruegamer/inference_boosting`.

2 L_2-Boosting

We now present the L_2-Boosting algorithm as a special generic CFGD algorithm. Let $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ be a fixed set of covariates and $\boldsymbol{y}$ a realization of the random response variable $\boldsymbol{Y} \in \mathbb{R}^n$. The goal is to minimize a loss function $\ell(\cdot, \boldsymbol{y})$ for the given realization $\boldsymbol{y}$ with respect to an additive model $\boldsymbol{f} := \sum_{j=1}^{J} g_j(\boldsymbol{X}_j)$, where function evaluations of g_j are evaluated row-wise. The functions $g_j(\cdot)$, the so called base-learners, are defined for column subsets $\boldsymbol{X}_j \in \mathbb{R}^{n \times p_j}$ of $\boldsymbol{X}$ with $1 \leq p_j \leq p$ and can be fitted to some vector $\boldsymbol{u} \in \mathbb{R}^n$, which yields $\hat{\boldsymbol{g}}_j$ as estimate for $g_j(\boldsymbol{X}_j)$. We estimate $\boldsymbol{f}$ by $\hat{\boldsymbol{f}}$ using the component-wise functional gradient descent algorithm:

(1) Initialize an offset value $\hat{\boldsymbol{f}}^{(0)} \in \mathbb{R}^n$. If $\boldsymbol{y}$ is centered, a natural choice is $\hat{\boldsymbol{f}}^{(0)} = (0, \ldots, 0)^\top$. Define $m = 0$.

(2) Do the following for $m = 1, \ldots, m_{stop}$:

(2.1) Compute the pseudo-residuals $\boldsymbol{u}^{(m)} \in \mathbb{R}^n$ of step m as $\boldsymbol{u}^{(m)} = -\frac{\partial}{\partial \boldsymbol{f}} \ell(\boldsymbol{f}, \boldsymbol{y})\Big|_{\boldsymbol{f}=\hat{\boldsymbol{f}}^{(m-1)}}$.

(2.2) Approximate the negative gradient vector with $\hat{\boldsymbol{g}}_j$ by fitting each of the base-learners $g_j(\cdot), j = 1, \ldots, J$ to the pseudo-residuals and find the base-learner $j^{(m)}$, for which $j^{(m)} = \text{argmin}_{1 \leq j \leq J} ||\boldsymbol{u}^{(m)} - \hat{\boldsymbol{g}}_j||_2^2$ holds.

(2.3) Update $\hat{\boldsymbol{f}}^{(m)} = \hat{\boldsymbol{f}}^{(m-1)} + \nu \cdot \hat{\boldsymbol{g}}_{j^{(m)}}$, where $\nu \in (0, 1]$ is the so called *step-length* or *learning rate*.

When defining $\ell(\boldsymbol{f}, \boldsymbol{y}) = \frac{1}{2}||\boldsymbol{y} - \boldsymbol{f}||_2^2$ with quadratic L_2-Norm $||\cdot||_2^2$, L_2-Boosting is obtained, which corresponds to mean regression using the model $\mathbb{E}(\boldsymbol{Y}|\boldsymbol{X}) = \sum_{j=1}^{J} g_j(\boldsymbol{X}_j)$. The vector $\boldsymbol{u}^{(m)}$ then corresponds to the residuals $\boldsymbol{y} - \hat{\boldsymbol{f}}^{(m)}$. In the framework of additive regression models, each base-learner $g_j(\cdot)$ constitutes a partial effect and is represented as linear effect of a covariate or of a basis evaluated at that covariate vector, i.e., $\boldsymbol{g}_j(\boldsymbol{X}_j) = \boldsymbol{X}_j\boldsymbol{\beta}_j$. $\boldsymbol{\beta}_j$ is estimated using ordinary or penalized least squares. The model fit $\hat{\boldsymbol{g}}_j^{(m)}$ of each base-learner in the mth step is therefore given by $\hat{\boldsymbol{g}}_j^{(m)} = \boldsymbol{H}_j\boldsymbol{u}^{(m)} = \boldsymbol{X}_j(\boldsymbol{X}_j^\top\boldsymbol{X}_j + \lambda_j\boldsymbol{D}_j)^{-1}\boldsymbol{X}_j^\top\boldsymbol{u}^{(m)}$, where the hat matrix $\boldsymbol{H}_j$ is defined by the corresponding design matrix $\boldsymbol{X}_j$, a penalty matrix $\boldsymbol{D}_j$ and a pre-specified smoothing parameter $\lambda_j \geq 0$ controlling the penalization. As only one base-learner is chosen in each iteration, the final effective degrees of freedom of the jth base-learner depend on the number of selections.

As L_2-Boosting scales well to large data sets due to its component-wise fitting nature and is particularly suited for the estimation of structured additive regression models, it is often used as an estimation algorithm for a statistical additive model (see, e.g., Mayr et al. 2017). It has the additional advantage of being able to handle $n < p$-settings and conducting variable selection, as not all J model terms are necessarily selected in at least one iteration. However, when constructing a measure of uncertainty for regression coefficients, the preceding variable selection has to be accounted for. As for other variable selection procedures, the iterative nature of L_2-Boosting restricts the space of $\boldsymbol{Y}$ and thereby the space of estimated parameters.

3 A Review of Selective Inference for Sequential Regression Procedures

We first define the considered model framework and some necessary notations before reviewing existing selective inference approaches we build on in Section 4. Let $\boldsymbol{Y} = \boldsymbol{\mu} + \boldsymbol{\varepsilon}$ with $\boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I}_n)$ and n-dimensional identity matrix $\boldsymbol{I}_n$. Furthermore, assume that σ^2 is known and $\boldsymbol{\mu}$ is an unknown parameter of interest. In particular, we do not assume

any true linear relationship between $\boldsymbol{\mu}$ and covariates, but estimate $\boldsymbol{\mu}$ with a "working model", which is of additive nature based on fixed covariates $\boldsymbol{X} \in \mathbb{R}^{n \times p}$, for which p potentially exceeds n. Furthermore, define the selection procedure or selection event $\mathcal{S}$: $\mathbb{R}^n \to \mathcal{P}(\{1, \ldots, p\}), \boldsymbol{y} \mapsto \mathcal{S}(\boldsymbol{y})$ with power set function $\mathcal{P}(\cdot)$. For the given realization $\boldsymbol{y}$ of $\boldsymbol{Y}$, we denote $\mathcal{S}(\boldsymbol{y}) =: \mathcal{A}$, for which we assume $|\mathcal{A}| \leq n$.

We focus on estimating the best linear projection of $\boldsymbol{\mu}$ into the space spanned by the variables given by $\mathcal{A}$ after model selection. We therefore run the selection procedure defined by $\mathcal{S}$, select the subset $\boldsymbol{X}_\mathcal{A}$ of $\boldsymbol{X}$ defined by the selected column indices $\mathcal{S}(\boldsymbol{y}) = \mathcal{A}$ and estimate regression coefficients $\boldsymbol{\beta}_\mathcal{A}$ by projecting $\boldsymbol{y}$ into the linear subspace $\boldsymbol{W}_\mathcal{A} \subseteq \mathbb{R}^n$ spanned by the columns of $\boldsymbol{X}_\mathcal{A}$. With the goal to infer about $\beta_j, j \in \mathcal{A}$, in $\boldsymbol{\beta}_\mathcal{A}$, we test the hypothesis $H_0 : \beta_j = 0$. This is equivalent to testing

$$H_0 : \boldsymbol{v}^\top \boldsymbol{\mu} := \boldsymbol{e}_j^T (\boldsymbol{X}_\mathcal{A}^\top \boldsymbol{X}_\mathcal{A})^{-1} \boldsymbol{X}_\mathcal{A}^\top \boldsymbol{\mu} = 0 \tag{1}$$

with $\boldsymbol{e}_j$ the unit vector selecting $j \in \mathcal{A}$ (see, e.g., Tibshirani et al. 2016).

3.1 Inference based on a Polyhedral Space Characterization

In a classical statistical approach without selection, (1) is tested by using $\tilde{R} := \boldsymbol{v}^\top \boldsymbol{Y}$, which follows a normal distribution with expectation $\tilde{\rho} = \boldsymbol{v}^\top \boldsymbol{\mu}$ and variance $\sigma^2 \boldsymbol{v}^\top \boldsymbol{v}$ under the null. However, after model selection, the space of $\boldsymbol{Y}$ is restricted to $\mathcal{G} = \{\boldsymbol{y} : \mathcal{S}(\boldsymbol{y}) = \mathcal{A}\}$, which we call the *inference region*. Many of the proposed methods for selective inference then describe this space restriction mathematically and derive the distribution of $\boldsymbol{v}^\top \boldsymbol{Y} \mid \boldsymbol{Y} \in \mathcal{G}$. For sequential regression procedures such as Forward Stepwise Regression (FSR) or the Least Angle Regression (LAR, Efron et al. 2004), Tibshirani et al. (2016) characterize the restricted region of the on-going selection mechanism as a polyhedral set $\mathcal{G} = \{\boldsymbol{y} : \Gamma \boldsymbol{y} \geq \boldsymbol{b}\}$ with $\Gamma \in \mathbb{R}^{\varkappa \times n}$, $\boldsymbol{b} \in \mathbb{R}^\varkappa$ for some $\varkappa \in \mathbb{N}$ and an inequality $\geq$ which is to be interpreted componentwise. In other words, for FSR, LAR and also for other algorithms, Γ and $\boldsymbol{b}$ can be explicitly derived by reformulating inequalities determining the selection in each step. As shown in 4, this is also the case for L_2-Boosting when conditioning on the selection path. Let $\boldsymbol{P}_\boldsymbol{W}$ be the projection onto a linear subspace $\text{span}(\boldsymbol{W}) \subset \mathbb{R}^n$ defined by $\boldsymbol{W} \in \mathbb{R}^{n \times w}, w \geq 1$ and $\boldsymbol{P}_\boldsymbol{W}^\perp$ be the projection onto the orthogonal complement of this linear subspace. Furthermore, define the direction of $\boldsymbol{P}_\boldsymbol{W} \boldsymbol{y}$ as the unit vector $\text{dir}_\boldsymbol{W}(y) = \frac{\boldsymbol{P}_\boldsymbol{W} \boldsymbol{y}}{||\boldsymbol{P}_\boldsymbol{W} \boldsymbol{y}||_2}$.

In the framework of Tibshirani et al. (2016), $\boldsymbol{Y}$ is written as $\tilde{R}\cdot\frac{\boldsymbol{v}}{\boldsymbol{v}^\top\boldsymbol{v}}+\boldsymbol{Z}$ with $\boldsymbol{Z}=\boldsymbol{P}_v^\perp\boldsymbol{Y}$. By construction $\boldsymbol{Z}$ is independent of $\tilde{R}$. The selection event $\boldsymbol{Y}\in\mathcal{G}$ can thus be rewritten

$$\mathcal{G}=\{\boldsymbol{Y}\text{ with }\mathcal{V}^{lo}(\boldsymbol{Z})\leq\tilde{R}\leq\mathcal{V}^{up}(\boldsymbol{Z}),\mathcal{V}^0(\boldsymbol{Z})\geq 0\},\tag{2}$$

where $\mathcal{V}^{lo}$, $\mathcal{V}^{up}$ and $\mathcal{V}^0$ are functions of $\boldsymbol{Z}$ as well as of the fixed quantities Γ and $\boldsymbol{v}$. By additionally conditioning on the realization $\boldsymbol{z}$ of $\boldsymbol{Z}$ as well as on a list of signs for each step similar to those defined in (9) and which will be explained in Section 4, $\mathcal{V}^{lo}$, $\mathcal{V}^{up}$ are fixed limits for $\tilde{R}$ (see, e.g., Lee et al. 2016) with $\boldsymbol{Y}\in\mathcal{G}$ corresponding to $\tilde{R}\in\mathcal{R}_y:=\{\tilde{R}:\ \mathcal{V}^{lo}(\boldsymbol{z})\leq\tilde{R}\leq\mathcal{V}^{up}(\boldsymbol{z})\}$. Incorporating these boundaries into the distribution of $\tilde{R}\sim\mathcal{N}(\rho,\sigma^2\boldsymbol{v}^\top\boldsymbol{v})$ yields a truncated Gaussian distribution with truncation limits $\mathcal{V}^{lo}=\mathcal{V}^{lo}(\boldsymbol{z}),\mathcal{V}^{up}=\mathcal{V}^{up}(\boldsymbol{z})$. Let $F^{[\mathcal{V}^{lo},\mathcal{V}^{up}]}_{\tilde{\rho},\sigma^2\boldsymbol{v}^\top\boldsymbol{v}}(\tilde{R})$ denote the cumulative distribution function of the truncated normal distribution evaluated at $\tilde{R}$. Then, for

$$H_0:\tilde{\rho}\leq 0\qquad\text{vs.}\qquad H_1:\tilde{\rho}>0,$$

the test statistic

$$T=1-F^{[\mathcal{V}^{lo},\mathcal{V}^{up}]}_{0,\sigma^2\boldsymbol{v}^\top\boldsymbol{v}}(\tilde{R})$$

is a valid conditional p-value, conditional on the polyhedral selection, as

$$\mathbb{P}_{H_0}(T\leq\alpha\mid\Gamma\boldsymbol{Y}\geq\boldsymbol{b})=\alpha$$

for any $0\leq\alpha\leq 1$. For a two-sided hypothesis

$$H_0:\tilde{\rho}=0\qquad\text{vs.}\qquad H_1:\tilde{\rho}\neq 0,$$

Tibshirani et al. (2016) define

$$T=2\cdot\min\left(F^{[\mathcal{V}^{lo},\mathcal{V}^{up}]}_{0,\sigma^2\boldsymbol{v}^\top\boldsymbol{v}}(\tilde{R}),1-F^{[\mathcal{V}^{lo},\mathcal{V}^{up}]}_{0,\sigma^2\boldsymbol{v}^\top\boldsymbol{v}}(\tilde{R})\right)$$

and the validity of inference based on this p-value holds analogously. A valid conditional confidence interval $[\delta_{\alpha/2},\delta_{1-\alpha/2}]$ can then be derived by inverting the given test, i.e., finding the limits $\delta_{\alpha/2}$ and $\delta_{1-\alpha/2}$, which satisfy $1-F^{[\mathcal{V}^{lo},\mathcal{V}^{up}]}_{\delta_{\alpha/2},\sigma^2\boldsymbol{v}^\top\boldsymbol{v}}(\tilde{r})=\alpha/2$ and $1-F^{[\mathcal{V}^{lo},\mathcal{V}^{up}]}_{\delta_{1-\alpha/2},\sigma^2\boldsymbol{v}^\top\boldsymbol{v}}(\tilde{r})=1-\alpha/2$ for the observed value $\tilde{R}=\tilde{r}$. Limits in this case are unique due to the monotonicity of the survival function $1-F^{[\mathcal{V}^{lo},\mathcal{V}^{up}]}_{\gamma,\sigma^2\boldsymbol{v}^\top\boldsymbol{v}}(\tilde{r})$ in the mean γ. For more details, see section 4 and Tibshirani et al. (2016).

The characterization of the inference region as a polyhedral set, however, is only possible if the algorithmic decision in each selection step is a linear restriction on the space of $\boldsymbol{Y}$. For example for groups of variables, the underlying inequality for the choice of the covariate is inherent quadratic and no polyhedral representation can be obtained. Loftus & Taylor (2015) therefore introduce a framework for inference after model selection procedures which can be described by affine inequalities.

Apart from a different characterization of the space restriction, a different test statistic must be used for groups of variables. For testing the jth group variable coefficient $\boldsymbol{\beta}_{\mathcal{A},j} \in \mathbb{R}^w$ in the best linear approximation $\boldsymbol{\beta}_{\mathcal{A}} = \arg\min \mathbb{E}\left[\,||\boldsymbol{Y} - \boldsymbol{X}_{\mathcal{A}}\boldsymbol{\beta}||_2^2\,\right]$, Loftus & Taylor (2015), Yang et al. (2016) rewrite the null hypothesis $\boldsymbol{\beta}_{\mathcal{A},j} = \boldsymbol{0}$ as $\boldsymbol{P}_{\boldsymbol{W}}\boldsymbol{\mu} = \boldsymbol{0} \Leftrightarrow$

$$H_0 : \rho := ||\boldsymbol{P}_{\boldsymbol{W}}\boldsymbol{\mu}||_2 = 0 \tag{3}$$

with $\boldsymbol{W} = \boldsymbol{P}^{\perp}_{\boldsymbol{X}_{\mathcal{A}\setminus j}}\boldsymbol{X}_j$, where $\boldsymbol{X}_{\mathcal{A}\setminus j}$ denotes $\boldsymbol{X}_{\mathcal{A}}$ without the p_j columns corresponding to the jth group variable. In other words we want to test the correlation of $\boldsymbol{X}_j$ and $\boldsymbol{\mu}$ after adjusting for all other predictors $\mathcal{A}\setminus j$ in the selected model $\mathcal{A}$. Using $R := ||\boldsymbol{P}_{\boldsymbol{W}}\boldsymbol{Y}||_2$ as test statistic, the authors then conduct inference. Under the null and when additionally conditioning on the direction $\text{dir}_{\boldsymbol{W}}(\boldsymbol{y})$, R follows a truncated χ-distribution and truncation limits of R can again be derived analytically. With the goal to also facilitate the computation of confidence intervals, Yang et al. (2016) note that R and $\text{dir}_{\boldsymbol{W}}(\boldsymbol{y})$ are not independent for $\rho \neq 0$ and as a consequence, the χ-conditional distribution of R as derived in Loftus & Taylor (2015) for (3) when $\rho = 0$ no longer holds for more general hypotheses.

Similar to (2), Yang et al. (2016) decompose $\boldsymbol{Y}$ as $R \cdot \text{dir}_{\boldsymbol{W}}(\boldsymbol{Y}) + \boldsymbol{P}^{\perp}_{\boldsymbol{W}}\boldsymbol{Y}$ and condition on $\text{dir}_{\boldsymbol{W}}(\boldsymbol{Y}) = \text{dir}_{\boldsymbol{W}}(\boldsymbol{y})$ as well as on $\boldsymbol{P}^{\perp}_{\boldsymbol{W}}\boldsymbol{Y} = \boldsymbol{P}^{\perp}_{\boldsymbol{W}}\boldsymbol{y}$. Then, the only variation left is in R and the selection $\mathcal{A}$ can be equally written as $R \in \mathcal{R}_y$ with

$$\mathcal{R}_y = \left\{R > 0 : \mathcal{S}(R \cdot \text{dir}_{\boldsymbol{W}}(\boldsymbol{y}) + \boldsymbol{P}^{\perp}_{\boldsymbol{W}}\boldsymbol{y}) = \mathcal{A}\right\}.$$

Yang et al. (2016) then derive the conditional distribution of R, conditional on $\text{dir}_{\boldsymbol{W}}(\boldsymbol{y})$ as well as on $\boldsymbol{P}^{\perp}_{\boldsymbol{W}}\boldsymbol{y}$. The corresponding density is

$$f(R) \propto R^{w-1} \exp\left\{-\frac{1}{2\sigma^2}(R^2 - 2R \cdot \langle \text{dir}_{\boldsymbol{W}}(\boldsymbol{y}), \boldsymbol{\mu}\rangle)\right\} \cdot \mathbb{1}\{R \in \mathcal{R}_y\} \tag{4}$$

with indicator function $\mathbb{1}\{\cdot\}$. (4) can be used to conduct inference on the inner product $\langle \text{dir}_{\boldsymbol{W}}(\boldsymbol{y}), \boldsymbol{\mu}\rangle$. As for the quantity of interest $\rho = ||\boldsymbol{P}_{\boldsymbol{W}}\boldsymbol{\mu}||_2 \geq \langle \text{dir}_{\boldsymbol{W}}(\boldsymbol{y}), \boldsymbol{\mu}\rangle$ holds, (4) can

also be used to construct a lower bound for ρ. As a byproduct of generalizing the idea of Loftus & Taylor (2015), the authors additionally bypass the problem of having to define the selection region analytically. We describe this idea in the following in more detail.

3.2 Inference without explicit inference region definition

Whereas most approaches for selective inference require an explicit definition of the space $\mathcal{G}$, to which $\boldsymbol{Y}$ is restricted by the selection procedure, a mathematical description of $\mathcal{G}$ is not always feasible. However, as pointed out by Fithian et al. (2014), Yang et al. (2016), such a characterization is not mandatory when sampling from the conditional distribution of $\boldsymbol{Y}$ is possible. In the following, we describe the idea of Yang et al. (2016), who use an importance sampler when conducting inference for (3).

Theorem 1 in Yang et al. (2016) states that, conditional on $\text{dir}_{\boldsymbol{W}}(\boldsymbol{y})$, $\boldsymbol{P}_{\boldsymbol{W}}^{\perp}\boldsymbol{y}$ and the selection event, inference can be conducted using

$$\varsigma(t) = \frac{\int_{R\in\mathcal{R}_y, R>||\boldsymbol{P}_{\boldsymbol{W}}\boldsymbol{y}||_2} R^{w-1}e^{-(R^2-2Rt)/2\sigma^2}\,\mathrm{d}R}{\int_{R\in\mathcal{R}_y} R^{w-1}e^{-(R^2-2Rt)/2\sigma^2}\,\mathrm{d}R} \tag{5}$$

as $\varsigma(t_Y)$, a p-value for $\langle \text{dir}_{\boldsymbol{W}}(\boldsymbol{y}), \boldsymbol{\mu}\rangle = t_Y$, is Uniform$[0,1]$-distributed. Here, $\varsigma(\cdot)$ can also be seen as the survival function derived from the density defined in (4). In order to circumvent an explicit definition of the selection region $\mathcal{R}_y$, the authors note that (5) is equal to

$$\frac{\mathbb{E}_{R\sim\sigma\chi_w}(e^{Rt/\sigma^2}\cdot\mathbb{1}\{R\in\mathcal{R}_y, R>||\boldsymbol{P}_{\boldsymbol{W}}\boldsymbol{y}||_2\})}{\mathbb{E}_{R\sim\sigma\chi_w}(e^{Rt/\sigma^2}\cdot\mathbb{1}\{R\in\mathcal{R}_y\})}, \tag{6}$$

which can be approximated by the ratio of empirical expectations computed with a large number of samples $r^b \sim \sigma\cdot\chi_w, b = 1,\ldots,B$. In particular, to evaluate the argument of both expectations in (6) for some r^b, $r^b\in\mathcal{R}_y$ must be checked. To this end, note that the only variation of $(\boldsymbol{Y} \mid \text{dir}_{\boldsymbol{W}}(\boldsymbol{y}), \boldsymbol{P}_{\boldsymbol{W}}^{\perp}\boldsymbol{y})$ is in R. We therefore define $\boldsymbol{y}^b = \boldsymbol{P}_{\boldsymbol{W}}^{\perp}\boldsymbol{y} + r^b\cdot\text{dir}_{\boldsymbol{W}}(\boldsymbol{y})$ and rerun the algorithm to check whether $\mathcal{S}(\boldsymbol{y}^b) = \mathcal{A}$, or equivalently, whether $r^b\in\mathcal{R}_y$. Drawing samples from the $\sigma\chi_w$-distribution, however, is less promising when $||\boldsymbol{P}_{\boldsymbol{W}}\boldsymbol{y}||_2$ is large. In this case, $\mathbb{P}(R\in\mathcal{R}_y)$ may be very small and an excessively large number of samples is needed to obtain a good approximation of $\varsigma(t)$. Yang et al. (2016) therefore suggest an importance sampling algorithm, which draws new samples r^b from a proposal

distribution $\mathcal{F}_{prop}$ such as $\mathcal{N}(||\boldsymbol{P_W y}||_2, \sigma^2)$ with density f_{prop} and then approximates (6) by

$$\varsigma(t) \approx \hat{\varsigma}(t) = \frac{\sum_b w_b \cdot e^{r^b t/\sigma^2} \cdot \mathbb{1}\{r^b \in \mathcal{R}_Y, r^b > ||\boldsymbol{P_W y}||_2\}}{\sum_b w_b \cdot e^{r^b t/\sigma^2} \cdot \mathbb{1}\{r^b \in \mathcal{R}_Y\}} \tag{7}$$

with sampling weights $w_b = f_{\sigma\chi_w}(r^b)/f_{prop}(r^b)$.

4 Selective Inference concepts for L_2-Boosting

4.1 Polyhedron representation-based inference for L_2-Boosting

Consider using L_2-Boosting with only linear base-learners to fit a linear regression model. Following Tibshirani et al. (2016) we can derive a polyhedron representation $\mathcal{G} = \{\boldsymbol{y} : \Gamma \boldsymbol{y} \geq \boldsymbol{b}\}$ in a similar fashion to other stepwise regression procedures **for the given selection path** $j^{(1)}, \ldots, j^{(m_{\text{stop}})}$ of L_2-boosting.

This can easily be proven by regarding the residual vector $\boldsymbol{u}^{(m)}$ of step m as a function of $\boldsymbol{y}$. The selection condition for the mth chosen base-learner

$$\begin{aligned} & ||(\boldsymbol{I} - \boldsymbol{H}_{j^{(m)}})\boldsymbol{u}^{(m)}||^2 \leq ||(\boldsymbol{I} - \boldsymbol{H}_j)\boldsymbol{u}^{(m)}||^2 \qquad \forall j \neq j^{(m)} \\ \Leftrightarrow \quad & \left(s_m \boldsymbol{X}_{j^{(m)}}^\top / ||\boldsymbol{X}_{j^{(m)}}||_2 \pm \boldsymbol{X}_j^\top / ||\boldsymbol{X}_j||_2\right) \boldsymbol{u}^{(m)} \geq 0 \qquad \forall j \neq j^{(m)}, \end{aligned} \tag{8}$$

with $s_m = \text{sign}(\boldsymbol{X}_{j^{(m)}}^\top \boldsymbol{u}^{(m)})$, can be written as affine restriction on $\boldsymbol{y}$ by plugging

$$\boldsymbol{u}^{(m)} = \left[\prod_{l=1}^{m-1} \left(\boldsymbol{I} - \nu \boldsymbol{H}_{j^{(m-l)}}\right)\right] =: \Upsilon^{(m)} \boldsymbol{y}$$

into (8). This yields the polyhedron representation $\mathcal{G}$ for a given selection path and list of signs $s_m, m = 1, \ldots, m_{\text{stop}}$ with corresponding $(2 \cdot (p-1) \cdot m_{\text{stop}}) \times n$ matrix Γ as stacked matrix of n-dimensional row vectors, where the rows $\Gamma_{[(\tilde{m}+2j-1):(\tilde{m}+2j),]}$ with $\tilde{m} = 2 \cdot (p-1) \cdot (m-1)$ are given by

$$\left(s_m \boldsymbol{X}_{j^{(m)}}^\top / ||\boldsymbol{X}_{j^{(m)}}||_2 \pm \boldsymbol{X}_j^\top / ||\boldsymbol{X}_j||_2\right) \Upsilon^{(m)} \quad \forall\, j \neq j^{(m)}. \tag{9}$$

As for other procedures described in the post-selection inference literature, this representation only holds if the columns of $\boldsymbol{X}$ are in general position, which however, is not a very stringent assumption (see, e.g., Tibshirani et al. 2016, section 4).

By showing that the L_2-Boosting path results in a space restriction for $\boldsymbol{Y}$, which can be described as a polyhedral set, quantities of interest $\boldsymbol{v}^\top \boldsymbol{\mu}$ can be tested based on the conditional distribution of $\boldsymbol{v}^\top \boldsymbol{Y} \,|\, \boldsymbol{Y} \in \mathcal{G}$ as proposed by Tibshirani et al. (2016). To this end, we have to condition on the selection path. If we do not additionally condition on the list of signs, $\mathcal{G}$ is a union of polyhedra (cf. Lee et al. 2016).

4.2 Choice of the Conditioning Event for Slow Learners

For the selection approaches discussed in Section 3, conditioning on the selection path is equivalent to conditioning on the selected model, which helps in deriving the corresponding conditional distribution. For boosting and other slow learners that can repeatedly select the same base-learner, conditioning on the selection path and thus on variable selection decisions in each algorithmic step will result in a loss of power. In fact, such a conditional inference will have almost no power in most practically relevant situations, as we show empirically for the polyhedron approach in the simulation section. In order to avoid excessive conditioning, we propose to condition only on the set of selected covariates, i.e., on the selected statistical model.

Conditioning only on the selected covariates, however, means that the mathematical description of the inference region becomes far more difficult. For L_2-Boosting with linear base-learners, this would result in a union of not necessarily overlapping polyhedra for the different selection paths leading to the same selected model. In particular for L_2-Boosting, we do not think that an analytical description of the inference region is possible. We thus circumvent this problem using a Monte Carlo approximation, adapting and extending the existing approaches presented in Section 3.

4.3 Powerful Inference for L_2-Boosting with Linear Base-learners

We now combine the ideas of Section 3.1 and 3.2 to practically realize the idea of the previous Section 4.2. We base inference on the potentially multiply truncated Gaussian distribution of $R = \boldsymbol{v}^\top \boldsymbol{Y}$ conditional on $\boldsymbol{P}_v^\perp \boldsymbol{y}$ and the selection $R \in \mathcal{R}_y$. Then, the

truncated normal density of R is given by

$$f(R) \propto \exp\left\{-\frac{1}{2\sigma^2 \boldsymbol{v}^\top \boldsymbol{v}}(R - \boldsymbol{v}^\top \boldsymbol{\mu})^2\right\} \cdot \mathbb{1}\{R \in \mathcal{R}_y\},$$

where $\mathcal{R}_y$ is a union of polyhedra. Let $r_{\text{obs}} = \boldsymbol{v}^\top \boldsymbol{y}$. Then, analogous to Yang et al. (2016) we can define a p-value by

$$P = \frac{\int_{R > r_{\text{obs}}, R \in \mathcal{R}_y} e^{-(2\sigma^2 \boldsymbol{v}^\top \boldsymbol{v})^{-1} R^2} \, \mathrm{d}R}{\int_{R \in \mathcal{R}_y} e^{-(2\sigma^2 \boldsymbol{v}^\top \boldsymbol{v})^{-1} R^2} \, \mathrm{d}R}$$

for $H_0 : \boldsymbol{v}^\top \boldsymbol{\mu} = 0$ and since the truncated Gaussian distribution with potentially multiple truncation limits is monotone increasing in its mean ρ (see, e.g., Rügamer & Greven 2018), we can find unique values $\rho_{\alpha/2}, \rho_{1-\alpha/2}$ for any $\alpha \in (0, 1)$, such that

$$\varsigma(\rho_a) = \frac{\int_{R > r_{\text{obs}}, R \in \mathcal{R}_y} e^{-(2\sigma^2 \boldsymbol{v}^\top \boldsymbol{v})^{-1}(R^2 - 2R\rho_a)} \, \mathrm{d}R}{\int_{R \in \mathcal{R}_y} e^{-(2\sigma^2 \boldsymbol{v}^\top \boldsymbol{v})^{-1}(R^2 - 2R\rho_a)} \, \mathrm{d}R} = a, \quad a \in \{\alpha/2, 1 - \alpha/2\}$$

to construct a confidence interval $[\rho_{\alpha/2}, \rho_{1-\alpha/2}]$. Note that $P = \varsigma(0)$, and $\varsigma(\rho_a)$ can then be rewritten as

$$\frac{\mathbb{E}_{R \sim \mathcal{N}(0, \sigma^2 \boldsymbol{v}^\top \boldsymbol{v})}\left[\mathbb{1}\{R \in \mathcal{R}_y, R > r_{\text{obs}}\} \cdot e^{(\sigma^2 \boldsymbol{v}^\top \boldsymbol{v})^{-1} R \rho_\alpha}\right]}{\mathbb{E}_{R \sim \mathcal{N}(0, \sigma^2 \boldsymbol{v}^\top \boldsymbol{v})}\left[\mathbb{1}\{R \in \mathcal{R}_y\} \cdot e^{(\sigma^2 \boldsymbol{v}^\top \boldsymbol{v})^{-1} R \rho_\alpha}\right]},$$

which allows for an empirical approximation as in (7).

In practice, importance sampling from $\Pi = \mathcal{N}(r_{\text{obs}}, \sigma^2 \boldsymbol{v}^\top \boldsymbol{v})$ works well if truncation limits around r_{obs} are fairly symmetric, yielding the weights $w_b = \exp((2r^b r_{\text{obs}} - r_{\text{obs}}^2)/(-2\sigma^2 \boldsymbol{v}^\top \boldsymbol{v}))$ for the importance sampler. A refinement of the sampling routine is necessary to also work well in more extreme cases. An example frequently encountered in practice is given when r_{obs} is rather large and at the same time lies very close to one truncation limit, yielding an insufficient amount of samples $r^b \in \mathcal{R}_y$ to approximate the truncated distribution well. We therefore propose a more efficient sampling routine, motivated by and applicable to selection procedures, for which the support of the truncated distribution is known to be a single interval $[\mathcal{V}^{lo}, \mathcal{V}^{up}]$. In this case, we do not even need to characterize the space empirically since the distribution of interest is known with the exception of the interval limits (the variance is assumed to be known and the null distribution determines the mean ρ). By employing a line search, we can find $\mathcal{V}^{lo}, \mathcal{V}^{up}$ and conduct inference based on the truncated normal distribution function $F^{[\mathcal{V}^{lo}, \mathcal{V}^{up}]}_{\rho, \sigma^2 \boldsymbol{v}^\top \boldsymbol{v}}(\cdot)$. We use such a corresponding line search here to

refine the importance sampling. By searching through the space of potential values $R \in \mathcal{R}_y$, a preliminary interval $[\tilde{R}^{lo}, \tilde{R}^{up}]$ covering $\mathcal{R}_y$ can be found with negligible computational cost by, e.g., successively checking extreme quantiles of Π for their congruency with respect to $\mathcal{R}_y$. By checking extremely small and large values of R and defining $\tilde{R}^{lo}$, $\tilde{R}^{up}$ such that both limits include all values, for which $R \in \mathcal{R}_y$, we can find a superset of the support of R up to numerical precision. We then draw from a uniform distribution with support $[\tilde{R}^{lo}, \tilde{R}^{up}]$. In comparison to the approach, which simply draws samples from Π, finding preliminary truncation limits $[\tilde{R}^{lo}, \tilde{R}^{up}]$ to refine the sampling space prior to the actual sampling proves to notably enhance accuracy and efficiency due the increased amount of accepted samples.

4.4 Further extensions

The ideas of section 4.2 and 4.3 can be extended to allow for computations in further relevant settings. An obvious extension is that to groups of variables. analogous to Yang et al. (2016). We additionally discuss four practically important extensions in the following.

Inference for groups of variables. In order to test groups of variables, the approach by Yang et al. (2016) described in Subsection 3.2 can almost directly be applied. To this end, we define $\mathcal{S}$ based on the set of chosen variables and use the sampling approach proposed in Subsection 4.3 for the χ-distribution on $\mathbb{R}^+$, such that $\tilde{R}^{lo} \geq 0$.

Incorporating cross-validation and other sub-sampling techniques. One of the most common ways to choose a final stopping iteration for the boosting algorithm is by using a resampling technique such as k-fold cross-validation and estimating the prediction error of the model in each step. By choosing the model with the smallest estimated prediction error, we again exploit information from the data, which we have to discard in the following inference. For the sampling approach described in 4.3 the extension is straightforward as we simply incorporate the cross-validation conditions in the space definition of $\mathcal{R}_y$. In order to check the congruency with the selection event $\mathcal{R}_y$, we keep the folds fixed and identical to the original fit when rerunning the algorithm with a new sample $\boldsymbol{y}^b$. In fact, this approach is

not only restricted to resampling methods. Stability selection (Shah & Samworth 2013) or other possibilities to choose an "optimal" number of iterations, as for example, by selection criteria such as the Akaike Information Criterion (AIC, Akaike 1974) can be incorporated into the inference framework in the same manner.

Unknown error variance. If the true error variance is unknown, we may use a consistent estimator instead. Judging by our simulation results, the effect of plugging in the empirical variance of the boosting model residuals is negligible in many cases and may also be a better (less anti-conservative) choice than the analogous estimator given by ordinary least squares estimation in the selected model due to the shrinkage effect. In cases with smaller signal-to-noise ratio, however, the plug-in approach may also yield invalid p-values under the null as shown in our simulation section. Tibshirani et al. (2015) present a plug-in as well as a bootstrap version of the test statistic, which yield asymptotically conservative p-values for $\boldsymbol{v}^\top \boldsymbol{\mu} = 0$. The bootstrap approach, however, can only be conducted efficiently if truncation limits of the test statistic are known. In the simulation section, we investigate the first suggestion by Tibshirani et al. (2015) – using the empirical variance of $\boldsymbol{y}$ as a conservative estimate for σ^2 – which better suits the presented framework.

Smooth effects. The given approach can also be used for additive models when the linear predictor $\eta_i = \boldsymbol{x}_i^\top \boldsymbol{\beta}$ in the working model $y_i = \eta_i + \varepsilon_i, i = 1, \dots, n$ is extended by additive terms of the form $g(c_i)$ for some covariate $\boldsymbol{c} = (c_1, \dots, c_n)^\top$. For the ease of presentation, we assume only one covariate $\boldsymbol{c}$ that is incorporated as an additive term. We therefore use a basis representation $g(c_i) = \boldsymbol{B}(c_i)\boldsymbol{\gamma} = \sum_{\varpi=1}^{M} \boldsymbol{B}_\varpi(c_i)\gamma_\varpi$ with M basis function $B_\varpi(\cdot)$ evaluated at the observed value c_i, basis coefficients γ_ϖ, $\boldsymbol{B}(c_i) = (B_1(c_i), \dots, B_M(c_i))$ and $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_M)^\top$. When $\boldsymbol{X}_\mathcal{A}$ is the composed matrix of all covariates, which are assumed to have a linear effect, and of the evaluated basis functions $\widetilde{\boldsymbol{B}} = (\boldsymbol{B}(c_1)^\top, \dots, \boldsymbol{B}(c_n)^\top)^\top$ of $\boldsymbol{c}$, we again might be interested in testing the best linear approximation of $\boldsymbol{\mu}$ in the space spanned by a given design matrix $\boldsymbol{X}_\mathcal{A}$. To this end, we can perform a point-wise test $H_0 : \mathfrak{g}(c) = 0$ for $\mathfrak{g}$ the true function in the basis space resulting from the best linear approximation of $\boldsymbol{\mu}$ by the given model. This can be done by using the proposed framework

with test vector $\boldsymbol{v}^\top = \boldsymbol{B}^0(c)(\boldsymbol{X}_\mathcal{A}^\top \boldsymbol{X}_\mathcal{A})^{-1}\boldsymbol{X}_\mathcal{A}^\top$ as $\mathfrak{g}(c) = \boldsymbol{v}\top\boldsymbol{\mu}$, where $\boldsymbol{B}^0(c)$ is defined as $\boldsymbol{X}_\mathcal{A}$ for which all columns but those corresponding to $\boldsymbol{B}(c)$ are set to zero. Instead of a pointwise test, the whole function can be tested

$$H_0 : \mathfrak{g}(\cdot) \equiv \mathbf{0} \tag{10}$$

by regarding the columns in $\widetilde{\boldsymbol{B}}$ as groups of variables and setting $\boldsymbol{W}$ in (3) to $\boldsymbol{P}^\perp_{\boldsymbol{X}_{\mathcal{A}\setminus j}}\widetilde{\boldsymbol{B}}$, where $\boldsymbol{X}_{\mathcal{A}\setminus j}$ denotes $\boldsymbol{X}_\mathcal{A}$ without the p_j columns of $\widetilde{\boldsymbol{B}}$.

The proposed tests and testvectors can also be used when smooth effects are estimated using a penalized base-learner.

5 Simulations

We now provide evidence for the validity of our method for linear and spline base-learners based on $B = 1000$ samples. We also show the performance of the proposed method in comparison to the polyhedron approach in a relevant setting and investigate the effect of different variance values. For linear regression with linear base-learners the true underlying model is given by

$$y_i = \eta_i + \varepsilon_i = \boldsymbol{X}_{[i,1:4]}\boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \ldots, n, \tag{11}$$

where $\boldsymbol{\beta} = (4, -3, 2, -1)$, $\varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$ with σ defined such that the signal-to-noise ration SNR $\in \{1, 4\}$ and $[i, 1:4]$ indicates the rows and columns of $\boldsymbol{X}$, respectively. We construct four linear base-learners for the four covariates $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_4$ in $\boldsymbol{X}_{[,1:4]}$ and additionally build $p_0 \in \{4, 22\}$ base-learners based on noise variables for $n \in \{25, 100\}$ observations, where the columns in $\boldsymbol{X}$ are independently drawn from a standard normal distribution (empirical correlations range from -0.53 to 0.48). Figure 1 shows the observed p-values versus the expected quantiles of the standard uniform distribution for settings, in which either the true model or a model larger than the true model with all four signal variables is selected. This corresponds to selection events, in which the null hypothesis (1) holds for $j > 4$ and thus p-values of inactive variables should exhibit uniformity given the selection event $\mathcal{A}$. The mixture of uniform $U[0, 1]$ p-values when aggregating across selected models again results in $U[0, 1]$ p-values.

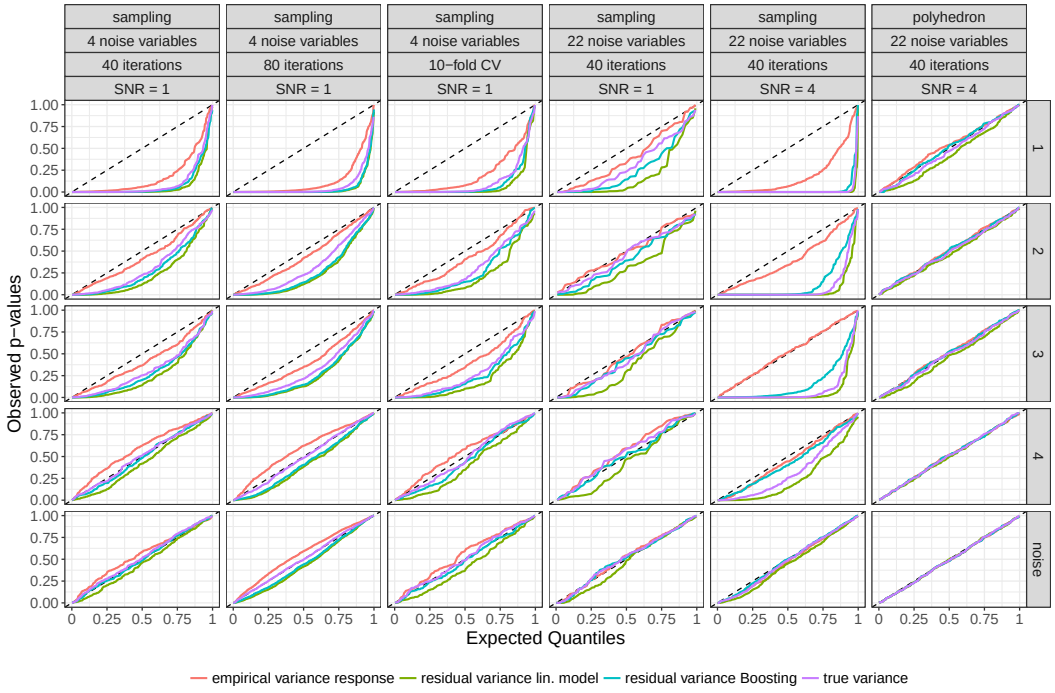

Figure 1: Observed p-values vs. expected quantiles across different covariates (rows) as well as different number of boosting iterations, number of noise variables and iterations, SNR and methods (columns) after boosting with a step-length of 0.1 using different variance types (colors), and a total of 1000 simulation iterations in settings with n = 25. p-values are shown for simulation iterations, in which either the true model or a model larger than the true model is selected.

Results: p-values for effects of "true effect" variables show deviations from the angle bisecting line, indicating the ability of the proposed procedure to correctly infer about the significance of the effects. The power decreases for a smaller number of observations (not shown), a smaller SNR and a larger number of noise variables. The polyhedron approach yields correct p-values under the null, but shows undesirable properties for non-noise variables. p-values for the proposed approach are uniform under the null when using the true variance, with more conservative results when using the empirical variance of the response and slightly non-uniform p-values when using a plugin estimator. In this respect, the em-

pirical variance of boosting residuals is more favorable than that of an OLS refit, but can also lead to deviations. However, note that the empirical approximation of p-values is not very accurate in the settings where specific selection events are rather unlikely, as only a small number of samples $r^b \in \mathcal{R}_y$ can be used. This, in particular, is the case for the setting with $m_{\text{stop}} = 150$ iterations and 26 covariates, where the selection probabilities for each path are rather small due to the large number of possible paths, and this may be the reason for deviations from the angle bisecting line for noise variables. Furthermore, corresponding confidence intervals of the proposed test procedure reveal approximately $1 - \alpha\%$ coverage. Results for $\alpha = 0.05$ are given in Table 1. Deviations from the ideal coverage of 95% are primarily due to numerical imprecision when inverting the hypothesis test and more accurate results can be obtained by increasing the number of samples B.

	p_0, number of iterations, SNR				
	4, 40, 1	4, 80, 1	4, CV, 1	22, 40, 1	22, 40, 4
coverage noise variables	0.9566	0.9571	0.9618	0.9485	0.9211
coverage signal variables	0.9699	0.9559	0.9326	0.9444	0.9429

Table 1: Estimated coverage of selective confidence intervals obtained by the proposed sampling approach for $n = 25$ observations when using the true variance in different settings (columns).

In the supplementary material, we additionally provide results for other settings of the previous simulation study as well as results for additive models using spline base-learners, where the true underlying function is given by $y_i = \sin(2X_{[i,1]}) + \frac{1}{2}X^2_{[i,2]} + \varepsilon_i, i = 1, \ldots, 300$, $\varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$ with σ defined such that the signal-to-noise ration SNR $\in \{0.5, 1\}$ and 13 further covariates $\boldsymbol{X}_{[,3:15]}$. All covariate effects are represented using penalized B-splines (P-spline; Eilers & Marx 1996) with B-Spline basis of degree 3, 5 knots and second order differences penalty. Tests for the whole function are performed as proposed in (10). Results suggest very high power and uniformity of p-values for noise variables, supporting the conclusion that the proposed test also works well for additive terms.

6 Application

We now apply our framework to the *prostate cancer data set* (Stamey et al. 1989) to model logarithmic PSA level (lpsa) of patients having prostate cancer. This data set has already been analyzed with regard to post-selection inference by, for example, Tibshirani et al. (2016) using forward stepwise regression and testing after a prespecified number of steps. In contrast to previous approaches, we do not enforce effects of continuous covariates to be linear but assume a more flexible additive model

$$lpsa_i = \beta_0 + \sum_{j=1}^{7} g_j(X_{[i,j]}) + \sum_{j=1}^{4} I(gleason_i = j)\beta_j + \varepsilon_i, \quad i = 1, \ldots, 97,$$

with 7 metric variables $\boldsymbol{X}_j, j = 1, \ldots, 7$ and categorical variable *gleason*. In order to estimate the smooth effects, we fit the model using P-spline base-learners with difference penalties. To facilitate a fair base-learner selection (Hofner et al. 2011), we split up effects of continuous covariates into a linear effect and a non-linear deviation from the corresponding linear effect. The optimal stopping iteration $m_{\text{stop}} = 47$ for the boosting algorithm with step-length $\nu = 0.1$ is found by using 10-fold cross-validation, which is incorporated into the selection mechanism $\mathcal{S}$. After 47 iterations, five effects are selected by the boosting procedure, including two non-linear deviations for the covariate *lbph* (logarithmic benign prostatic hyperplasia amount) and the covariate *pgg*45 (percentage Gleason scores 4 or 5). The two covariates reveal a U-shaped effect, which is shown in the supplementary material. The following table shows the results for componentwise tests of linear and additive terms for hypothesis tests based on the proposed sampling approach with 5000 samples. Testing additive terms, which have been split up into a linear part and a non-linear deviation, can be done by defining $\boldsymbol{B}$ as concatenated matrix of the covariate vector itself and the corresponding matrix of evaluated basis functions orthogonalized to the linear effect. The logarithmic cancer volume (*lvacol*) is found to be the only variable having a significant influence on the response for the given model.

	lbph (NL)	pgg45 (NL)	lcavol (L)	lweight (L)	svi (L)
magnitude	2.3319	2.8518	4.0992	2.2067	1.9520
lower limit	0	0	2.4859	0	0
p-value	0.3452	0.2467	0.0004	0.3752	0.1212

Table 2: Magnitude of linear (L) and non-linear (NL) projections $||\boldsymbol{P_W y}||_2$ for the selected model terms *lbph* (logarithmic benign prostatic hyperplasia amount), *pgg*45 (percentage Gleason scores 4 or 5), *lcavol* (logarithmic cancer volume), *lweight* (logarithmic prostate weight) and *svi* (seminal vesicle invasion) as well as corresponding lower confidence interval limits and p-values.

7 Discussion

In this paper we review several recently proposed selective inference frameworks and transfer and adapt them to the L_2-Boosting algorithm. As far as we know, there are no previous general methods available to quantify uncertainty of boosting estimates in a classical statistical manner when variable selection is performed. We propose tests and confidence intervals for linear base-learners as well as for group variable and penalized base-learners. We apply our framework to the prostate cancer data set and in contrast to published analyses of this data also allow for non-linear partial effects. Using simulation studies with a range of settings, we verify the properties of our approach.

This work opens up a variety of future research topics, including a mathematical description of the space restriction of test statistics given by the boosting algorithm.

An extension to generalized linear models (GLMs) and beyond, however, proves to be difficult since conditions involving $\boldsymbol{y}$ might imply conditioning on $\boldsymbol{y}$ itself if the response is discrete (see Fithian et al. 2014, for more details on selective inference for GLMs). It would also be interesting to investigate whether asymptotic results of Tian & Taylor (2017) can be used to construct inference for CFGD algorithms other than L_2-Boosting.

References

Akaike, H. (1974), 'A new look at the statistical model identification', *IEEE Transactions on Automatic Control* **19**(6), 716–723.

Berk, R., Brown, L., Buja, A., Zhang, K., Zhao, L. et al. (2013), 'Valid post-selection inference', *The Annals of Statistics* **41**(2), 802–837.

Brockhaus, S., Fuest, A., Mayr, A. & Greven, S. (2018), 'Signal regression models for location, scale and shape with an application to stock returns', *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **67**(3), 665–686.

Brockhaus, S., Scheipl, F., Hothorn, T. & Greven, S. (2015), 'The functional linear array model', *Statistical Modelling* **15**(3), 279–300.

Bühlmann, P. & Yu, B. (2003), 'Boosting with the l_2 loss: regression and classification', *Journal of the American Statistical Association* **98**(462), 324–339.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R. et al. (2004), 'Least angle regression', *The Annals of Statistics* **32**(2), 407–499.

Eilers, P. H. C. & Marx, B. D. (1996), 'Flexible smoothing with B-splines and penalties', *Statistical Science* **11**(2), 89–121.

Fithian, W., Sun, D. & Taylor, J. (2014), 'Optimal Inference After Model Selection', *arXiv e-prints arXiv:1410.2597* .

Friedman, J. H. (2001), 'Greedy function approximation: a gradient boosting machine', *Annals of statistics* pp. 1189–1232.

Hofner, B., Hothorn, T., Kneib, T. & Schmid, M. (2011), 'A framework for unbiased model selection based on boosting', *Journal of Computational and Graphical Statistics* **20**(4), 956–971.

Hothorn, T., Bühlmann, P., Kneib, T., Schmid, M. & Hofner, B. (2010), 'Model-based boosting 2.0', *Journal of Machine Learning Research* **11**(Aug), 2109–2113.

Lee, J. D., Sun, D. L., Sun, Y. & Taylor, J. E. (2016), 'Exact post-selection inference, with application to the lasso', *The Annals of Statistics* **44**(3), 907–927.

Loftus, J. R. & Taylor, J. E. (2014), 'A significance test for forward stepwise model selection', *arXiv e-prints arXiv:1405.3920* .

Loftus, J. R. & Taylor, J. E. (2015), 'Selective inference in regression models with groups of variables', *arXiv e-prints arXiv:1511.01478* .

Luo, Y. & Spindler, M. (2017), 'L2-boosting for economic applications', *American Economic Review* **107**(5).

Mayr, A., Hofner, B., Waldmann, E., Hepp, T., Meyer, S. & Gefeller, O. (2017), 'The evolution of boosting algorithms', *Computational and Mathematical Methods in Medicine* **2017**.

Melcher, M., Scharl, T., Luchner, M., Striedner, G. & Leisch, F. (2017), 'Boosted structured additive regression for escherichia coli fed-batch fermentation modeling', *Biotechnology and Bioengineering* **114**(2), 321–334.

Rügamer, D., Brockhaus, S., Gentsch, K., Scherer, K. & Greven, S. (2018), 'Boosting factor-specific functional historical models for the detection of synchronization in bioelectrical signals', *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **67**(3), 621–642.

Rügamer, D. & Greven, S. (2018), 'Selective inference after likelihood- or test-based model selection in linear models', *Statistics and Probability Letters* . To appear.

Shah, R. D. & Samworth, R. J. (2013), 'Variable selection with error control: Another look at stability selection', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **75**(1), 55–80.

Stamey, T. A., Kabalin, J. N., Mcneal, J. E., Johnstone, I. M., Freiha, F., Redwine, E. A. & Yang, N. (1989), 'Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. ii. radical prostatectomy treated patients', *The Journal of Urology* **141**(5), 1076 – 1083.

Tian, X. & Taylor, J. (2017), 'Asymptotics of selective inference', *Scandinavian Journal of Statistics* **44**(2), 480–499.

Tibshirani, R. J., Rinaldo, A., Tibshirani, R. & Wasserman, L. (2015), 'Uniform Asymptotic Inference and the Bootstrap After Model Selection', *arXiv e-prints arXiv:1506.06266* .

Tibshirani, R. J., Taylor, J., Lockhart, R. & Tibshirani, R. (2016), 'Exact post-selection inference for sequential regression procedures', *Journal of the American Statistical Association* **111**(514), 600–620.

Wasserman, L. & Roeder, K. (2009), 'High dimensional variable selection', *The Annals of Statistics* **37**(5A), 2178–2201.

Yang, F., Barber, R. F., Jain, P. & Lafferty, J. (2016), Selective inference for group-sparse linear models, *in* 'Advances in Neural Information Processing Systems', pp. 2469–2477.

Yuan, M. & Lin, Y. (2006), 'Model selection and estimation in regression with grouped variables', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68**(1), 49–67.

Part II

Model Estimation, Model Choice and Uncertainty Quantification in Function-on-Function Regression Models

Chapter 5

Boosting factor-specific functional historical models for the detection of synchronization in bioelectrical signals

Chapter 5 introduces extensions of functional historical models to models including random historical effects, factor-specific historical effects, and factor-specific random historical effects. The proposed methodology is motivated by research questions in the field of cognitive affective neuroscience and is used to analyse the functional relationship between electroencephalography and facial electromyography signals. The presented method is further investigated numerically in simulation studies.

Contributing article:

Rügamer, D., Brockhaus, S., Gentsch, K., Scherer, K., and Greven, S. (2018). Boosting factor-specific functional historical models for the detection of synchronization in bioelectrical signals. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 67(3):621–642.

Author contributions:

David Rügamer prepared a first draft with simulation studies as well as application to the emotion component data. Based on comments and valuable input of Sonja Greven and Sarah Brockhaus the methodological and simulation sections have been revised. Kornelia Gentsch and Klaus Scherer added valuable input to the application section as well as to the description of the emotion component data. All authors proofread the manuscript.

Supplementary material available at:

https://rss.onlinelibrary.wiley.com/doi/full/10.1111/rssc.12241

ROYAL STATISTICAL SOCIETY
DATA | EVIDENCE | DECISIONS

Journal of the Royal Statistical Society
Applied Statistics
Series C

Appl. Statist. (2018)
67, *Part* 3, *pp.* 621–642

Boosting factor-specific functional historical models for the detection of synchronization in bioelectrical signals

David Rügamer and Sarah Brockhaus,

Ludwig-Maximilians-Universität, Munich, Germany

Kornelia Gentsch and Klaus Scherer

University of Geneva, Switzerland

and Sonja Greven

Ludwig-Maximilians-Universität, Munich, Germany

[Received September 2016. Final revision August 2017]

Summary. The link between different psychophysiological measures during emotion episodes is not well understood. To analyse the functional relationship between electroencephalography and facial electromyography, we apply historical function-on-function regression models to electroencephalography and electromyography data that were simultaneously recorded from 24 participants while they were playing a computerized gambling task. Given the complexity of the data structure for this application, we extend simple functional historical models to models including random historical effects, factor-specific historical effects and factor-specific random historical effects. Estimation is conducted by a componentwise gradient boosting algorithm, which scales well to large data sets and complex models.

Keywords: Factor-specific functional historical effect; Functional data analysis; Function-on-function regression; Gradient boosting; Signal synchronization

1. Introduction

Bioelectrical signals such as electromyography (EMG), electroencephalography (EEG) or electrocardiogram are variations in electrical energy that carry information about living systems (Semmlow and Griffel, 2014). An appropriate analysis of bioelectrical signals, which are usually obtained in the form of time series data, is a crucial point in many research areas, including (tele-)medicine, automotive technology and psychology (Kang *et al.*, 2006; Kaniusas, 2012). In the field of cognitive affective neuroscience, a particular interest lies in the link of measured brain activity recorded with EEG and peripheral response systems such as the heart rate or facial muscle activity. In this context, our motivating study (Gentsch *et al.*, 2014) investigated the coherence between emotion components. In componential emotion theory, an emotional episode is thought to be an emergence of coherent or temporally correlated changes in emotion components, such as appraisals or facial expressions. This is referred to as synchronization (Grandjean and Scherer, 2009).

Address for correspondence: David Rügamer, Fachbereich Statistik, Ludwig-Maximilians-Universität, Ludwigstraße 33, Munich 80539, Germany.
E-mail: david.ruegamer@stat.uni-muenchen.de

 0035–9254/18/67621

1.1. The emotion components data

In the study of Gentsch *et al*. (2014), brain activity (EEG) as well as facial muscle activity (EMG) were simultaneously recorded. The data set at hand consists of time series of 384 equidistant observed time points for both EEG and EMG signals, eight different study settings (conditions in a computerized gambling game) and 24 participants. The traditional approach of analysing EEG and EMG data is to calculate the average signal for each participant across all trials of one study setting. For EEG data, this is referred to as event-related potential analysis (see, for example, Pfurtscheller and da Silva (1999)). Such an aggregation yields a reduced data set of $N = 8 \times 24 \times 384 = 73\,728$ observed data points. At each of the N time points, measurements are available for three EMG and 64 EEG electrodes. Fig. 1 depicts one EEG and EMG signal for one participant and all eight study settings with a common starting point of 200 ms after stimulus onset.

Efferent signals from the brain (signals originating from the brain) innervate or activate facial muscles (see, for example, Rinn (1984)). Therefore, it should be possible to trace back facial muscle activity recorded with facial EMG to brain activity captured with EEG. As certain cognitive processes can be related to different brain areas and facial regions, our particular interest lies in investigating the link between a selected EEG electrode signal and a specific EMG signal. We expect any association between these two signals

(a) to be time varying,
(b) to exhibit a temporal lag that is *a priori* unknown (even though a minimum lag can be inferred from the literature),
(c) to be specific to a study setting and/or
(d) to be present only during certain time intervals.

1.2. Existing methods for detecting synchronization

Previous approaches to detect synchrony in brain activity and autonomic physiology data have

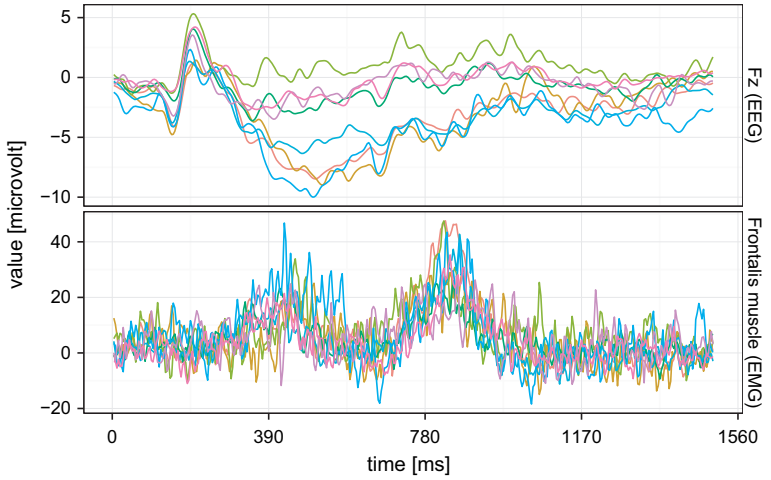

Fig. 1. Example for one EEG signal (Fz-electrode) and one EMG signal (*frontalis* muscle for raising eyebrows) of one participant, averaged over all trials for each of the eight possible game conditions; ——, high control, high power, gain; ——, low control, high power, gain; ——, high control, high power, loss; ——, low control, high power, loss; ——, high control, low power, gain; ——, low control, low power, gain; ——, high control, low power, loss; ——, low control, low power, loss

mostly focused on coherence or cross-correlation. Examinations of EEG and EMG synchronization can, *inter alia*, be found in Hollenstein and Crowell (2014), Mima and Hallett (1999), Brown (2000), Mima *et al*. (2000a, b), Grosse *et al*. (2002), Quiroga *et al*. (2002), Bortel and Sovka (2006) and Hashimoto *et al*. (2010). Whereas coherence is a function of the frequency measuring the explained variance of one time series by another time series in the frequency domain, cross-correlation is a function of time, yielding the correlation of two time series for a given lag (see, for example, Pawitan (2005)). With the aim to relate different time points of two signals to each other, we focus on methods in the time domain. Established methods are, however, concerned with the estimation of the association between two observed time series rather than the analysis of a large number of time series observations given in pairs of signals. This applies to (cross-)correlation, which additionally does not provide the possibility of taking covariates into account, as well as for other methods such as the *generalized synchronization* approach based on the state space representation (Diab *et al*., 2013) or auto-regressive times series approaches (see for example Ozaki (2012)). Furthermore, most of these approaches require the definition of a specific or a maximum time lag.

1.3. Function-on-function regression

As both the EEG and the EMG signal can be understood as noisy observations of functional variables, function-on-function regression approaches offer another possibility to describe and infer the relationship of such time series (see Morris (2015) for a recent review). Function-on-function regression models adapt the principle of standard regression by allowing for a functional response as well as functional covariates. The so-called historical model (Malfait and Ramsay, 2003; Harezlak *et al*., 2007) is one possibility to explain a functional response $Y(t),\ t \in \mathcal{T} = [T_1, T_2]$ with $T_1, T_2 \in \mathbb{R}$, using a linear effect of the complete history of a functional covariate $X(s), s \in \mathcal{T}$:

$$\mathbb{E}\{Y(t)|X = x\} = \int_{T_1}^{t} x(s)\beta(s,t)\,\mathrm{d}s. \tag{1}$$

In contrast with the existing approaches that were discussed above, historical models enable us to relate a given time point of one time series to more than one time point in $[T_1, t]$ of another time series.

Early core work on functional historical models is limited to historical models with only one functional covariate. A multitude of application possibilities are conceivable and historical models have been used in different research areas including health and biological science (Malfait and Ramsay, 2003; Harezlak *et al*., 2007; Gervini, 2015; Brockhaus *et al*., 2017). Brockhaus *et al*. (2017) extended the framework of a simple historical model such as equation (1) to functional regression models with a high number of functional historical effects and potentially further covariate effects by utilizing gradient boosting for estimation.

Alternative estimation procedures for flexible function-on-function regression models including historical effects are based on a mixed model representation (see Scheipl *et al*. (2015)) or componentwise gradient boosting (see Brockhaus *et al*. (2015, 2017)) These are implemented in the `pffr` function of the R package `refund` (Huang *et al*., 2015) and in the R package `FDboost` (Brockhaus and Rügamer, 2016) respectively.

1.4. Proposed approach

To reflect the study design in this application, we extend functional historical models to historical effects that vary over one or two (penalized) categorical covariates to allow for subject-, setting- and subject-by-setting-specific effects. We provide mathematical concepts for the construction of

design matrices and penalty matrices as well as suitable identifiability constraints. We integrate these concepts into the framework of Brockhaus *et al.* (2015, 2017) and implement them for estimation via componentwise gradient boosting. We also speed up the estimation by making use of our particular model structure. By carrying out estimation with componentwise gradient boosting as in Brockhaus *et al.* (2015), our approach has several advantages. In particular, it can fit multiple factor- and subject- as well as subject-by-factor-specific functional effects, which is not possible in alternative approaches for function-on-function regression such as implemented in the `pffr` function in the R package `refund` (Huang *et al.*, 2015). Furthermore, the algorithm allows for different loss functions and thus covers models beyond mean regression (Kneib, 2013), e.g. median, robust or quantile regression. It can deal with high dimensional data sets that often go hand in hand with multisensor bioelectrical signal data collections, as well as settings with more covariates than observations. Our approach can find multimodal effect surfaces and band effects, thereby covering special cases of time series approaches. In addition, we derive options to reduce computation time as well as memory storage considerably and address the question of uncertainty in complex boosted models.

The remainder of this paper describes the model and method proposed in Section 2, presents the gradient boosting algorithm in Section 3 and covers a simulation study in Section 4. We apply boosted historical models to the emotion components data in Section 5 and conclude with a discussion in Section 6. Our proposed methods are implemented in the R package `FDboost`, which is an extension of the model-based boosting package `mboost` (Hothorn *et al.*, 2016). The R code for our simulation, as well as code and data for our application, is provided in an on-line repository (`https://github.com/davidruegamer/BoostingSignalSynchro`).

2. Functional response models and historical effects

After outlining the functional historical model in Section 2.1, we extend the model of Brockhaus *et al.* (2017) to models with functional historical terms interacting with categorical covariates and to random functional historical effects in Section 2.2.

2.1. Functional historical models

We focus on additive functional regression models of the form (Brockhaus *et al.*, 2015, 2017)

$$\xi\{Y(t)|\mathbf{X}=\mathbf{x}\}=h(\mathbf{x})(t)=\sum_{j=1}^{J}h_j(\mathbf{x})(t), \tag{2}$$

where ξ is a transformation function for the conditional distribution of the functional response $Y(t), t\in\mathcal{T}$. In our application ξ is equal to the conditional expectation $\mathbb{E}$, although it could also be for example the (pointwise) median or a quantile. The covariate set $\mathbf{x}$ comprises functional observations $x_1(\cdot),\ldots,x_{p_x}(\cdot)$ and scalar covariates $z_1,\ldots,z_{p_z}$ with $p:=p_x+p_z$. $h_j(\mathbf{x})(t)$ are partial effects, which can depend on scalar as well as on functional covariates. In particular, this general model class includes models with one or more historical effects

$$h_j(\mathbf{x})(t)=\int_{l(t)}^{u(t)}x_{k_j}(s)\beta_j(s,t)\,\mathrm{d}s, \tag{3}$$

$k_j\in\{1,\ldots,p_x\}$, which can have general integration limits $l(t)$ and $u(t)$, for example, defined by $l(t)=T_1$ and $u(t)=t$, $l(t)=t-\delta$ and $u(t)=t$ or partial histories $l(t)=t-\delta_l$ and $u(t)=t-\delta_u, t>\delta_l>\delta_u>0$ as in Harezlak *et al.* (2007). Functional historical effects are particularly suited to

settings where both response $Y(t)$ and covariates $X_{k_j}(s)$ are observed over the same time interval, $s, t \in \mathcal{T}$.

In practice, $x_{k_j}(\cdot)$ is observed on a grid $s_1, \ldots, s_R$ and the integral as well as the smooth coefficient surface $\beta_j(s,t)$ in equation (3) must be approximated. We use numerical integration and a tensor product spline basis expansion respectively. For $k = 1, \ldots, K_x, l = 1, \ldots, K_t$ define the basis functions $\Phi^s_{j,k}(s)$ and $\Phi^t_{j,l}(t)$ for the s- and the t-direction of the coefficient surface $\beta_j(s,t)$ respectively. Let $\theta_{j,k,l}$ be the corresponding basis coefficients and $\Delta(s_r)$ numerical integration weights for the observed time points s_r. Then, the historical effect can be represented by (Scheipl *et al.*, 2015; Brockhaus *et al.*, 2017)

$$\int_{l(t)}^{u(t)} x_{k_j}(s)\beta_j(s,t)\mathrm{d}s \approx \mathbf{B}_j(x_{k_j}, t)\boldsymbol{\theta}_j \tag{4}$$

with $\boldsymbol{\theta}_j = (\theta_{j,1,1}, \ldots, \theta_{j,K_x,K_t})^{\mathrm{T}}$, $\mathbf{B}_j(x_{k_j}, t) = \mathbf{B}^x_j(x_{k_j}, t) \otimes \mathbf{B}^t_j(t)$ by using the Kronecker product '$\otimes$' and by defining

$$\mathbf{B}^x_j(x_{k_j}, t) = \left(\sum_{r=1}^{R} \Delta(s_r) x_{k_j}(s_r, t) \Phi^s_{j,1}(s_r) \; \ldots \; \sum_{r=1}^{R} \Delta(s_r) x_{k_j}(s_r, t) \Phi^s_{j,K_x}(s_r) \right)$$

as well as $\mathbf{B}^t_j(t) = (\Phi^t_{j,1}(t) \ldots \Phi^t_{j,K_t}(t))$. Let $I(\cdot)$ be the indicator function. Following Scheipl *et al.* (2015), for n observed curves $x_{k_j,1}(\cdot), \ldots, x_{k_j,n}(\cdot)$ at grid points s_r, $x_{k_j}(s_r, t) = x_{k_j}(s_r)\, I\{l(t) \leqslant s_r \leqslant u(t)\}$ and response observations $y_i(t_{i,d})$ at potentially curve-specific time points $t_{i,d} \in \mathcal{T}$, $i = 1, \ldots, n$, $d = 1, \ldots, D_i$, $N = \Sigma_{i=1}^n D_i$, the design matrix of a historical effect can be summarized by

$$\mathcal{B}_j := \mathbf{B}^x_j \odot \mathbf{B}^t_j = (\mathbf{B}^x_j \otimes \mathbf{1}^{\mathrm{T}}_{K_t}) * (\mathbf{1}^{\mathrm{T}}_{K_x} \otimes \mathbf{B}^t_j), \tag{5}$$

where $\mathbf{B}^x_j \in \mathbb{R}^{N \times K_x}$ with rows $\mathbf{B}^x_j(x_{k_j,i}, t_{i,d})$, $\mathbf{B}^t_j \in \mathbb{R}^{N \times K_t}$ with rows $\mathbf{B}^t_j(t_{i,d})$, '$\odot$' is the rowwise tensor product, '$*$' the Hadamard product (elementwise matrix multiplication) and $\mathbf{1}^{\mathrm{T}}_a$ a row vector of length a. In the on-line supplemental material, we provide a simple example of how to interpret estimated coefficient surfaces of historical effects, as we believe that this is an important part in using historical models.

Regularization of the coefficient vector $\boldsymbol{\theta}_j$ in expression (4) is achieved by an anisotropic penalty. Using the marginal penalties $\mathbf{P}^x_j \in \mathbb{R}^{K_x \times K_x}$ and $\mathbf{P}^t_j \in \mathbb{R}^{K_t \times K_t}$ of the historical effect basis in the s- and t-direction respectively, a quadratic penalty term can be constructed as

$$\boldsymbol{\theta}^{\mathrm{T}}_j \mathcal{P}_j \boldsymbol{\theta}_j = \boldsymbol{\theta}^{\mathrm{T}}_j \{\lambda^x_j(\mathbf{P}^x_j \otimes \mathbf{I}_{K_t}) + \lambda^t_j(\mathbf{I}_{K_x} \otimes \mathbf{P}^t_j)\}\boldsymbol{\theta}_j = \boldsymbol{\theta}^{\mathrm{T}}_j (\lambda^x_j \mathbf{P}^x_j \oplus \lambda^t_j \mathbf{P}^t_j)\boldsymbol{\theta}_j, \tag{6}$$

where $\lambda^x_j, \lambda^t_j \geqslant 0$ are smoothing parameters and '$\oplus$' is the Kronecker sum (Wood, 2006; Scheipl *et al.*, 2015). More details on the penalization and potential extensions can be found in the next subsection. Similarly, penalized basis expansions like expressions (4)–(6) can also be constructed for a multitude of other effects of scalar and/or functional covariates, including all effects of scalar covariates in our proposed model for the emotion components data (Scheipl *et al.*, 2015; Brockhaus *et al.*, 2015).

In addition to ordinary historical effects, this approach can incorporate a time varying intercept $h_j(\mathbf{x})(t) = \alpha(t)$ as well as time varying categorical or random effects

$$h_j(\mathbf{x})(t) = \gamma_{j,e}(t)\, I(z_{q_j} = e), \tag{7}$$

where $q_j \in \{1, \ldots, p_z\}$, z_{q_j} is a categorical covariate with levels $e \in \{1, \ldots, \eta\}$ and $\gamma_{j,e}(t)$ the corresponding time varying coefficient. The smoothness of the coefficient functions $\alpha(t)$ and

$\gamma_{j,e}(t)$ is obtained with a spline basis representation such as expression (4) and a Kronecker sum penalty such as equation (6) with $\mathbf{P}_j^x$ set to zero for categorical effects and $\mathbf{P}_j^x = \mathbf{I}_{K_z}$ for (independent) functional random effects (see Brockhaus *et al*. (2015) for more details). In particular, for functional random effects, the quadratic penalty in equation (6) is equivalent to a normal distribution assumption on $\boldsymbol{\theta}_j$ with zero mean and covariance proportional to the generalized inverse of $\mathcal{P}_j$ (Brumback *et al*., 1999), inducing a Gaussian process assumption for the functional random effects. Furthermore, we consider interaction effects of z_{q_j} and a second categorical covariate $z_{q'_j}$ with levels $f = 1, \ldots, \varphi$ of the form

$$h_j(\mathbf{x})(t) = \rho_{j,e,f}(t)\, I(z_{q_j} = e)\, I(z_{q'_j} = f). \tag{8}$$

Identifiability constraints for time varying categorical effects such as equations (7) and (8) are discussed in the following subsection.

2.2. Factor-specific historical effects

In light of our application, we newly introduce factor-specific historical effects for functional regression models. Factor-specific historical effects can be useful when historical effects are assumed to vary, e.g. between different study settings or subjects. First, consider a categorical covariate z_{q_j} with levels $e = 1, 2, \ldots, \eta$ and a functional covariate $x_{k_j}(s)$, which is modelled via a historical effect. A simple additive model of the form (2) would then include a main historical effect (3) and a factor-specific historical effect

$$h_j(\mathbf{x})(t) = I(z_{q_j} = e) \int_{l(t)}^{u(t)} x_{k_j}(s)\beta_{j,e}(s,t)\,\mathrm{d}s. \tag{9}$$

Given a total of N observations and the covariate vector $\mathbf{z}_{q_j} = ((z_{q_j,1} \otimes \mathbf{1}_{D_1})^{\mathrm{T}}, \ldots, (z_{q_j,n} \otimes \mathbf{1}_{D_n})^{\mathrm{T}})^{\mathrm{T}}$ the factor-specific historical effect is constructed similarly to equation (5). The design matrix is extended to

$$\mathcal{B}_j = \mathbf{B}_j^z(\mathbf{z}_{q_j}) \odot \mathbf{B}_j^x \odot \mathbf{B}_j^t = \widetilde{\mathbf{B}}_j^x \odot \mathbf{B}_j^t, \tag{10}$$

where $\mathbf{B}_j^z(\mathbf{z}_{q_j})$ is a design matrix for the factor variable depending on the constraints on $\beta_{j,e}(s,t)$ (see below) and $\widetilde{\mathbf{B}}_j^x = \mathbf{B}_j^z(\mathbf{z}_{q_j}) \odot \mathbf{B}_j^x$. An important special case is given for the unconstrained estimation of $\beta_{j,e}$ when the observations are sorted by the factor levels $e = 1, \ldots, \eta$. This yields a block diagonal incidence matrix for $\mathbf{B}_j^z(\mathbf{z}_{q_j}) = \mathrm{diag}(\mathbf{1}_{\kappa_1}, \mathbf{1}_{\kappa_2}, \ldots, \mathbf{1}_{\kappa_\eta}) \in \mathbb{R}^{N \times \eta}$ and an $N \times \eta K_x$ block diagonal matrix for $\widetilde{\mathbf{B}}_j^x = \mathrm{diag}(\mathbf{B}_{j,1}^x, \ldots, \mathbf{B}_{j,\eta}^x)$. Here, $\mathbf{B}_{j,e}^x \in \mathbb{R}^{\kappa_e \times K_x}$ contains the rows $\Sigma_{k=1}^{e-1}\kappa_k + 1, \ldots, \Sigma_{k=1}^{e}\kappa_k$ of $\mathbf{B}_j^x$ corresponding to all rows with factor level e and κ_e being the total number of observation points for factor level e. This special structure can be exploited for a more efficient computational implementation (see Section 3.2 for more details).

When the historical effect of x_{k_j} is not only factor or subject specific, but varies for a categorical covariate z_{q_j} with levels $e = 1, 2, \ldots, \eta$ as well as for subjects $z_{q'_j}$ with levels $f = 1, 2, \ldots, \varphi$, we let

$$h_j(\mathbf{x})(t) = I(z_{q_j} = e)\, I(z_{q'_j} = f) \int_{l(t)}^{u(t)} x_{k_j}(s)\beta_{j,e,f}(s,t)\,\mathrm{d}s. \tag{11}$$

The design matrix for the random factor-specific historical effect or *doubly varying historical effect* (11) is then defined by extending $\mathbf{B}_j^z(\mathbf{z}_{q_j})$ in equation (10) to

$$\mathbf{B}_j^z(\mathbf{z}_{q_j}, \mathbf{z}_{q'_j}) = \mathbf{B}_j^z(\mathbf{z}_{q_j}) \odot \mathbf{B}_j^z(\mathbf{z}_{q'_j}).$$

For these factor-specific historical effects (9) and (11), we must carefully consider their identifiability and regularization.

2.2.1. Identifiability constraints

To ensure that the main historical effect is separable from the factor-specific historical effects and vice versa, we impose the following constraint when both are included in the model:

$$\sum_{e=1}^{\eta} \psi_e \, \beta_{j,e}(s,t) = 0 \qquad \forall\, t \in \mathcal{T}, s \in [l(t), u(t)], \tag{12}$$

where ψ_e are weights for each level $e = 1, \ldots, \eta$ of the factor variable. Specifically, for observed curves $i = 1, \ldots, n$, we use $\psi_e = \Sigma_{i=1}^{n} I(z_{q_j,i} = e)$, which coincides with equal weighting in the case of balanced factor levels. This also enables $\beta_j(s,t)$ in equation (9) to be interpretable as average historical effects over the η subgroups. Constraint (12) ensures identifiability because the factor-specific historical effects are centred near the surface of the main effect for models including both equation (3) and equation (9).

For the doubly varying historical effects to be defined as deviations from both factor-specific historical effects, we impose the constraints

$$\sum_{e=1}^{\eta} \psi_{e,f} \, \beta_{j,e,f}(s,t) = 0 \qquad \forall t \in \mathcal{T}, s \in [l(t), u(t)],\, f \in \{1, \ldots, \varphi\} \tag{13}$$

and

$$\sum_{f=1}^{\varphi} \psi_{e,f} \, \beta_{j,e,f}(s,t) = 0 \qquad \forall t \in \mathcal{T}, s \in [l(t), u(t)], e \in \{1, \ldots, \eta\}, \tag{14}$$

for which we use the weights $\psi_{e,f} = \Sigma_{i=1}^{n} I(z_{q_j,i} = e, z_{q'_j,i} = f)$.

To ensure identifiability and interpretability of the whole model, further constraints must be placed on effects other than the historical effects, i.e. when including time varying effects in the model. As in Scheipl *et al.* (2015) and Brockhaus *et al.* (2015), all time varying effects in our models are specified as deviations from the smooth intercept $\alpha(t)$. This ensures the identifiability of each effect and enables a meaningful interpretation (as deviation from the sample mean $\alpha(t)$). Consider the factor variable z_{q_j} and an effect as in equation (7). We then impose $\Sigma_{e=1}^{\eta} \psi_e \gamma_{j,e}(t) = 0 \; \forall\, t \in \mathcal{T}$. A similar constraint is enforced for interaction effects (8) with coefficients $\rho_{j,e,f}(t)$: $\Sigma_{e=1}^{\eta} \psi_{e,f} \, \rho_{j,e,f}(t) = 0 \; \forall\, t \in \mathcal{T}, f \in \{1, \ldots, \varphi\}$ and $\Sigma_{f=1}^{\varphi} \psi_{e,f} \, \rho_{j,e,f}(t) = 0 \; \forall\, t \in \mathcal{T}, e \in \{1, \ldots, \eta\}$, i.e. each interaction effect must be centred near its corresponding main effects. For details on the implementation, see section B in the on-line supplementary material.

2.2.2. Parameterization

The separation of the factor-specific historical effect and the corresponding main historical effect together with constraint (12) is particularly useful in the light of model selection. However, an alternative model formulation that does not separate main and factor-specific historical effects may sometimes be beneficial for the interpretation of estimated effects and the simplicity of the model definition. A historical model with a main and factor-specific historical effects can be rewritten as $\int_{l(t)}^{u(t)} x_{k_j}(s)\{\beta_j(s,t) + I(z_{q_j} = e)\, \beta_{j,e}(s,t)\}\mathrm{d}s$, combining main and factor-specific historical effects by estimating the sum $\tilde{\beta}_{j,e}(s,t) := (\beta_j(s,t) + I(z_{q_j} = e)\, \beta_{j,e}(s,t))$ and thereby making constraint (12) obsolete.

2.2.3. Regularization

For the regularization of a factor-specific historical effect, the penalty depends on whether we want to regularize over the factor levels, e.g. for '*random historical effects*', or not, e.g. for study settings. In general, the quadratic penalty matrix in expression (6) is extended to an anisotropic penalty

$$\mathcal{P}_j = (\lambda_j^z \mathbf{P}_j^z \oplus (\lambda_j^x \mathbf{P}_j^x \oplus \lambda_j^t \mathbf{P}_j^t)), \tag{15}$$

where $\mathbf{P}_j^z$ is the $K_z \times K_z$ marginal penalty matrix over the factor levels and λ_j^x, λ_j^t and λ_j^z are the smoothing parameters controlling the regularization of the historical effect part in the s- as well as t-direction and of the factor variable part respectively. Usually, K_z is the number of factor levels (minus one, depending on the constraint on the effect) and $\mathbf{P}_j^z$ is a simple ridge penalty $\mathbf{P}_j^z = \mathbf{I}_{K_z}$. Whereas the factor-specific historical effect is therefore shrunk towards the main historical effect in a model with both main and factor-specific historical effect, the penalty in the alternative parameterization without constraint on the factor-specific historical effect enforces shrinkage of $\beta_{j,e}$ towards 0. In practice, the s- and t-directions of the historical effect are typically measured on the same scale (i.e. time); thus we introduce an isotropic penalty for the historical effect part by defining $\lambda_j^t \equiv \lambda_j^x =: \lambda_j^h$ and $\mathbf{P}_j^x \oplus \mathbf{P}_j^t =: \mathcal{P}_j^h$. For the doubly varying historical effect (11), the term $\lambda_j^z \mathbf{P}_j^z$ in equation (15) is replaced by $(\lambda_j^z \mathbf{P}_j^z \oplus \lambda_j^{z'} \mathbf{P}_j^{z'})$. If one or both factors are not penalized, the corresponding penalty matrices are set to 0.

3. Estimation: componentwise gradient boosting

The estimation via componentwise gradient boosting (Bühlmann and Hothorn, 2007; Brockhaus *et al.*, 2015) has several advantages. The main advantage of using componentwise boosting over conventional estimation procedures lies in the nature of componentwise fitting, as the feasibility of componentwise fitting procedures depends on the most complex individual component only. Adding partial effects step by step, boosting provides implicit variable selection and enables model estimation in settings with $J > n$ or $p > n$.

3.1. Componentwise gradient boosting

The componentwise gradient boosting algorithm for a function-on-function regression model was introduced by Brockhaus *et al.* (2015) and is based on the functional gradient descent (FGD) algorithm (see Bühlmann and Hothorn (2007) and Hothorn *et al.* (2016)).

3.1.1. Loss function and empirical risk

In general, the componentwise FGD algorithm aims to minimize the expected loss $\mathbb{E}_{(Y,\mathbf{X})}[\rho\{(Y,\mathbf{X}),h\}]$ for response Y and covariates $\mathbf{X}$ with respect to the additive predictor h for a suitable loss function ρ. The loss is determined by the underlying regression problem, e.g. the L^2-loss for mean regression. To adapt the principle of FGD to functional observations, the loss function l for a whole trajectory is defined as $l\{(Y,\mathbf{X}),h\} = \int_{\mathcal{T}} \rho\{(Y,\mathbf{X}),h\}(t)\mathrm{d}t$, i.e. the integrated pointwise loss ρ over the domain $\mathcal{T}$. For potentially functional observations $(y_i, \mathbf{x}_i)$, $i = 1, \ldots, n$, the objective function, the risk, is then given by $\mathbb{E}_{(Y,\mathbf{X})}[l\{(Y,\mathbf{X}),h\}]$ and the FGD algorithm for functional regression models aims at minimizing the empirical risk

$$n^{-1} \sum_{i=1}^{n} \sum_{d=1}^{D_i} w_i \Upsilon(t_{i,d}) \rho\{(y_i, \mathbf{x}_i), h\}(t_{i,d}),$$

where sampling weights w_i are used to select or deselect all observations of one functional trajectory in resampling approaches and $\Upsilon(t)$ are weights of a numerical integration scheme used to approximate the integrated loss l (Brockhaus *et al.*, 2017).

3.1.2. Routine and base learners

In each step, the FGD algorithm evaluates a set of *base learners* (in this case corresponding to penalized regression for the partial effects h_j), chooses the base learner that best fits the negative gradient at the current estimate $-\partial \mathbb{E}_{(Y,\mathbf{X})}[l\{(Y,\mathbf{X}),h\}]/\partial h$ and updates the fit in the light of this choice. As in representation (4), we assume that every base learner can be represented as a linear effect in $\boldsymbol{\theta}_j \in \mathbb{R}^{K_j}$, i.e. $h_j(\mathbf{x})(t) = \mathbf{B}_j(\mathbf{x}_{k_j}, t)\boldsymbol{\theta}_j$, with suitable penalty, e.g. expression (6) or (15).

3.1.3. Algorithm

The full algorithm is given by the following five steps.

Step 1: set $m = 0$; initialize the estimates, e.g. $\hat{\boldsymbol{\theta}}_j^{[m]} \equiv \mathbf{0}$ for each base learner $j \in \{1, \ldots, J\}$, and define $\hat{h}^{[m]}(\mathbf{x})(t) = \sum_{j=1}^{J} \mathbf{B}_j(\mathbf{x}, t)\hat{\boldsymbol{\theta}}_j^{[m]}$; choose a step length $\nu \in (0, 1]$ and a maximal stopping iteration m_{stop}.

Step 2: compute the negative gradient $-\partial\rho\{(y, \mathbf{x}), h\}/\partial h$ and define the so-called *pseudoresiduals*

$$u_i(t_{i,d}) := -\frac{\partial}{\partial h}\rho\{(y_i, \mathbf{x}_i), h\}(t_{i,d})\Big|_{h=\hat{h}^{[m]}}.$$

Step 3: fit the base learners $j = 1, \ldots, J$ to the pseudoresiduals

$$\hat{\boldsymbol{\vartheta}}_j = \underset{\boldsymbol{\vartheta} \in \mathbb{R}^{K_j}}{\arg\min} \sum_{i=1}^{n} \sum_{d=1}^{D_i} w_i \Upsilon(t_{i,d})\{u_i(t_{i,d}) - \mathbf{B}_j(\mathbf{x}_{k_j,i}, t_{i,d})\boldsymbol{\vartheta}\}^2 + \boldsymbol{\vartheta}^{\mathrm{T}}\mathcal{P}_j\boldsymbol{\vartheta}$$

and find the best-fitting j^*th base learner such that

$$j^* = \underset{j=1,\ldots,J}{\arg\min} \sum_{i=1}^{n} \sum_{d=1}^{D_i} w_i \Upsilon(t_{i,d})\{u_i(t_{i,d}) - \mathbf{B}_j(\mathbf{x}_{k_j,i}, t_{i,d})\hat{\boldsymbol{\vartheta}}_j\}^2.$$

Step 4: set $\hat{\boldsymbol{\theta}}_{j^*}^{[m+1]} = \hat{\boldsymbol{\theta}}_{j^*}^{[m]} + \nu\hat{\boldsymbol{\vartheta}}_{j^*}$ and $\hat{\boldsymbol{\theta}}_j^{[m+1]} = \hat{\boldsymbol{\theta}}_j^{[m]}\ \forall j \neq j^*$ and update $\hat{h}^{[m]}$ accordingly.

Step 5: set $m = m + 1$; as long as $m \leqslant m_{\text{stop}}$, repeat steps 2–5.

The final model with corresponding parameters $\hat{\boldsymbol{\theta}}_j^{m^*}$, $j = 1, \ldots, J$, $m^* \in \{1, \ldots, m_{\text{stop}}\}$, is chosen from the set of m_{stop} estimated models via cross-validation or other resampling methods on the level of curves (Brockhaus *et al.*, 2015) to prevent overfitting. This so-called early stopping of the boosting procedure introduces regularization on coefficient estimates (Zhang and Yu, 2005).

3.2. Unbiased base learner selection and smoothing parameter computation

It is important to set equal degrees of freedom df_j for every base learner j for a fair selection of base learners (Hofner *et al.*, 2011). A regularization over factor levels for categorical covariates with a moderate or large number of factor levels is thus often necessary in practice as df_j would otherwise become very large. The smoothing parameters λ_j, which have a one-to-one correspondence with df_j, must therefore be computed and fixed appropriately beforehand for $j = 1, \ldots, J$. Model complexity and smoothness are then controlled for fixed ν by the stopping iteration, which is chosen by resampling.

The `FDboost` package, which is based on the `mboost` package, uses the Demmler–Reinsch orthogonalization (see, for example, Ruppert *et al*. (2003)), which avoids repeated matrix inversions to find a suitable λ_j efficiently. Nonetheless, computing the Demmler–Reinsch orthogonalization may be very expensive, particularly for factor- and subject-specific historical effects, because of a singular value decomposition, and can take up to 99% of total computing time. To tackle this problem, on the one hand, we recommend reducing the number of knots for (doubly) varying historical effects to a small number (e.g. 4), if this is not expected to lead to unwanted oversmoothing. On the other hand, we exploit the model structure for factor-specific historical effects and derive a presentation that allows for a blockwise singular value decomposition with computation time of the order of an ordinary historical effect. This reduces overall computation time dramatically (see section C in the on-line supplementary material for more details). For the application in Section 5, for example, the most complex model with partially aggregated data could be fitted in under 16 min with less than 45 Gbytes of random-access memory, whereas the brute force method (fitting the model with 10 knots without exploitation of the model structure) failed, exceeding the memory limit of 1 Tbyte of random-access memory after running for more than 10 days. Although the first approach can be a good (approximate) *ad hoc* solution, the second approach is exact and thus generally recommended if feasible.

3.3. Quantification of uncertainty

Because of the large fluctuation in bioelectrical signals, a very important aspect in the analysis of such signals is the assessment and quantification of uncertainty. For the detection of synchronization with a large number of potentially relevant time intervals of both signals, 'significant' effects for specific time point combinations are of particular interest. Apart from rank-based p-values provided in the context of likelihood-based boosting (Binder *et al*., 2009) using permutations of the response, no general inferential framework in the classical statistical sense exists for boosting methods. An alternative approach is stability selection (Meinshausen and Bühlmann, 2010), which evaluates the importance of explanatory variables by looking at the stability of term selection under subsampling and has already been adapted for functional regression boosting (see, for example, Brockhaus *et al*. (2015)). In the emotion components application, however, the applied research question defines the chosen covariates and the statistical analysis needs to address the uncertainty of estimated coefficient surfaces. We therefore use a non-parametric curve level bootstrap to assess the variability of estimated effects. Because of the shrinkage effect of boosting, the corresponding bootstrap intervals are useful for quantification or variability of the regularized coefficients but are on average not centred at the true coefficient surface, unlike unbiased estimators. In consequence, the distribution of bootstrap estimates does not provide valid confidence intervals. In the following section, we investigate whether, despite the shrinkage effect, variability bands can be used at least to assess pointwise difference from zero. As simulation results suggest, these variability bands find most of the truly non-zero surface regions in all of our simulation settings.

4. Simulations

We provide results for the estimation performance of simple historical effects (Section 4.1), factor-specific historical effects (Section 4.2) and for the uncertainty quantification via the bootstrap (Section 4.3). In Section 4.4, we briefly address results on different parameterizations and boosting step lengths.

Similarly to our application, we use historical effects with integration limits $l(t) = T_1 = 0$ and $u(t) = t - \delta$ with $\delta = 0.025$. We compare the estimated surface with the underlying true function and, wherever possible, with an estimate by using a functional additive mixed model as implemented in the `pffr` function in R package `refund` (Scheipl *et al.*, 2015). Apart from visual comparisons, we estimate the relative integrated mean-squared error $reliMSE \int\int\{\hat{\beta}(s,t) - \beta(s,t)\}^2 \mathrm{d}s\,\mathrm{d}t \{\int\int \beta(s,t)^2 \mathrm{d}s\,\mathrm{d}t\}^{-1}$ by its discrete approximation to compare the estimates of our method, referred to as `FDboost`.

Simulation settings were generally based on $n \in \{80, 160, 320, 640\}$ observed curves with $D_i \equiv D \in \{20, 40, 60\}$ observed grid points per trajectory and a *signal-to-noise ratio* $\mathrm{SNR} \in \{0.1, 1, 10\}$. For the following subsections the combinations were customized or restricted accordingly, in particular for simulations with very time-consuming bootstrap calculations. Whereas the number of curves in our application $n = 184$ is within the range of simulated settings, we use fewer observations per trajectory in our simulations than are available in our application ($D = 384$) to reduce computational time. Increasing sampling density D from 60 to 180 or 380 in additional simulations with $\mathrm{SNR} \in \{0.01, 0.1, 1\}$ and $n = 160$ almost always results in an improvement of estimation performance. The average estimated SNR in our application is 0.42, which, because of the shrinkage effect, might potentially be underestimating the true SNR. We also present results of another simulation for $n \in \{24, 48\}$, $D \in \{190, 380\}$ and $\mathrm{SNR} \in \{0.01, 0.1, 1\}$ in the on-line appendix. The results suggest that, even for a small number of observations, the estimation performance is satisfactory when the density of sampling is sufficiently large.

The results of our simulation studies are briefly summarized in the following sections. See the on-line supplementary material for a full presentation of results.

4.1. Estimation of historical effects

Though estimation performance for simple historical effects has already been examined in Brockhaus *et al.* (2017), we provide additional simulation results for complex multimodal effect surfaces. The simulation settings are motivated by our application, in which several time windows may show a relationship between the two biosignals. We thus simulate data sets where the effect surface is multimodal for both the s-direction and the t-direction. Samples were generated from the model

$$Y_i(t) = \alpha(t) + \int_0^{t-\delta} x_i(s)\beta(s,t)\,\mathrm{d}s + \varepsilon_i(t), \qquad i = 1, \ldots, n, \tag{16}$$

for which the functional covariate $x_i(s)$ is simulated as a sum of $\varkappa \in \{5, 7, 9, 11\}$ natural cubic B-splines with independent random coefficients from a standard normal distribution. The true underlying coefficient surface is given by $\beta(s,t) = \sin(10\,|s-t|)\cos(10t)\,I(s \leqslant t - \delta)$ with $I(s \leqslant t - \delta)$ equal to 1 if $s \leqslant t - \delta$, and 0 otherwise. The independent Gaussian error process $\varepsilon(t)$ with mean 0 has constant variance σ^2 defined via $\mathrm{SNR} = \sqrt{\mathrm{var}(\Xi)}/\sigma$ with $\mathrm{var}(\Xi)$ being the empirical variance of the linear predictor.

In addition, we simulate effect surfaces with a band structure. This is done by using the data-generating process in expression (16) and restricting the influence of x_i to values s, for which $s \leqslant t - \delta$, $s \geqslant t - 0.1$ and $t \leqslant 0.75$, $s, t \in [0, 1]$. With 40 observed time points the restriction $s \geqslant t - 0.1$ corresponds to an auto-regressive model with time varying effects and a lag of $0.1/(1/40) = 4$ time points. With this simulation, we want to investigate whether our approach can adequately recover the effect of x_i restricted to a certain number of lags without having to predefine lags. This would be an advantage over time series models which must specify the assumed lag structure *a priori* and would enable a corresponding dimension reduction without restricting the analysis.

4.1.1. Results

For combinations in which n and SNR are not very small at the same time, our gradient boosting approach works well and recovers the true underlying functional relationship. These findings are depicted in Fig. 2. As can be seen in Figs 2(a)–2(c), both `pffr` and `FDboost` can recover the true underlying effect well (Figs 2(d)–2(f)) with `FDboost` having an advantage for low SNR and low n (Figs 2(a)–2(c)). For higher SNR, where `FDboost` shows less of an improvement than `pffr` compared with the low SNR setting, boosting estimates may potentially be further improved by using a higher number of iterations (limited to 1500 for this subsection). In the on-line supplementary material, we additionally provide estimates with average reliMSE for a smaller number of observations, visualizing the deterioration in estimation performance with decreasing sample size.

Similarly to the multimodal example, `FDboost` outperforms `pffr` (Figs 2(g)–2(l)) for band surfaces in settings with a lower SNR, whereas for SNR $= 10$ `pffr` shows partly better performances. As exemplarily shown in Figs 2(j)–2(l), `FDboost` can often correctly detect the non-zero regions, whereas the typical estimated surface of `pffr` exhibits larger parts with false positive estimates.

4.2. Estimation performance for factor-specific historical effects

For random historical effects, we adapt the ideas of Scheipl and Greven (2016) and Brockhaus *et al*. (2017), Web appendix C, and generate random coefficient functions $\beta_f(s,t)$ as linear combinations of cubic P-splines (Eilers and Marx, 1996) for $n_{\text{subject}} = 10$ factor levels (subjects). The coefficient functions $\beta_f(s,t)$, $f = 1, \ldots, \varphi = 10$, are then centred to comply with constraint (12). For factor-specific historical effects, we specify multiples $\iota(e)$ of one fixed coefficient function $\varpi(s,t) = (s/\sqrt{2})\cos(\pi\sqrt{t})$ with $\iota(e)$ being centred coefficients drawn uniformly between -5 and 5 for each factor level $e = 1, \ldots, \eta = 4$, allowing for a more systematic examination of estimation accuracy in specific regions of the coefficient function. An additional doubly varying effect is simulated by multiplying $\varpi(s,t)$ with centred random coefficients drawn from a standard normal distribution.

In a first series of settings (correctly specified case), the data are generated on the basis of the fitted model, including a main historical effect and

(a) a time varying categorical effect as well as a factor-specific historical effect,
(b) a time varying random effect as well as a random historical effect,
(c) combining (a) and (b) or
(d) combining (c) with a doubly varying historical effect (full model).

In a second series of settings, the model is misspecified by fitting a single historical effect, whereas the data are simulated by using a main and

(e) a factor-specific historical effect or
(f) a random historical effect or alternatively
(g) by generating the data from the full model whereas the model is fitted without the doubly varying effect.

4.2.1. Results

Whereas the main historical effect for the settings (a)–(d) shows a similar logarithmic reliMSE as in previous simulation settings in Section 4.1, the historical effects varying with a categorical covariate show more diverse performances and larger deviations. The factor-specific and random historical effect estimation mostly capture the main features of the true underlying surface but

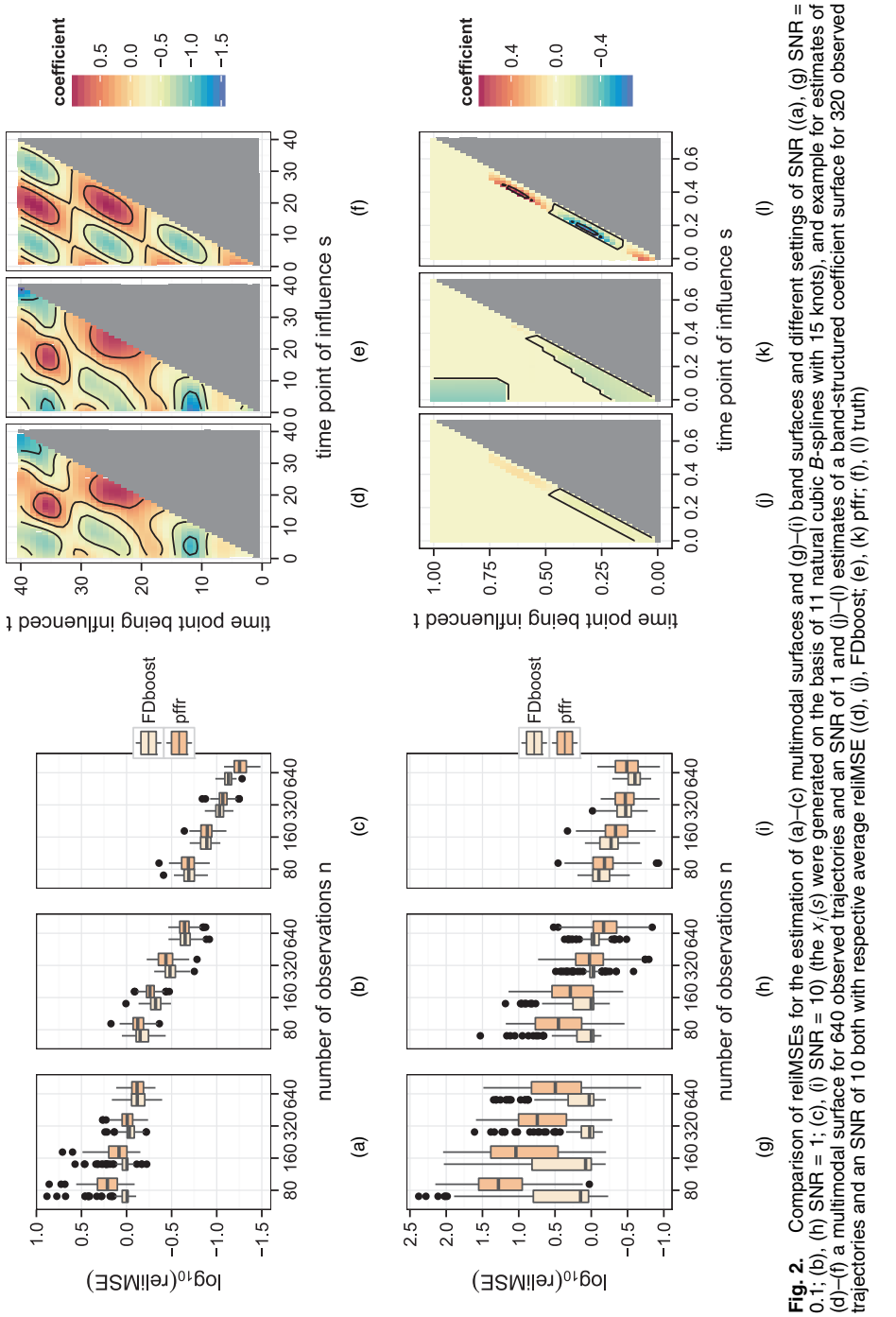

Fig. 2. Comparison of reliMSEs for the estimation of (a)–(c) multimodal surfaces and (g)–(i) band surfaces and different settings of SNR ((a), (g) SNR = 0.1; (b), (h) SNR = 1; (c), (i) SNR = 10) (the $x_i(s)$ were generated on the basis of 11 natural cubic *B*-splines with 15 knots), and example for estimates of (d)–(f) a multimodal surface for 640 observed trajectories and an SNR of 1 and (j)–(l) estimates of a band-structured coefficient surface for 320 observed trajectories and an SNR of 10 both with respective average reliMSE ((d), (j), FDboost; (e), (k) pffr; (f), (l) truth)

are not estimated as reliably as the main historical effect. Estimates for the doubly varying historical effect are often shrunk almost to 0 because of an insufficient number of observations.

In settings (e) or (f) where the true underlying model includes a random or factor-specific historical effect, estimation performance for the main historical effect is equally good when fitting the correct or the misspecified model. For setting (g) the performance is practically the same for the estimation of the main historical effect. The difference in estimation performance varies more strongly for the factor-specific as well as random historical effect and, in particular, indicates a better performance of the correctly specified model for high SNR and larger n. The fact that estimation performance is not affected more strongly is likely to be due to the orthogonality of the omitted effect to the effects that are included in the model; see equations (12)–(14).

4.3. Quantification of uncertainty

In what follows, we examine the ability of 95% bootstrap intervals to identify correctly (non-) zero coefficients in the manner of conventional confidence intervals by looking at the inclusion of zero. On the basis of 100 non-parametric bootstrap iterations, we calculate the *false negative rate FNR* and *false positive rate FPR* over the surface for each of 100 simulated data sets. In addition, the frequencies of false negative, *FFN*, and false positive estimates, *FFP*, for each surface point across all data sets are obtained. We present results for a model including only one main historical effect in addition to a model with main and factor-specific historical effects, for both of which true coefficient surfaces are partly equal to zero. The true coefficient surface for the main historical effect is defined as $\beta(s,t) = \mathcal{Q}_{0.001}\{\sin(|t-s|+10)\cos(5s)\}$ and surfaces for factor-specific historical effects are simulated as multiples of $\varpi(s,t) = \mathcal{Q}_{0.001}\{\phi_{0.9,0.2}(s)\,\phi_{0.9,0.2}(t)\}$, where $\mathcal{Q}_a(x) = x\,I(x \geqslant a)$ and $\phi_{\mu,\sigma}(\cdot)$ is the normal density function with expectation μ and variance σ^2. We additionally investigate the performance of our uncertainty quantification for a model including main and random historical effects, which are simulated as described in Section 4.2.

4.3.1. Results

Fig. 3 depicts the results for a simple historical effect simulation with SNR $= 1$, $n = 160$ and $D = 40$. Both FNR and FPR are below 0.05 in all except a few cases. When decreasing SNR to 0.1, the bootstrap approach yields smaller FPR at the cost of a larger FNR. Considering FFP and FFN, 8% of all non-zero surface points reveal an FFN of above 0.05 and 30% of all zero surface points reveal an FFP of above 0.05. Plotting FFN against the coefficient size indicates that FFNs larger than 0.05 occur only for coefficient values of below 0.2 (below 0.6 if SNR $= 0.1$). Fig. 3(d) reveals a strong relationship between FFP and a smaller distance to non-zero points on the surface, with FFP mostly below about 0.1 for points not next to a non-zero coefficient.

Though the performance depends on the specific surface, the bootstrap approach finds the majority of non-zero coefficient points in simulations for a simple historical model and tends to have an FFN of almost 0. A large FFP occurs only for surface points that are directly adjacent to true non-zero coefficient points.

For a more complex model also including a factor-specific historical effect, the bootstrap approach works well regarding the detection of the truly non-zero surface area. However, it reveals considerably higher FNR as well as higher FFN particularly for smaller coefficients of both effect surfaces. In the case of correlated observations, e.g. given by repeated measurements per subject, we subsample on the level of independent observation units (subjects). In the simulation with a main and a random historical effect, higher frequencies of false positive estimates for the main historical effect occur, which, however, are again located around the true non-zero coefficient area.

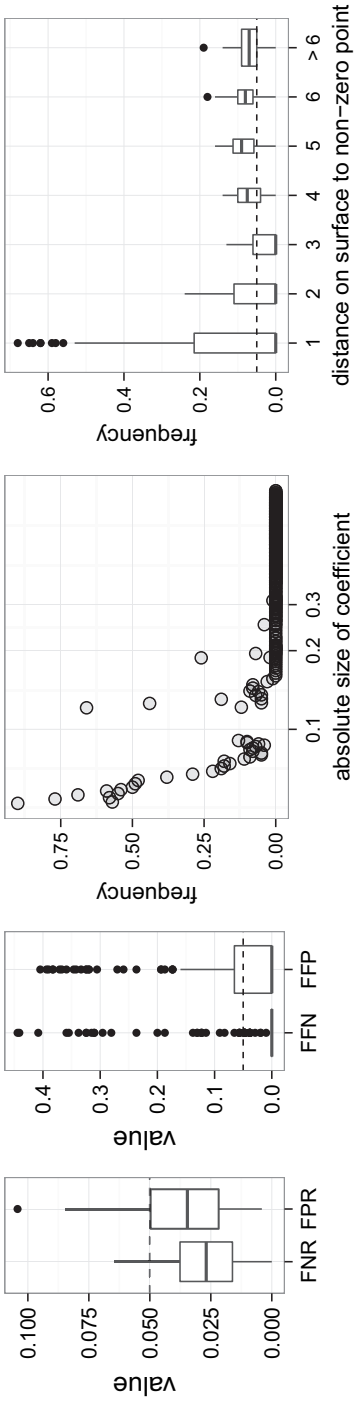

Fig. 3. Results for uncertainty quantification of a simple historical effect and data generated with SNR = 1, $n = 160$ and $D = 40$: (a) FNR and FPR for each surface with boxplots over iterations as well as (b) FFN and FFP for each surface point over all simulation iterations with boxplots over surface points; (c) frequency of bootstrap intervals including zero plotted against the coefficient size for each truly non-zero coefficient surface point; (d) frequency of false positive estimates plotted against the minimal distance to a true non-zero point for each zero surface point

In summary, simulation results suggest that the bootstrap approach does not comply with the chosen confidence level in the manner of conventional confidence intervals but proves to find most of the truly non-zero surface regions for all simulation settings. Large FFN and FFP are mainly revealed at the edges of non-zero coefficient areas, such that an interpretation of detected non-zero areas of the surface are still possible as long as exact pixel locations of edges are not taken at face value.

4.4. Further simulations

In addition to the simulations presented, we investigate the performance of boosting for different parameterizations as introduced in Section 2.2 and compare boosting estimates with step length $\nu = 0.1$ and $\nu = 1$. The gradient boosting algorithm is defined for step length $\nu \in (0, 1]$. In general, it is recommended to set the step length 'sufficiently small' (Bühlmann and Hothorn, 2007) for predictive accuracy, e.g. in the range of 0.01 and 0.1. A larger step length, and in particular $\nu = 1$, requires much fewer iteration steps and therefore speeds up the model fit but may result in a deterioration of prediction performance due to overfitting. Since we are rather interested in the estimation performance of model components, we investigate whether or how much overfitting is a problem in our particular setting.

4.4.1. Results

For the two different parameterizations, performances differ on a relatively small scale, suggesting that the choice of parameterization can be based on the given research question. In the comparison of step lengths, there appears to be no clear best choice in all settings. Thus estimation with $\nu = 1$ might be a reasonable alternative to smaller step lengths, requiring less computing time and memory consumption due to a smaller number of necessary iterations, especially in complex models applied to large data sets.

5. Application to the detection of synchronization in bioelectrical signals

5.1. Data and background

Gentsch *et al.* (2014) conducted a study in which 24 participants played a computerized gambling game with real monetary outcome. During the gambling rounds, Gentsch *et al.* (2014) modified three factors (so-called appraisals) related to Scherer's component process model (Scherer, 2009) and simultaneously recorded brain activity with EEG and facial muscle activity with EMG. In componential emotion theories such as the component process model an emotion episode is assumed to emerge through the synchronization of the emotion components (e.g. appraisals, expressions or feelings). To investigate synchronization processes, Gentsch *et al.* (2014) operationalized three dichotomous appraisals, which are included as dummy variables in the present data set:

(a) *goal conduciveness*, which was related to the monetary outcome at the end of each gambling round (*gain* coded as $G = 1$ or *loss* with $G = 0$),
(b) *power*, which allowed players to change the final outcome if the setting was *high power*, *hp*, coded as $P = 1$, otherwise referred to as *low power*, *lp*, with $P = 0$, and
(c) *control*.

The control setting was manipulated in blocks to change the participant's subjective feeling about her ability to cope with the situation. Before a block with several gambling rounds would start, participants were told whether they were going to have high or low power for the majority

of upcoming games, which corresponds to *high* or *low control* settings (respectively hc coded as $C = 1$ and lc with $C = 0$). In rounds with high control, for example, the player was told to have high power frequently, thereby trying to induce a subjective feeling of control over the situation, and vice versa for low control. Each participant played over 100 gambling rounds for each of the eight appraisal settings, which we also refer to as *trials*.

Before performing statistical analyses, EEG as well as EMG signals are preprocessed (see the on-line supplementary material for further details). After removing the data of one participant because of considerably deviating observations, which imply a defective or displaced sensor, several hundred gambling rounds each with 384 equally spaced EEG and EMG measurements within around 1500 ms are available for each of the 23 participants. Analogously to previous studies on synchronization and, in particular, the study of Gentsch *et al*. (2014), we use aggregated observations for each participant and game condition by averaging the corresponding trials for each time point. On the one hand, this results in less computing time and the feasibility to quantify uncertainty in effect estimates via the bootstrap; on the other hand, this is motivated by investigations on event-related potentials. Event-related potential analysis is a commonly practised method to infer from neuronal activity. Neuronal activity is thought to be time locked in delay to a certain stimulus, wherefore aggregating over a large number of trials is used to cancel out random brain activity and strengthens those parts of the signal, which are commonly observed for all trials (see, for example, Pfurtscheller and da Silva (1999), Handy (2005) and Rousselet *et al*. (2008)).

Instead of combining the (spatially correlated) EEG signals to maximize the explanatory power of the analysis, the question of interest rather lies in the dominant influence of certain selected EEG signals. We fit a model for each EEG signal of interest (Fz-, FCz-, POz- and Pz-electrode) to determine the direct effect on the facial muscle activity. To demonstrate the ability of our approach to handle high-dimensional data sets, we also provide sample code in the repository for fitting a model, in which all 64 EEG signals are potentially included with historical, factor-specific and random historical effects. In the on-line supplementary material, we additionally provide a visualization for the selection frequency of this model after 2000 iterations.

5.2. Model

It is predicted that facial expression is largely driven by efferent brain signals reflecting appraisal processes. We use the following maximal model:

$$Y_{il}(t) = \sum_{j=1}^{13} h_j(\mathbf{x}_{il})(t) + \varepsilon_{il}(t), \tag{17}$$

for $l = 1, \ldots, n_{\text{setting}} = 8, i = 1, \ldots, n_{\text{subject}} = 23, t \in \mathcal{T} = [0\,\text{ms}, 1500\,\text{ms}]$ and $D_i \equiv D = 384$ observed time points in $\mathcal{T}$. In model (17), $Y_{il}(t)$ represents a chosen EMG signal for subject i, game condition l and time point t in the game. $h_j(\mathbf{x}_{il})(t)$, or, for short, $h_j(t)$ are 13 partial effects of covariates $\mathbf{x}_{il}$ including a time varying intercept, game condition effects (C, P, G) and EEG signal effects depending on the selected electrode signal ω_{il}. Table 1 provides the details on each part of the linear predictor. For the integration limits, we use $l(t) = 0$ and a lead parameter $u(t) = t - \delta = t - 12$ ms, which is meaningful because of restrictions given by the neuroanatomy of humans and is just below the time lag between EMG and EEG of 14.3 ms (Mima and Hallett, 1999). To reflect subject-specific variation, we include time varying random intercepts and subject-specific historical EEG effects in the model.

Though game condition-specific historical effects may be subject specific, simulations in the

Table 1. Partial effects in the EMG–EEG model

Partial effect $h_j(x_{il})(t)$	*Effect (of)*
$h_1(t) = \alpha(t)$	Intercept
$h_2(t) = b_{0,i}(t)$	Subject-specific intercepts
$h_3(t) = \gamma_1(t) C_{il}$	Game condition control
$h_4(t) = \gamma_2(t) P_{il}$	Game condition power
$h_5(t) = \gamma_3(t) G_{il}$	Game condition goal conduciveness
$h_6(t) = \gamma_4(t) C_{il} P_{il}$	Interaction of control and power
$h_7(t) = \gamma_5(t) C_{il} G_{il}$	Interaction of control and goal conduciveness
$h_8(t) = \gamma_6(t) P_{il} G_{il}$	Interaction of power and goal conduciveness
$h_9(t) = \gamma_7(t) C_{il} P_{il} G_{il}$	Interaction of all game conditions
$h_{10}(t) = \int_0^{t-12} \omega_{il}(s) \beta_1(s,t) \mathrm{d}s$	EEG signal
$h_{11}(t) = \int_0^{t-12} \omega_{il}(s) \beta_{2,l}(s,t) \mathrm{d}s$	EEG signal (game condition specific)
$h_{12}(t) = \int_0^{t-12} \omega_{il}(s) b_{1,i}(s,t) \mathrm{d}s$	EEG signal (subject specific)
$h_{13}(t) = \int_0^{t-12} \omega_{il}(s) b_{2,i,l}(s,t) \mathrm{d}s$	EEG signal (subject and game condition specific)

previous section suggest that, even if the true model corresponds to the full model, estimation performance is only slightly affected when using a misspecified model without a random factor-specific historical effect $h_{13}(t)$. As a sensitivity analysis, we also fit the full model including $h_{13}(t)$ on a finer aggregation of the data, for which we average over fewer trials per subject and thus obtain repeated measurements per subject–game condition combination.

5.3. Results

For the historical effects, the estimated coefficient surfaces are depicted in Fig. 4 for the EEG covariate in the form of the electrode Fz (in particular measuring intentional and motivational activities; Teplan (2002)) and the EMG response signal of the *frontalis* muscle (which raises the eyebrows). The lower panel in each part of Fig. 4 depicts the average EEG signal per game condition, demeaned per time point by the overall mean and with negative or positive values highlighted in blue or red respectively. Two further panels (left and centre of each part) for the EMG signal show the overall mean, the prediction with and without the historical effects (left) as well as the difference between these predictions (centre). For predictions, the average EEG signal per game condition was used. Additionally, corresponding bootstrap results for uncertainty assessment are incorporated in the figures by different degrees of transparency related to different pointwise bootstrap intervals $\mathrm{BI}_\alpha = [q_{\alpha/2}, q_{1-\alpha/2}]$, q_a as $\alpha\%$-bootstrap quantile and $\alpha \in \{1, 5, 10\}$. Surface points are coloured with the corresponding coefficient value and are less transparent if the specified bootstrap interval does not contain the value zero.

Fig. 4 shows the sum of the estimated coefficient surfaces of main and game condition-specific historical effects for the four high control settings (the other four surfaces are included in the on-line appendix). In all four effect surfaces a similar pattern can be found, which reflects the structure of the main historical effect. The coefficients near the diagonal reveal a positive sign at around $s \approx 500$ ms, whereas the upper left as well as the upper right of the surface, visually separated by a thick black contour line, are estimated with a negative sign. In contrast with the upper left negative coefficient area, which is mostly indicated to be not different from zero by the boostrap, the upper right negative coefficient area is indicated to be non-zero for all eight conditions at least to some extent. The positive area in between those two negative subareas is mostly estimated to be either zero or non-zero but with relatively small coefficient values. The

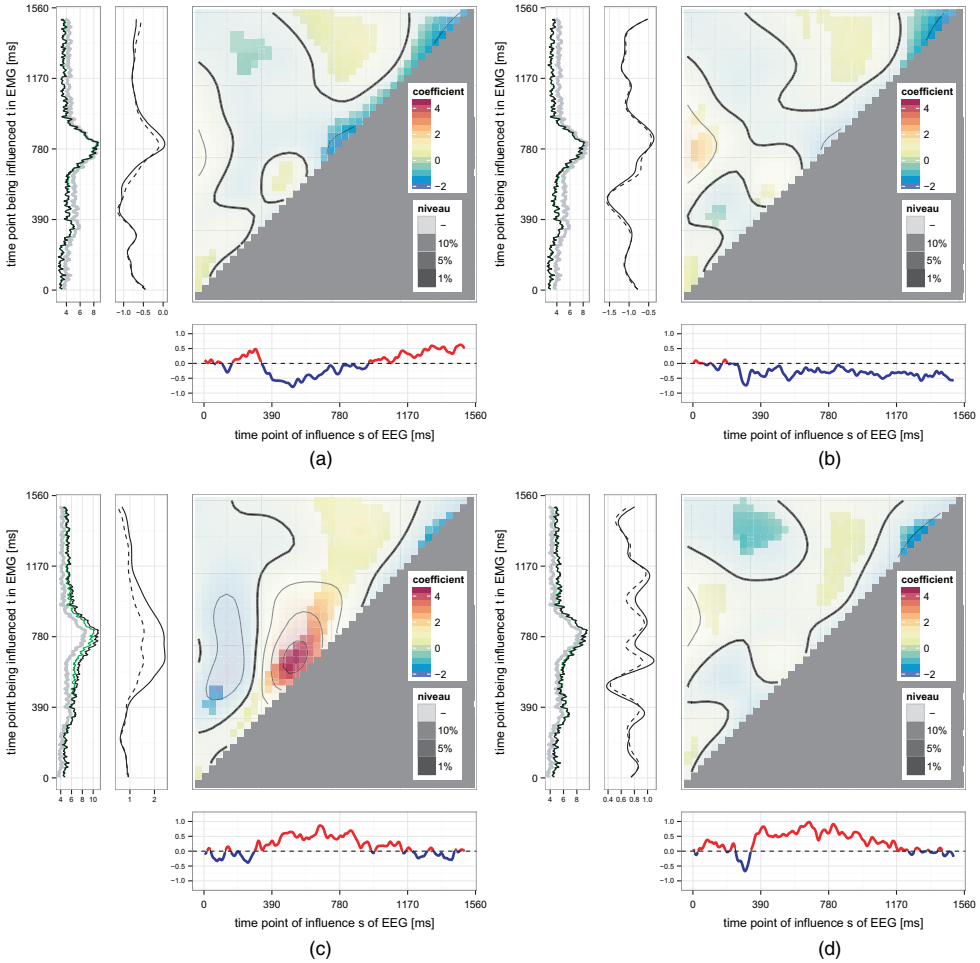

Fig. 4. Estimated coefficient surfaces for the model with EEG covariate Fz (plot of average signals per game condition at the bottom with negative and positive values highlighted in blue and red respectively; signals are demeaned per time point by the overall mean), all four high control settings and the EMG response signal of the *frontalis* muscle (left panels: overall mean (1) in grey, prediction without historical effects (2) in green, with historical effects (3) by using the average EEG signal per game condition in black; centre panel: ------, difference between (1) and (2); ———, difference between (1) and (3)) (surfaces correspond to estimated main historical effect plus game condition-specific historical effect; different degrees of transparency in the coefficient plots indicate surface points having 1 – niveau bootstrap intervals which do not contain the value zero; to obtain a reasonably sized image estimated effects are visualized on a 40 × 40 grid): (a) hc–hp–gain; (b) hc–hp–loss; (c) hc–lp–gain; (d) hc–lp–loss

positive effect near the diagonal at $s \approx 500$ ms is estimated to have the largest values for hc-settings in combination with hp–loss and lp–gain situations and is found to be non-zero by the bootstrap only for the latter scenario. This very strong short-term synchronization of EEG and EMG signals seems to be very reasonable from a theoretical point of view, as facial reactions including raising of the eyebrows are usually brief and are linked to appraisals such as novelty,

which is consistent in the hc–lp–gain case with low power not being expected in a high control setting (Scherer, 2009).

The estimated effect can on the one hand be interpreted on the subject level. A person with a higher EEG signal at $s \approx 500$ ms, for example, will on average show a higher EMG signal (i.e. stronger muscle activity) for $t \approx 600$ ms, given that the preceding EEG signal and game condition remain the same. On the other hand, effects can be explained by relating the demeaned average EEG signal for one game condition and the corresponding coefficients to the changes in the average EMG signal, which is illustrated by the hc–lp–gain setting in Fig. 4. As EEG values related to this game condition are on average above the overall mean EEG values for $s \in [300, 1000]$ ms, the EEG value seems to have an increasing effect on subsequent EMG values and thus muscle activity, with the effect lasting for at least 100 ms.

In theory, muscle activity should be traceable to brain signals. Therefore the results indicate that brain activity measured at the Fz electrode contributes to only a relatively small amount in explaining the movement of eyebrows (the difference panels on the left of each plot in Fig. 4). However, for the game condition hc–lp–gain, the model explains a considerable amount of EMG activity (which is particularly visible in the difference plot of EMG predictions).

When reparameterizing the factor-specific historical effects without historical main effect, when boosting with step length 1 as well as in the full model with more finely aggregated data, the estimated effects are similar to the reported effects. Further results for the application are given in the on-line appendix, including results for the scalar covariates.

Gentsch *et al*. (2014) analysed EEG and EMG signals separately and made statements regarding differences in game conditions for one of the signals at a time. Although this and other similar strategies may yield results on significant changes in one signal for different study settings, no statement on the association of the two signals can be made. In contrast, investigating the emotion components data with our proposed approach facilitates the modelling of synchronization of EMG and EEG signals in the first place and additionally allows the simultaneous EEG and EMG analysis to differ for influence factors given by the study design. Our method therefore can recreate parts of the theoretical emotion components model and leads to new insights on the underlying synchronization process. Specifically, we found associations between EEG and EMG signals that are time localized (without the need to prespecify time lags) and which differ between experimental settings, with setting hc–lp–gain showing the clearest association.

6. Discussion

The focus of this paper is the development of a regression framework for the synchronization analysis of bioelectrical signal data. Bioelectrical signals like EEG or EMG signals are recorded in many research areas, such as in neuroscience or cognitive neuropsychology, where the goal is to develop an understanding of synchronization processes in emotion episodes. In contrast with previous approaches, which are mostly based on coherence, cross-correlation or similar concepts (see, for example, Mima and Hallett (1999), Brown (2000) and Grosse *et al*. (2002)), we use a function-on-function regression model (see, for example Morris (2015)) with factor-specific historical effects. Our model extends the simple historical model (Malfait and Ramsay, 2003; Harezlak *et al*., 2007; Brockhaus *et al*., 2017) by factor-specific and/or random historical effects. As far as we know, no methods are available other than `FDboost` allowing historical effects to vary with other covariates. We develop constraints to make the resulting estimates both interpretable as well as identifiable. This flexible class of function-on-function regression models is implemented in the R package `FDboost`. Using the componentwise gradient boosting

approach by Brockhaus *et al*. (2015, 2017) for estimation, this approach can deal with high dimensional data, even $p > n$ settings, and includes variable selection. The algorithm can recover different effect surfaces, including relationships that are assumed in time series approaches, and allows for potentially time varying associations. The quality of estimates is comparable with those of the function `pffr` of the R package `refund` for special cases of function-on-function regression where `pffr` is applicable.

A bootstrap can be employed to assess the variability of boosted estimates. Although bootstrap intervals, because of the shrinkage, do not constitute confidence intervals with proper coverage, simulations show that the bootstrap approach can recover areas with non-zero effects very well and shows a larger FPR and FNR only at the edges of true non-zero effect surfaces. A better uncertainty quantification would be a relevant avenue for future developments.

Although we do not focus on this feature here, our approach can also model other characteristics of the conditional response distribution than the mean, such as the median or a quantile. A more complex yet interesting class of models would be obtained by combining functional regression models with generalized additive models for location, scale and shape as done for scalar response by Brockhaus *et al*. (2016).

For the emotion components data, our model contributes to the understanding of the componential theory by estimating a functional relationship between the EEG and EMG signals without having to prespecify a certain time lag between these two signals. In addition, our proposed extension for historical models enables appraisal-specific investigations on synchronization processes of emotion components.

Acknowledgements

We thank Fabian Scheipl for his help and useful comments. Sonja Greven, Sarah Brockhaus and David Rügamer acknowledge funding by Emmy Noether grant GR 3793/1-1 from the German Research Foundation. Kornelia Gentsch and Klaus Scherer were funded by a European Research Council advanced grant in the European Community's seventh framework programme under grant agreement 230331-PROPEREMO to Klaus Scherer and by the National Center of Competence in Research Affective Sciences financed by the Swiss National Science Foundation (grant 51NF40-104897) hosted by the University of Geneva.

References

Binder, H., Porzelius, C. and Schumacher, M. (2009) Rank-based p-values for sparse high-dimensional risk prediction models fitted by componentwise boosting. *Preprint 101*. University of Freiburg, Freiburg.

Bortel, R. and Sovka, P. (2006) EEG-EMG coherence enhancement. *Signl Process.*, **86**, 1737–1751.

Brockhaus, S., Fuest, A., Mayr, A. and Greven, S. (2016) Signal regression models for location, scale and shape with an application to stock returns. *Preprint*. (Available from `https://arxiv.org/abs/1605.04281`.)

Brockhaus, S., Melcher, M., Leisch, F. and Greven, S. (2017) Boosting flexible functional regression models with a high number of functional historical effects. *Statist. Comput.*, **27**, 913–926.

Brockhaus, S. and Rügamer, D. (2016) FDboost: boosting functional rgression models. *R Package Version 0.2-0.* (Available from `http://CRAN.R-project.org/package=FDboost`.)

Brockhaus, S., Scheipl, F., Hothorn, T. and Greven, S. (2015) The functional linear array model. *Statist. Modllng*, **15**, 279–300.

Brown, P. (2000) Cortical drives to human muscle: the piper and related rhythms. *Progr. Neurbiol.*, **60**, 97–108.

Brumback, B. A., Ruppert, D. and Wand, M. P. (1999) Comment on "Variable selection and function estimation in additive nonparametric regression using a data-based prior". *J. Am. Statist. Ass.*, **94**, 794–797.

Bühlmann, P. and Hothorn, T. (2007) Boosting algorithms: regularization, prediction and model fitting (with discussion). *Statist. Sci.*, **22**, 477–505.

Diab, A., Hassan, M., Boudaoud, S., Marque, C. and Karlsson, B. (2013) Nonlinear estimation of coupling and directionality between signals: application to uterine EMG propagation. In *Proc. 35th A. Int. Conf. Engineering in Medicine and Biology Society*, pp. 4366–4369. New York: Institute of Electrical and Electronics Engineers.

Eilers, P. H. C. and Marx, B. D. (1996) Flexible smoothing with B-splines and penalties. *Statist. Sci.*, **11**, 89–121.

Gentsch, K., Grandjean, D. and Scherer, K. R. (2014) Coherence explored between emotion components: evidence from event-related potentials and facial electromyography. *Biol. Psychol.*, **98**, 70–81.
Gervini, D. (2015) Dynamic retrospective regression for functional data. *Technometrics*, **57**, 26–34.
Grandjean, D. and Scherer, K. (2009) Synchronization (and emotion). In *The Oxford Companion to Emotion and the Affective Sciences* (eds D. Sander and K. Scherer). Oxford: Oxford University Press.
Grosse, P., Cassidy, M. and Brown, P. (2002) EEG-EMG, MEG-EMG and EMG-EMG frequency analysis: physiological principles and clinical applications. *Clin. Neurphysiol.*, **113**, 1523–1531.
Handy, T. (2005) *Event-related Potentials: a Methods Handbook*. Cambridge: MIT Press.
Harezlak, J., Coull, B. A., Laird, N. M., Magari, S. R. and Christiani, D. C. (2007) Penalized solutions to functional regression problems. *Computnl Statist. Data Anal.*, **51**, 4911–4925.
Hashimoto, Y., Ushiba, J., Kimura, A., Liu, M. and Tomita, Y. (2010) Correlation between EEG-EMG coherence during isometric contraction and its imaginary execution. *Acta Neurbiol. Exp.*, **70**, 76–85.
Hofner, B., Hothorn, T., Kneib, T. and Schmid, M. (2011) A framework for unbiased model selection based on boosting. *J. Computnl Graph. Statist.*, **20**, 956–971.
Hollenstein, T. and Crowell, S. (2014) Whither concordance?: Autonomic psychophysiology and the behaviors and cognitions of emotional responsivity. *Biol. Psychol.*, **98**, 1–94.
Hothorn, T., Bühlmann, P., Kneib, T., Schmid, M. and Hofner, B. (2016) mboost: model-based boosting. *R Package Version 2.6-0.* (Available from `http://CRAN.R-project.org/package=mboost`.)
Huang, L., Scheipl, F., Goldsmith, J., Gellar, J., Harezlak, J., McLean, M. W., Swihart, B., Xiao, L., Crainiceanu, C. and Reiss, P. (2015) refund: regression with functional data. *R Package Version 0.1-13.* (Available from `http://CRAN.R-project.org/package=refund`.)
Kang, J. M., Yoo, T. and Kim, H. C. (2006) A wrist-worn integrated health monitoring instrument with a tele-reporting device for telemedicine and telecare. *IEEE Trans. Instrumntn Measmnt*, **55**, 1655–1661.
Kaniusas, E. (2012) *Fundamentals of Biosignals*, pp. 1–26. Berlin: Springer.
Kneib, T. (2013) Beyond mean regression. *Statist. Modllng*, **13**, 275–303.
Malfait, N. and Ramsay, J. O. (2003) The historical functional linear model. *Can. J. Statist.*, **31**, 115–128.
Meinshausen, N. and Bühlmann, P. (2010) Stability selection (with discussion). *J. R. Statist. Soc.* B, **72**, 417–473.
Mima, T. and Hallett, M. (1999) Electroencephalographic analysis of cortico-muscular coherence: reference effect, volume conduction and generator mechanism. *Clin. Neurphysiol.*, **110**, 1892–1899.
Mima, T., Matsuoka, T. and Hallett, M. (2000a) Functional coupling of human right and left cortical motor areas demonstrated with partial coherence analysis. *Neursci. Lett.*, **287**, 93–96.
Mima, T., Steger, J., Schulman, A. E., Gerloff, C. and Hallett, M. (2000b) Electroencephalographic measurement of motor cortex control of muscle activity in humans. *Clin. Neurphysiol.*, **111**, 326–337.
Morris, J. S. (2015) Functional regression. *A. Rev. Statist. Appl.*, **2**, 321–359.
Ozaki, T. (2012) *Time Series Modeling of Neuroscience Data*. Boca Raton: CRC Press.
Pawitan, Y. (2005) Coherence between time series. In *Encyclopedia of Biostatistics*, vol. 2 (eds P. Armitage and T. Colton). New York: Wiley.
Pfurtscheller, G. and da Silva, F. L. (1999) Event-related EEG/MEG synchronization and desynchronization: basic principles. *Clin. Neurphysiol.*, **110**, 1842–1857.
Quiroga, R. Q., Kreuz, T. and Grassberger, P. (2002) Event synchronization: a simple and fast method to measure synchronicity and time delay patterns. *Phys. Rev.* E, **66**, article 041904.
Rinn, W. E. (1984) The neuropsychology of facial expression: a review of the neurological and psychological mechanisms for producing facial expressions. *Psychol. Bull.*, **95**, 52–77.
Rousselet, G. A., Husk, J. S., Bennett, P. J. and Sekuler, A. B. (2008) Time course and robustness of ERP object and face differences. *J. Visn*, **8**, article 3.1–18.
Ruppert, D., Wand, M. P. and Carroll, R. J. (2003) *Semiparametric Regression*. New York: Cambridge University Press.
Scheipl, F. and Greven, S. (2016) Identifiability in penalized function-on-function regression models. *Electron. J. Statist.*, **10**, 495–526.
Scheipl, F., Staicu, A.-M. and Greven, S. (2015) Functional additive mixed models. *J. Computnl Graph. Statist.*, **24**, 477–501.
Scherer, K. R. (2009) The dynamic architecture of emotion: evidence for the component process model. *Cogn. Emotn*, **23**, 1307–1351.
Semmlow, J. L. and Griffel, B. (2014) *Biosignal and Medical Image Processing*. Boca Raton: CRC Press.
Teplan, M. (2002) Fundamentals of EEG measurement. *Measmnt Sci. Rev.*, **2**, 1–11.
Wood, S. N. (2006) *Generalized Additive Models: an Introduction with R*. Boca Raton: Chapman and Hall–CRC.
Zhang, T. and Yu, B. (2005) Boosting with early stopping: convergence and consistency. *Ann. Statist.*, **33**, 1538–1579.

Supporting information
Additional 'supporting information' may be found in the on-line version of this article:

'Boosting factor-specific functional historical models for the detection of synchronisation in bioelectrical signals'.

Chapter 6

Boosting Functional Regression Models with FDboost

Chapter 6 describes the implementation of a large variety of functional regression models given in the R package `FDboost`. Based on the component-wise functional gradient descent algorithm, the package constitutes a flexible estimation framework for scalar-on-function regression, function-on-scalar regression and function-on-function regression, including models presented in Chapter 5.

Contributing article:

Brockhaus, S., Rügamer, D., and Greven, S. (2017). Boosting Functional Regression Models with FDboost. *ArXiv e-prints arXiv:1705.10662*.

Author contributions:

The following manuscript was written by Sarah Brockhaus and David Rügamer in equal parts. Contributions were mostly divided on a thematical basis with mutual feedback. The first draft of all (sub-)sections concerned with scalar response were authored by Sarah Brockhaus and all (sub-)sections concerned with functional response were authored by David Rügamer. Sonja Greven added valuable input and proofread the manuscript.

Boosting Functional Regression Models with FDboost

Sarah Brockhaus
LMU Munich

David Rügamer
LMU Munich

Sonja Greven
LMU Munich

Abstract

The R add-on package **FDboost** is a flexible toolbox for the estimation of functional regression models by model-based boosting. It provides the possibility to fit regression models for scalar and functional response with effects of scalar as well as functional covariates, i.e., scalar-on-function, function-on-scalar and function-on-function regression models. In addition to mean regression, quantile regression models as well as generalized additive models for location scale and shape can be fitted with **FDboost**. Furthermore, boosting can be used in high-dimensional data settings with more covariates than observations. We provide a hands-on tutorial on model fitting and tuning, including the visualization of results. The methods for scalar-on-function regression are illustrated with spectrometric data of fossil fuels and those for functional response regression with a data set including bioelectrical signals for emotional episodes.

Keywords: functional data analysis, function-on-function regression, function-on-scalar regression, gradient boosting, model-based boosting, scalar-on-function regression.

1. Introduction

With the progress of technology today, we have the ability to observe more and more data of a functional nature, such as curves, trajectories or images (Ramsay and Silverman 2005). Functional data can be found in many scientific fields like demography, biology, medicine, meteorology and economics (see, e.g., Ullah and Finch 2013). In practice, the functions are observed on finite grids. In this paper, we deal with one-dimensional functional data that are observed over a real valued interval. Examples for such data are growth curves over time, acoustic signals, temperature curves and spectrometric measurements in a certain range of wavelengths. Regression models are a versatile tool for data analysis and various models have been proposed for regression with functional variables; see Morris (2015) and Greven and Scheipl (2017) for recent reviews of functional regression models. One can distinguish between three different types of functional regression models: scalar-on-function regression, a regression with scalar response and functional covariates, function-on-scalar regression referring to models with functional response and scalar covariates and function-on-function regression, which is used when both response and covariates are functional. Models for scalar-on-function regression are sometimes also called signal regression.

Greven and Scheipl (2017) lay out a generic framework for functional regression models including the three mentioned model types. Many types of covariate effects are discussed including linear and non-linear effects of scalar covariates as well as linear effects of functional covariates and interaction terms. They describe that estimation can be based on a mixed models framework (Scheipl, Staicu, and Greven 2015; Scheipl, Gertheiss, and Greven 2016) or on component-wise gradient boosting (Brockhaus, Scheipl, Hothorn, and Greven 2015; Brockhaus, Melcher, Leisch, and Greven 2017). In this paper, we describe the latter approach and provide a hands-on tutorial for its

implementation in R (R Core Team 2017) in the comprehensive R package **FDboost** (Brockhaus and Rügamer 2017).

Boosting estimates the model by iteratively combining simple models and can be seen as a method that conducts gradient descent (Bühlmann and Hothorn 2007). Boosting is capable of estimating models in high-dimensional data settings and implicitly does variable selection. The modeled features of the conditional response distribution can be chosen quite flexibly by minimizing different loss functions. The framework includes linear models (LMs), generalized linear models (GLMs) as well as quantile and expectile regression. Furthermore, generalized additive models for location, scale and shape (GAMLSS, Rigby and Stasinopoulos 2005) can be fitted (Mayr, Fenske, Hofner, Kneib, and Schmid 2012). GAMLSS model all distribution parameters of the conditional response distribution simultaneously depending on potentially different covariates. Brockhaus, Fuest, Mayr, and Greven (2018) discuss GAMLSS with scalar response and functional covariates. Stöcker, Brockhaus, Schaffer, von Bronk, Opitz, and Greven (2017) introduce GAMLSS for functional response. Due to variable selection and shrinkage of the coefficient estimates, no classical inference concepts are available for the boosted models. However, it is possible to quantify uncertainty by bootstrap (Efron 1979) and stability selection (Meinshausen and Bühlmann 2010). The main advantages of the boosting approach are the possibility to fit models in high dimensional data settings with variable selection and to estimate not only mean regression models but also GAMLSS and quantile regression models. The main disadvantage is the lack of formal inference.

Other frameworks for flexible regression models with functional response exist. Morris and Carroll (2006) and Meyer, Coull, Versace, Cinciripini, and Morris (2015) use a basis transformations approach and Bayesian inference to model functional variables. Usually, loss-less transformations like a wavelet transformation are used. See Morris (2017) for a detailed comparison of the two frameworks.

In this tutorial, we present the R package **FDboost** (Brockhaus and Rügamer 2017), which is designed to fit a great variety of functional regression models by boosting. **FDboost** builds on the R package **mboost** (Hothorn, Bühlmann, Kneib, Schmid, and Hofner 2016) for statistical model-based boosting. Thus, in the back-end we rely on a well-tested implementation. **FDboost** provides a comprehensive implementation of the most important methods for boosting functional regression models. In particular, the package can be used to conveniently fit models with functional response. For effects of scalar covariates on functional responses, we provide base-learners with suitable identifiability constraints. In addition, base-learners that model effects of functional covariates are implemented. The package also contains functions for model tuning and for visualizing results.

As a case study for scalar-on-function regression, we use a dataset on fossil fuels, which was analyzed in Fuchs, Scheipl, and Greven (2015) and Brockhaus *et al.* (2015) and is part of the **FDboost** package. In this application, the heat value of fossil fuels should be predicted based on spectral data. As a case study for function-on-scalar and function-on-function regression, we use the emotion components data set, which is analyzed in Rügamer, Brockhaus, Gentsch, Scherer, and Greven (2018) in the context of factor-specific historical effect estimation and which is provided in an aggregated version in **FDboost**. Note that we use both data sets as a running example to illustrate the capabilities of the package. We give a more complex example with a stronger focus on answering the underlying research question in Appendix E.

The remainder of the paper is structured as follows. We shortly review the generic functional regression model (Section 2) for scalar and for functional response. Then the boosting algorithm used for model fitting is introduced in Section 3. In Section 4, we give details on the infrastructure of the package **FDboost**. Scalar-on-function regression with **FDboost** is described in Section 4.1.

Regression models for functional response with scalar and/or functional covariates are described in Section 4.2. We present possible covariate effects as well as discuss model tuning and show how to extract and display results. In Section 4.3, we discuss regression models that model other characteristics of the response distribution than the mean, in particular median regression and GAMLSS. In Section 4.4, we shortly comment on stability selection in combination with boosting. In Section 4.4 we comment on the computational burden of fitting models with **FDboost**. We conclude with a discussion in Section 5. The paper is structured such that the subsections on functional response can be skipped if one is only interested in scalar-on-function regression.

2. Functional regression models

In Section 2.1 we first introduce a generic model for scalar response with functional and scalar covariates. Afterwards, we deal with models with functional response in Section 2.2.

2.1. Scalar response and functional covariates

Let the random variable Y be the scalar response with realization $y \in \mathbb{R}$. The covariate set $\boldsymbol{X}$ can include both scalar and functional variables. We denote a generic scalar covariate by Z and a generic functional covariate by $X(s)$, with $s \in \mathcal{S} = [S_1, S_2]$ and $S_1 < S_2$, $S_1, S_2 \in \mathbb{R}$. We assume that we observe $i = 1, \ldots, N$ data pairs $(y_i, \boldsymbol{x}_i)$, where $\boldsymbol{x}_i$ comprises the realizations z_i of scalar covariates as well as the realizations $x_i(s)$ of $X_i(s)$. In practice, $x_i(s)$ is observed on a grid of evaluation points $s_1, \ldots, s_R$, such that each curve is observed as a vector $(x_i(s_1), \ldots, x_i(s_R))^\top$. While different functional covariates may be observed on different grid points over different intervals, which is supported by **FDboost** as also the following example will show, we do no introduce additional indices here for ease of notation.

We model the expectation of the response by an additive regression model

$$\mathbb{E}(Y_i|\boldsymbol{X}_i = \boldsymbol{x}_i) = h(\boldsymbol{x}_i) = \sum_{j=1}^{J} h_j(\boldsymbol{x}_i), \tag{1}$$

where $h(\boldsymbol{x}_i)$ is the additive predictor containing the additive effects $h_j(\boldsymbol{x}_i)$. Each effect $h_j(\boldsymbol{x}_i)$ can depend on one or more covariates in $\boldsymbol{x}_i$. Possible effects include linear, non-linear and interaction effects of scalar covariates as well as linear effects of functional covariates. Moreover, group-specific effects and interaction effects between scalar and functional variables are possible. To give an idea of possible effects $h_j(\boldsymbol{x})$, Table 1 lists effects of functional covariates that are currently implemented in **FDboost**. A scalar-on-function model with only one functional covariate would

covariate(s)	type of effect	$h_j(x)$
functional covariate $x(s)$	linear functional effect	$\int_{\mathcal{S}} x(s)\beta(s)\,ds$
scalar and functional covariate, z and $x(s)$	linear interaction	$z\int_{\mathcal{S}} x(s)\beta(s)\,ds$
	smooth interaction	$\int_{\mathcal{S}} x(s)\beta(z,s)\,ds$

Table 1: Overview of possible covariate effects of functional covariates, including interaction effects with scalar covariates.

be $\mathbb{E}(Y_i|\boldsymbol{X}_i = \boldsymbol{x}_i) = \beta_0 + \int_{\mathcal{S}} x_i(s)\beta(s)\,ds$, see Section 4.1 for concrete examples of scalar-on-function models for the fossil fuel data set.

The effects $h_j(\boldsymbol{x}_i)$ are linearized using a basis representation:

$$h_j(\boldsymbol{x}_i) = \boldsymbol{b}_j(\boldsymbol{x}_i)^\top \boldsymbol{\theta}_j, \quad j = 1, \ldots, J, \tag{2}$$

with basis vector $\boldsymbol{b}_j(\boldsymbol{x}_i) \in \mathbb{R}^{K_j}$ and coefficient vector $\boldsymbol{\theta}_j \in \mathbb{R}^{K_j}$ that has to be estimated. The $N \times K_j$ design matrix for the jth effect consists of rows $\boldsymbol{b}_j(\boldsymbol{x}_i)^\top$ for all observations $i = 1, \ldots, N$. A ridge-type penalty term $\lambda_j \boldsymbol{\theta}_j^\top \boldsymbol{P}_j \boldsymbol{\theta}_j$ is used for regularization, where $\boldsymbol{P}_j$ is a suitable penalty matrix for $\boldsymbol{b}_j$ and λ_j is a non-negative smoothing parameter. The smoothing parameter controls the degrees of freedom of the effect.

Consider, for example, a linear effect of a functional covariate $\int_{\mathcal{S}} x_i(s)\beta(s)\, ds$. Using $\boldsymbol{\theta}_j = (\theta_{j1}, \ldots, \theta_{jK_j})^\top$, this effect is computed as

$$\begin{aligned}
\int_{\mathcal{S}} x_i(s)\beta(s)\, ds &= \int_{\mathcal{S}} x_i(s) \underbrace{\sum_{k=1}^{K_j} \phi_k(s)\theta_{jk}}_{\approx \beta(s)}\, ds \\
&\approx \sum_{r=1}^{R} \Big(\Delta(s_r) x_i(s_r) \sum_{k=1}^{K_j} \phi_k(s_r)\theta_{jk} \Big) \\
&= \sum_{k=1}^{K_j} \Big(\underbrace{\sum_{r=1}^{R} \Delta(s_r) x_i(s_r) \phi_k(s_r)}_{\text{entries in } \boldsymbol{b}_j(\boldsymbol{x}_i)}\, \theta_{jk} \Big) \\
&= \boldsymbol{b}_j(\boldsymbol{x}_i)^\top \boldsymbol{\theta}_j,
\end{aligned}$$

where first, the smooth effect $\beta(s)$ is expanded in basis functions, second, the integration is approximated by a weighted sum and, third, the terms are rearranged such that they fit into the scheme $\boldsymbol{b}_j(\boldsymbol{x}_i)^\top \boldsymbol{\theta}_j$. The basis $\boldsymbol{b}_j(\boldsymbol{x}_i)$ is thus computed as

$$\begin{aligned}
\boldsymbol{b}_j(\boldsymbol{x}_i)^\top &= \left[\sum_{r=1}^{R} \Delta(s_r) x_i(s_r) \phi_1(s_r) \;\cdots\; \sum_{r=1}^{R} \Delta(s_r) x_i(s_r) \phi_{K_j}(s_r) \right] \\
&\approx \left[\int_{\mathcal{S}} x_i(s)\phi_1(s)\, ds \;\cdots\; \int_{\mathcal{S}} x_i(s)\phi_{K_j}(s)\, ds \right],
\end{aligned} \tag{3}$$

with spline functions ϕ_k, $k = 1, \ldots, K_j$, for the expansion of the smooth effect $\beta(s)$ in s direction and integration weights $\Delta(s_r)$ for numerical computation of the integral. The penalty matrix $\boldsymbol{P}_j$ is chosen such that it is suitable to regularize the splines ϕ_k. In the current implementation only P-splines are readily available to estimate smooth effects. To set up a P-spline basis (Eilers and Marx 1996) for the smooth effect, ϕ_k in Equation 3 are B-splines and the penalty $\boldsymbol{P}_j$ is a squared difference matrix.

Case study: Heat value of fossil fuels

The aim of this application is to predict the heat value y of fossil fuels using spectral data (Fuchs *et al.* 2015, Siemens AG). For $N = 129$ samples, the dataset contains the heat value, the percentage of humidity z_{H2O} and two spectral measurements, which can be thought of as functional variables $x_{\text{NIR}}(s_{\text{NIR}})$ observed over $\mathcal{S}_{\text{NIR}} = [250.4, 876.8]$ and $x_{\text{UV}}(s_{\text{UV}})$ observed over $\mathcal{S}_{\text{UV}} = [800.4, 2761.0]$. One spectrum is ultraviolet-visible (UVVIS), the other a near infrared spectrum (NIR). For both

spectra, the observation points are not equidistant. The dataset is contained in the R package **FDboost**.

```
R> library(FDboost)
R> data("fuelSubset", package = "FDboost")
R> str(fuelSubset)

List of 7
 $ heatan      : num [1:129] 26.8 27.5 23.8 18.2 17.5 ...
 $ h2o         : num [1:129] 2.3 3 2 1.85 2.39 ...
 $ nir.lambda  : num [1:231] 800 803 805 808 810 ...
 $ NIR         : num [1:129, 1:231] 0.2818 0.2916 -0.0042 -0.034 -0.1804 ...
 $ uvvis.lambda: num [1:134] 250 256 261 267 273 ...
 $ UVVIS       : num [1:129, 1:134] 0.145 -1.584 -0.814 -1.311 -1.373 ...
 $ h2o.fit     : num [1:129] 2.58 3.43 1.83 2.03 3.07 ...
```

Figure 1 shows the two spectral measurements colored according to the heat value. Predictive models for the heat values, discussed in the next sections, will include scalar-on-function terms to accommodate the spectral covariates.

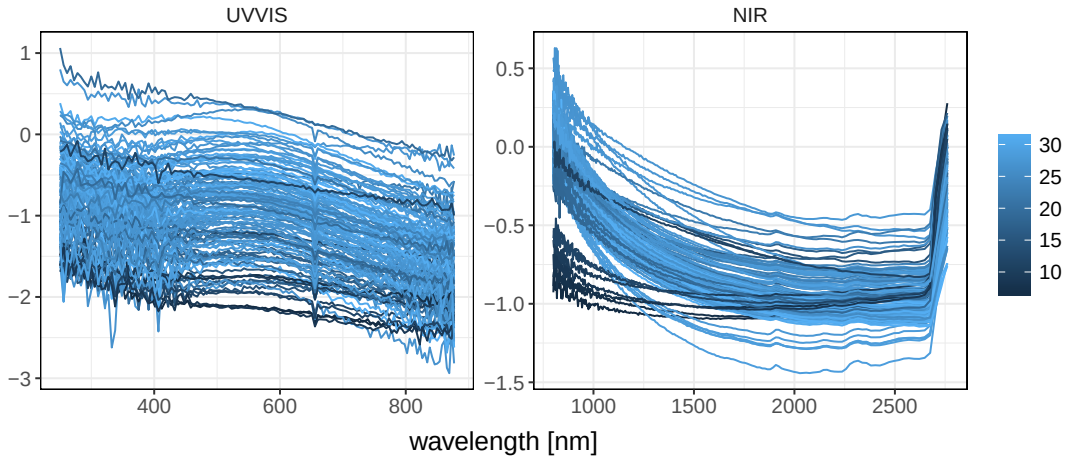

Figure 1: Spectral data of fossil fuels. Coloring of the spectral data depicts the corresponding heat value.

♦

2.2. Functional response

We denote the functional response by $Y(t)$, where t is the evaluation point at which the function is observed. We assume that $t \in \mathcal{T}$, where $\mathcal{T}$ is a real-valued interval $[T_1, T_2]$, for example a time-interval. All response curves can be observed on one common grid or on curve-specific grids. For responses observed on one common grid, we write $y_i(t_g)$ for the observations, with $t_g \in \{t_1, \ldots, t_G\}$ denoting the grid of evaluation points. For curve-specific evaluation points, the observations are denoted by $y_i(t_{ig})$, with $t_{ig} \in \{t_{i1}, \ldots, t_{iG_i}\}$. As above, the covariate set $\boldsymbol{X}$ can contain both scalar and functional variables.

As in model (1), we model the conditional expectation of the response. In this case, the expectation is modeled for each point $t \in \mathcal{T}$:

$$\mathbb{E}(Y_i(t)|\boldsymbol{X}_i = \boldsymbol{x}_i) = h(\boldsymbol{x}_i, t) = \sum_{j=1}^{J} h_j(\boldsymbol{x}_i, t). \tag{4}$$

As the response $Y_i(t)$ is a function of t, the linear predictor $h(\boldsymbol{x}_i, t)$ as well as the additive effects $h_j(\boldsymbol{x}_i, t)$ are functions of t. Each effect $h_j(\boldsymbol{x}_i, t)$ can depend on one or more covariates in $\boldsymbol{x}_i$ as well as on t. To give an idea of possible effects $h_j(\boldsymbol{x}_i, t)$, Table 2 lists some effects that are currently implemented. A function-on-function model with only one functional covariate would

covariate(s)	type of effect	$h_j(x, t)$
(none)	smooth intercept	$\beta_0(t)$
scalar covariate z	linear effect	$z\beta(t)$
	smooth effect	$f(z, t)$
two scalars z_1, z_2	linear interaction	$z_1 z_2 \beta(t)$
	functional varying coefficient	$z_1 f(z_2, t)$
	smooth interaction	$f(z_1, z_2, t)$
functional covariate $x(s)$	linear functional effect	$\int_{\mathcal{S}} x(s)\beta(s,t)\,ds$
scalar z and functional $x(s)$	linear interaction	$z \int_{\mathcal{S}} x(s)\beta(s,t)\,ds$
	smooth interaction	$\int_{\mathcal{S}} x(s)\beta(z,s,t)\,ds$
functional covariate $x(s)$,	concurrent effect	$x(t)\beta(t)$
with $\mathcal{S} = \mathcal{T} = [T_1, T_2]$	historical effect	$\int_{T_1}^{t} x(s)\beta(s,t)\,ds$
	lag effect, with lag $\delta > 0$	$\int_{t-\delta}^{t} x(s)\beta(s,t)\,ds$
	lead effect, with lead $\delta > 0$	$\int_{T_1}^{t-\delta} x(s)\beta(s,t)\,ds$
	effect with t-specific integration limits $[l(t), u(t)]$	$\int_{l(t)}^{u(t)} x(s)\beta(s,t)\,ds$
grouping variable g	group-specific smooth intercepts	$\beta_g(t)$
grouping variable g and scalar z	group-specific linear effects	$z\beta_g(t)$
curve indicator i	curve-specific smooth residuals	$e_i(t)$

Table 2: Overview of some possible covariate effects that can be represented within the framework of functional regression.

be $\mathbb{E}(Y_i|\boldsymbol{X}_i = \boldsymbol{x}_i) = \beta_0(t) + \int_{\mathcal{S}} x_i(s)\beta(s,t)\,ds$. In Section 4.2, we give several examples for concrete models with functional response.

All effects mentioned in Table 2 are varying over t but can also be modeled as constant in t. The upper part of the table contains linear, smooth and interaction effects for scalar covariates. The middle part of the table gives possible effects of functional covariates and interaction effects between scalar and functional covariates. The lower part of the table in addition shows some group-specific effects.

In practice, all effects $h_j(\boldsymbol{x}_i, t_{ig})$ are linearized using a basis representation (Brockhaus *et al.* 2017):

$$h_j(\boldsymbol{x}_i, t_{ig}) = \boldsymbol{b}_{jY}(\boldsymbol{x}_i, t_{ig})^\top \boldsymbol{\theta}_j, \quad j = 1, \ldots, J, \tag{5}$$

where the basis vector $\boldsymbol{b}_{jY}(\boldsymbol{x}_i, t_{ig}) \in \mathbb{R}^{K_{jY}}$ depends on covariates $\boldsymbol{x}_i$ and the observation-point of the response t_{ig}. The corresponding coefficient vector $\boldsymbol{\theta}_j \in \mathbb{R}^{K_{jY}}$ has to be estimated. The design matrix for the jth effect consists of rows $\boldsymbol{b}_{jY}(\boldsymbol{x}_i, t_{ig})^\top$ for all observations $i = 1, \ldots, N$ and all time-points t_{ig}, $g = 1, \ldots, G_i$.

In the following, we will use a modularization of the basis into a first part depending on covariates and a second part that only depends on t. This modular structure reduces the problem of specifying the basis $\boldsymbol{b}_{jY}(\boldsymbol{x}_i, t_{ig})$ to that of creating two suitable marginal bases. For many effects, the marginal bases are easy to define as they are known from regression with scalar response.

First, we focus on responses observed on one common grid $(t_1, \ldots, t_G)^\top$ which does not depend on i. In this case, we represent the effects using the Kronecker product $\otimes$ of two marginal bases (Brockhaus *et al.* 2015)

$$h_j(\boldsymbol{x}_i, t_g) = \left(\boldsymbol{b}_j(\boldsymbol{x}_i)^\top \otimes \boldsymbol{b}_Y(t_g)^\top\right)\boldsymbol{\theta}_j, \tag{6}$$

where the marginal basis vector $\boldsymbol{b}_j(\boldsymbol{x}_i) \in \mathbb{R}^{K_j}$, $i = 1, \ldots, N$, depends on covariates in $\boldsymbol{x}_i$ and the marginal basis vector $\boldsymbol{b}_Y(t_g) \in \mathbb{R}^{K_Y}$, $g = 1, \ldots, G$, depends on the grid point t_g. The $NG \times K_j K_Y$ design matrix is computed as the Kronecker product of the two marginal design matrices, which have dimensions $N \times K_j$ and $G \times K_Y$. If the effect can be represented as in Equation 6 it fits into the framework of linear array models (Currie, Durban, and Eilers 2006). The representation as array model has computational advantages, saving time and memory. Brockhaus *et al.* (2015) discuss array models in the context of functional regression.

Note that the representation in Equation 6 is only possible for responses observed on one common grid, as otherwise $\boldsymbol{b}_Y(t_{ig})$ depends on the curve-specific grid points t_{ig}. In this case, the marginal bases are combined by the row-wise tensor product (Scheipl *et al.* 2015; Brockhaus *et al.* 2017). This is a rather technical detail and is thoroughly explained in Brockhaus *et al.* (2017), also for the case where the basis for the covariates depends on t_{ig} such as for historical effects.

We regularize the effects by a ridge-type penalty term $\boldsymbol{\theta}_j^\top \boldsymbol{P}_{jY} \boldsymbol{\theta}_j$. The penalty matrix for the composed basis can be constructed as (Wood 2006, Sec. 4.1.8)

$$\boldsymbol{P}_{jY} = \lambda_j(\boldsymbol{P}_j \otimes \boldsymbol{I}_{K_Y}) + \lambda_Y(\boldsymbol{I}_{K_j} \otimes \boldsymbol{P}_Y), \tag{7}$$

where $\boldsymbol{P}_j = [p_{j,\varkappa,\varsigma}]_{\varkappa,\varsigma \in \{1,\ldots,K_s\}}$ is a suitable penalty for $\boldsymbol{b}_j$ and $\boldsymbol{P}_Y$ is a suitable penalty for $\boldsymbol{b}_Y$. The non-negative smoothing parameters λ_j and λ_Y determine the degree of smoothing in each direction. To illustrate the resulting penalty matrix, we explicitly compute the Kronecker products in Equation 7:

$$\boldsymbol{P}_{jY} = \lambda_j \begin{bmatrix} p_{j,1,1} \cdot \boldsymbol{I}_{K_y} & \cdots & p_{j,1,K_s} \cdot \boldsymbol{I}_{K_y} \\ \vdots & \ddots & \vdots \\ p_{j,K_s,1} \cdot \boldsymbol{I}_{K_y} & \cdots & p_{j,K_s,K_s} \cdot \boldsymbol{I}_{K_y} \end{bmatrix} + \lambda_Y \begin{bmatrix} \boldsymbol{P}_Y & & \boldsymbol{0} \\ & \ddots & \\ \boldsymbol{0} & & \boldsymbol{P}_Y \end{bmatrix}$$

This shows the block structure of the penalty matrix and how the two marginal penalty matrices are combined. The anisotropic penalty in Equation 7 can be simplified in the case of an isotropic penalty depending on only one smoothing parameter $\lambda_j \geq 0$:

$$\boldsymbol{P}_{jY} = \lambda_j(\boldsymbol{P}_j \otimes \boldsymbol{I}_{K_Y} + \boldsymbol{I}_{K_j} \otimes \boldsymbol{P}_Y). \tag{8}$$

In this simplified case only one instead of two smoothing parameters has to be estimated. If $\boldsymbol{P}_j = \boldsymbol{0}$ in Equation 8, this results in a penalty that only penalizes the marginal basis in t direction:

$$\boldsymbol{P}_{jY} = \lambda_j(\boldsymbol{I}_{K_j} \otimes \boldsymbol{P}_Y). \tag{9}$$

Consider, for example, a linear effect of a functional covariate $\int_{\mathcal{S}} x_i(s)\beta(s,t)\,ds$. The basis vector $\boldsymbol{b}_j(\boldsymbol{x}_i)$ and the penalty $\boldsymbol{P}_j$ are the same as in Equation 3. For the basis in t direction, we use a spline representation

$$\boldsymbol{b}_Y(t_g)^\top = [\phi_1(t_g) \ \cdots \ \phi_{K_Y}(t_g)] \tag{10}$$

with spline functions ϕ_k, $k = 1, \ldots, K_Y$ and the penalty matrix $\boldsymbol{P}_Y$ has to be chosen such that it is suitable for the chosen spline basis. Using P-splines again, ϕ_k are B-splines and $\boldsymbol{P}_Y$ is a squared difference matrix (Eilers and Marx 1996). The complete basis is

$$\boldsymbol{b}_j(\boldsymbol{x}_i)^\top \otimes \boldsymbol{b}_Y(t_g)^\top = \left[\int_{\mathcal{S}} x_i(s)\phi_1(s)\,ds \ \cdots \ \int_{\mathcal{S}} x_i(s)\phi_{K_j}(s)\,ds\right] \otimes \left[\phi_1(t_g) \ \cdots \ \phi_{K_Y}(t_g)\right].$$

This choice expands $\beta(s,t)$ in a tensor-product spline basis and approximates the integral using numerical integration. For this effect, the penalty matrix from Equation 7 ensures smoothness of $\beta(s,t)$ in s- and in t-direction.

Case study: Emotion components data with EEG and EMG

The emotion components data set is based on a study of Gentsch, Grandjean, and Scherer (2014), in which brain activity (EEG) as well as facial muscle activity (EMG) was simultaneously recorded during a computerised game. As the facial muscle activity should be traceable to the brain activity for a certain game situation, Rügamer *et al.* (2018) analyzed the synchronization of EEG and EMG signal using function-on-function regression models with factor-specific historical effects. During the gambling rounds, three binary game conditions were varied, resulting in a total of 8 different study settings:

- the goal conduciveness (`game_outcome`) corresponding to the monetary outcome (`gain` or `loss`) at the end of each game round,
- the `power` setting, which determined whether the player was able or not able to change the final outcome in her favor (`high` or `low`, respectively) and,
- the `control` setting, which was manipulated to change the participant's subjective feeling about her ability to cope with the game outcome. The player was told to frequently have high power in rounds with `high` control and have frequently low power in `low` control situations.

We focus on the EMG of the frontalis muscle, which is used to raise the eyebrow. The EMG signal is a functional response $Y(t)$, with $t \in \mathcal{T} = [0, 1560]$ ms, which is measured at a frequency of $256Hz$ resulting in 384 equidistant observed time points given by the vector `t`. The experimental conditions are scalar covariates. The EEG signal $x_{\text{EEG}}(s)$ is observed over the same time interval as the EMG signal. We use the EEG signal from the Fz electrode, which is in the center front of the head.

In the following, we consider an aggregated version of the data, in which the EEG and EMG signals are aggregated per subject and game condition. One participant is excluded, yielding $N = 23$ subjects.

```
R> data("emotion", package = "FDboost")
R> str(emotion)
```

```
List of 8
 $ power       : Factor w/ 2 levels "high","low": 1 1 2 2 1 1 2 2 1 1 ...
 $ game_outcome: Factor w/ 2 levels "gain","loss": 1 2 1 2 1 2 1 2 1 2 ...
 $ control     : Factor w/ 2 levels "high","low": 1 1 1 1 2 2 2 2 1 1 ...
 $ subject     : Factor w/ 23 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 2 2 ...
 $ EEG         : num [1:184, 1:384] -0.14 0.303 -0.715 0.7 0.11 ...
 $ EMG         : num [1:184, 1:384] -2.56 -4.06 -1.15 4.11 8.09 ...
 $ s           : int [1:384] 1 2 3 4 5 6 7 8 9 10 ...
 $ t           : int [1:384] 1 2 3 4 5 6 7 8 9 10 ...
```

In order to fit simple and meaningful models for function-on-function regression, we define a subset of the data that contains only the observations for a certain game condition. We use the game condition with high control, gain and low power:

```
R> subset <- emotion$control == "high" &
+   emotion$game_outcome == "gain" &
+   emotion$power == "low"
R> emotionHGL <- list()
R> emotionHGL$subject <- emotion$subject[subset]
R> emotionHGL$EMG <- emotion$EMG[subset,]
R> emotionHGL$EEG <- emotion$EEG[subset,]
R> emotionHGL$s <- emotionHGL$t <- emotion$t
```

In Figure 2 the EEG and EMG signal is depicted for each of the 23 participants and the 384 observation points.

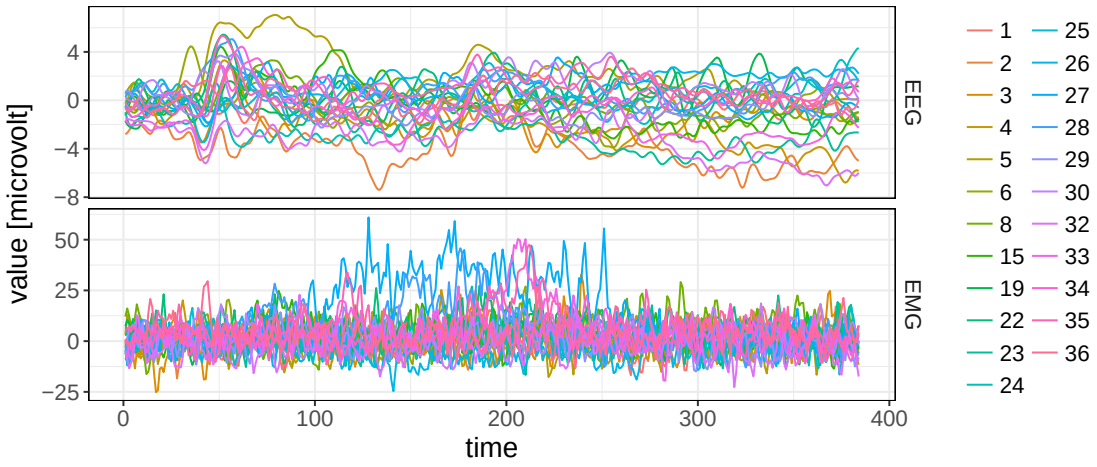

Figure 2: EEG signal (Fz electrode) and EMG signal (frontalis muscle) for each of the 23 participants (line colours) and the chosen game condition.

♦

3. Estimation by gradient boosting

Initially, boosting was proposed as a technique to iteratively improve the predictive performance of simple models or *base-learners* (Ridgeway 1999). Boosting was soon recognized as a model fitting technique for statistical applications. Based on the idea of Friedman (2001), Bühlmann and Hothorn (2007) proposed the model-based boosting framework, which allows for a component-wise fitting of additive terms in the linear predictor and can handle complex additive effects. Many boosting algorithms, which are purely used for prediction, fit a rather simple model using all covariates. In contrast, in model-based boosting it is possible to define the effects of each covariate separately in different base-learners. By iteratively selecting only one base-learner at a time, model-based boosting performs variable selection as base-learners that are never selected for the model update are excluded from the model. This framework is implemented in the `mboost` package. In contrast to other implementations of gradient boosting, such as `gbm` (Ridgeway 2017), the focus of model-based boosting lies in estimating an interpretable additive structure rather than aiming at optimal predictive performance.

Component-wise gradient boosting minimizes the expected loss (risk) via gradient descent in a step-wise procedure. In each boosting step, each base-learner is fitted separately to the negative gradient and only the best fitting base-learner is selected for the model update; hence the term 'component-wise'. To fit a model for the expectation, like the models in Equation 1 and 4, the squared error loss (L_2 loss) is minimized. In this case, the negative gradient corresponds to the residuals.

Resulting estimation and prediction performance of boosting depend on different tuning parameters, namely the *number of boosting iterations* m_{stop}, the *step-length* ν, and the specification of the base-learners, e.g., whether a continuous covariate has a linear or smooth effect and the set-up of spline functions and penalties for smooth effects. We will give guidance on the choice of these parameters in the following by briefly describing the functionality of the algorithm.

The most important tuning parameter of boosting is the number of boosting iterations, as the algorithm is usually stopped before convergence. This so-called early stopping leads to regularized effect estimates and therefore yields more stable predictions. Since some of the base-learners are never selected in the course of all iterations, boosting also performs variable selection. The optimal stopping iteration can be determined by methods like cross-validation, sub-sampling or bootstrap. For each fold, the empirical out-of-bag risk is computed and the stopping iteration that yields the lowest empirical risk is chosen. As resampling must be conducted on the level of independent observations, this is done on the level of curves for functional response.

In order to avoid overshooting the minimum of the loss function in each iteration, only a small step in the chosen direction is made. The length of the update is determined by the step-length ν. Some boosting frameworks adapt the choice of the step-length in each iteration. Bühlmann and Hothorn (2007) show that the estimation performance is barely affected by setting ν to a fixed and sufficiently small value for all iterations. They there propose to use a fixed step-length in the range 0.01 to 0.1. The appropriate size of the step-length depends on the loss that is minimized. In practice, the default value $\nu = 0.1$ works well for most applications when the model is specified using the L_2-loss. A smaller step-length than 0.01 is sometimes needed for loss functions, which result in discontinuous gradients, such as the check-function for quantile regression (Fenske, Kneib, and Hothorn 2011) or for loss functions, which can result in infinite pseudo-residuals, such as the Poisson likelihood loss. Since base-learner-specific tuning parameter are fixed for all iterations, the model fit is determined by the number of iterations for a given step-length.

By representing all base-learners as linear effects of covariates (if necessary, by using a basis representation for non-linear effects), base-learners also define the covariate effects in the sense of

additive regression models and can be associated with a specific hat matrix as well as a certain number of degrees of freedom. The degrees of freedom for each base-learner and other base-learner-specific tuning parameters have an influence on the prediction and estimation performance. The degrees of freedom df_j for each base-learner $j = 1, \ldots, J$ – not to be confused with the *effective degrees of freedom* for each model term in the final model – determine the flexibility of each base-learner prior to the model fit. In the model-based boosting framework each base-learner is fitted to the pseudo-residuals using a (penalized) least squares fit with fixed smoothing parameter λ_j, which is determined via the pre-specified degrees of freedom. Whereas defining a fixed smoothness for each model term prior to the model fit might seem restrictive at first sight, the final smoothness of each model term is in fact determined through the number of iterations in which the respective base-learner is chosen. The *effective degrees of freedom* for each smooth component after the model fit are cumulated over the iterations where the model term is selected and typically differ from the initially specified df_j. The model fit can thus adapt even to relatively complex functions by repeatedly selecting and updating a particular model term (cf. Brockhaus *et al.* 2015). Determining the smoothness through the number of iterations works well in practice and allows for a closed-form solution of the penalized least squares fit in each update. As boosting chooses base-learners in a greedy manner, selection in each step is biased towards more flexible base-learners with higher degrees of freedom, if base-learners exhibit different degrees of freedom. This is due to the fact that these base-learner more likely yield larger improvements of the fit in each iteration (see Hofner, Hothorn, Kneib, and Schmid 2011, for details). For parameter estimation quality, it is essential to facilitate a fair base-learner selection in each step (Hofner *et al.* 2011). It is recommended to set df_j to an equal and rather small number for all base-learners $j = 1, \ldots, J$ (Kneib, Hothorn, and Tutz 2009; Hofner *et al.* 2011). In the case of scalar-on-function regression, fulfilling this constraint is not straightforward as functional covariates must usually be incorporated with more than one degree of freedom whereas scalar linear effects are restricted to have one degree of freedom. In order to maintain a fair base-learner selection, more complex effects can be orthogonalized such that they represent deviations from less complex effects. For example, a smooth effect can be centered around its linear effect, thereby allowing both terms to have one degree of freedom. In Section 4.3 as well as in Appendix E different examples demonstrate how to facilitate a fair selection in this respect.

Due to the nature of the algorithm, other base-learner-specific tuning parameters are also defined prior to the model fit and kept fixed over the iterations. The number of knots is of primary interest for functional or smooth predictors and should be chosen considering as a trade-off between computing time and flexibility of each base-learner. Per default, 10 knots are used, which can be rather large for some applications, but allows for a large flexibility of the estimated effects. The number of knots can be decreased if computing time is a concern. Moreover, due to the smoothness penalty, with the default penalizing deviations from linearity for smooth functions, users need not to be concerned about overfitting when increasing the number of knots.

Functional Response

To adapt boosting for a functional response, we compute the loss at each point t and integrate it over the domain of the response $\mathcal{T}$ (Brockhaus *et al.* 2015).

For the L_2 loss the optimization problem for functional response aims at minimizing

$$\sum_{i=1}^{N} \int \left[y_i(t) - h(\boldsymbol{x}_i, t) \right]^2 dt, \tag{11}$$

which is approximated by numerical integration. To obtain identifiable models, suitable identifiability constraints for the base-learners are necessary and implemented. **FDboost** also contains base-learners that model the effects of functional covariates. For a discussion of both points, please see Brockhaus *et al.* (2015).

4. The package FDboost

Fitting functional regression models via boosting is implemented in the R package **FDboost**. The package uses the fitting algorithm and other infrastructure from the R package **mboost** (Hothorn *et al.* 2016). All base-learners and distribution families that are implemented in **mboost** can be used within **FDboost**. Many naming conventions and methods in **FDboost** are implemented in analogy to **mboost**. A tutorial for **mboost** can be found in Hofner, Mayr, Robinzonov, and Schmid (2014). We will mention all features of **mboost** that are important when working with **FDboost** in the following.

The main fitting function to estimate functional regression models, like the models in Equation 1 and 4, is called `FDboost()`. The interface of `FDboost()` is as follows:[1]

```
R> FDboost(formula, timeformula, id = NULL, numInt = "equal",
+   data, offset = NULL, ...)
```

First, we focus on the arguments that are necessary for regression models both with scalar and with functional response. `formula` specifies the base-learners for the covariate effects $\boldsymbol{b}_j$ and `timeformula` specifies $\boldsymbol{b}_Y$, which is the basis along t. Per default, this basis $\boldsymbol{b}_Y$ is the same for all effects $j = 1, \ldots, J$. To specify different base-learners along t, it is necessary to set up the Kronecker product of two base-learners explicitly in `formula`. For a detailed explanation, we refer to Appendix C. The data is provided in the `data` argument as a `data.frame` or a named `list`. The `data`-object has to contain the response, all covariates and the evaluation points of functional variables. Prior to the model fit, an offset is subtracted from the response to center it. This corresponds to initializing the fit with this offset, e.g., an overall average, and leads to faster convergence and better stability of the boosting algorithm. For mean regression, by default the offset is the smoothed point-wise mean of the response over time without taking into account covariates. This offset is part of the intercept and corresponds to an initial estimate that is then updated. In the dots-argument, '`...`', further arguments passed to `mboost()` and `mboost_fit()` can be specified. The most important argument is `family` determining the loss- and link-function for the model fit. The default is `family = Gaussian()`, which minimizes the squared error loss and uses the identity as link function. Thus, per default a mean regression model for continuous response is fitted. For the duality of loss-function and the `family` argument, we refer to Section 4.3. Further important arguments are `control`, which determines the number of boosting iterations and the step-length ν of the boosting algorithm specified by `nu`. The argument `control` must be supplied as a call to the function `boost_control()`. For example, `control = boost_control(mstop = 100, nu = 0.1)` implies 100 boosting iterations and step-length $\nu = 0.1$, which also corresponds to the default settings. Note that while 100 iterations are the default chosen to avoid a computationally expensive default, this might not be sufficient and should be chosen appropriately for the given application.

[1] Note that for the presentation of functions we restrict ourselves to the most important function arguments. For the full list of arguments, we refer to the corresponding manuals.

FDboost allows for (tensor product) spline or functional principle component bases, but user-specified base-learner allow for possible extensions (see, e.g. Hofner *et al.* 2014). Although the package only provides base-learners with ridge- or L_2-type penalization, model selection as facilitated by an L_1-penalty is achieved by early stopping of the algorithm. The covariance of final effects results from the additive fit with Kronecker separable penalty structure. Dependent functions can be modelled by including regularized cluster-specific functional intercepts or smooth temporal / spatial effects.

Specification for scalar response

For scalar response, we set `timeformula = NULL` as no expansion of the effects in t direction is necessary. `formula` specifies the base-learners for the covariates effects $\boldsymbol{b}_j$ as in Equation 2. The arguments `id` and `numInt` are only needed for functional responses. For scalar response, `offset = NULL` results in a default offset, as, for example, the overall mean for mean regression.

Arguments needed for functional response

For functional response, the set-up of the covariate effects generally follows Equation 6 by separating the effects into two marginal parts. The marginal effects $\boldsymbol{b}_j$, $j = 1, \ldots, J$, are represented in the `formula` as `y ~ b_1 + b_2 + ...+ b_J`. The marginal effect $\boldsymbol{b}_Y$ is represented in the `timeformula`, which has the form `~ b_Y`. The base-learners for the marginal effects also contain suitable penalty matrices. Internally, the base-learners specified in `formula` are combined with the base-learner specified in `timeformula` as in Equation 6 and a suitable penalty matrix is constructed according to Equation 8. Per default, the response is expected to be a matrix. In this case `id = NULL`. The matrix representation is not possible for a response which is observed on curve specific grids. In this case the response is provided as vector in long format and `id` specifies which position in the vector is attributed to which curve; see section 4.2 for details. The argument `numInt` provides the numerical integration scheme for computing the integral of the loss over $\mathcal{T}$ in Equation 11. Per default, `numInt = "equal"`, and thus all integration weights are set to one; for `numInt = "Riemann"` Riemann sums are used. For functional response, `offset = NULL` induces a smooth offset varying over t. For `offset = "scalar"`, a scalar offset is computed. This corresponds to an offset that is constant along t. For more details and the full list of arguments, see the manual of `FDboost()`.

4.1. Scalar response and functional covariates

In this subsection, we give details on models with scalar response and functional covariates like the model in Equation 1. Such models are called scalar-on-function regression models. As case study the data on fossil fuels is used.

Potential covariate effects: base-learners

In order to fit a scalar-on-function model as in Equation 1, the `timeformula` is set to `NULL` and potential covariate effects $h_j(\boldsymbol{x}_i)$ are specified in the `formula` argument. The effects of scalar covariates can be linear or non-linear. A linear effect $z\beta$ for the covariate z is obtained using the base-learner `bols(z)`, which is also suitable for factor variables, in which case dummy variables are constructed for each factor level (Hofner *et al.* 2014). Per default, `bols()` contains an intercept. If the specified degrees of freedom are less than the number of columns in the design matrix, `bols()` penalizes the linear effect by a ridge penalty with the identity matrix as penalty matrix. The

base-learner `brandom()` for factor variables sets up an effect, which is centered around zero and is penalized by a ridge penalty, having similar properties to a random effect, but no underlying distributional assumption. It is not possible to estimate random effects in the classical sense that they are estimated using variance parameters. See the web appendix of Kneib *et al.* (2009) for a discussion on `brandom()`. The ridge penalized effects, however, have a similar interpretation as random effects as a quadratic penalty is mathematically equivalent to a Gaussian prior. Note that this also allows for other types of random effects such as cluster-specific random effect functions. A non-linear effect expanded by P-splines is obtained by the base-learner `bbs()`. Within `bbs()`, the argument `knots` determines the number of knots of the P-spline basis, `degree` specifies the degree of the spline basis and `differences` the order of the differences in the penalty matrix. Per default, cubic B-splines on 20 knots with a second order difference penalty are used. For more details on base-learners with scalar covariates, we refer to Hofner *et al.* (2014).

Potential base-learners for functional covariates can be seen in Table 3. In this table exemplary linear predictors are listed in the left column. In the right column, the corresponding call to `formula` is given. Because of the scalar response, the call to `timeformula` is set to `NULL`. For simplicity, only one possible parameterization which leads to simple interpretations and one corresponding model call are shown, although **FDboost** allows to specify several parameterizations.

additive predictor $h(\boldsymbol{x}) = \sum_j h_j(\boldsymbol{x})$	call
$\beta_0 + \int_\mathcal{S} x(s)\beta_1(s)\,ds$	`y ~ 1 + bsignal(x, s = s)`
	`y ~ 1 + bfpc(x, s = s)`
$\beta_0 + z\beta_1 + \int_\mathcal{S} x(s)\beta_2(s)\,ds$	`y ~ 1 + bolsc(z) + bsignal(x, s = s)`
$\quad + z\int_\mathcal{S} x(s)\beta_3(s)\,ds$	`+ bsignal(x, s = s) %X% bolsc(z)`

Table 3: Additive predictors for scalar-on-function regression models.

For a linear effect of a functional covariate $\int_\mathcal{S} x(s)\beta_1(s)\,ds$, two base-learners exist that use different basis expansions. Assuming $\beta_1(s)$ to be smooth, `bsignal()` uses a P-spline representation for the expansion of $\beta_1(s)$. In this case, the observations $x(s)$ are used directly without any basis representation. Assuming that the main modes of variation in the functional covariate are the important directions for the coefficient function $\beta_1(s)$, a representation with functional principal components is suitable (Ramsay and Silverman 2005). In the base-learner `bfpc()`, the coefficient function $\beta_1(s)$ and the functional covariate $x(s)$ are both represented by an expansion in the estimated functional principal components of $x(s)$. As penalty matrix, the identity matrix is used. In Appendix B, technical details on the representation of functional effects are given.

The specification of a model with an interaction term between a scalar and a functional covariate is given at the end of Table 3. The interaction term is centered around the main effect of the functional covariate using `bolsc` for the scalar covariate (as is the linear effect of the scalar covariate around the intercept). Thus, the main effect of the functional covariate has to be included in the model. For more details on interaction effects, we refer to Brockhaus *et al.* (2015) and Rügamer *et al.* (2018). The interaction is formed using the operator `%X%` that builds the row-wise tensor product of the two marginal bases, see Appendix C.

As explained in Section 3, all base-learners in a model should have equal and rather low degrees of freedom. The number of degrees of freedom that can be given to a base-learner is restricted. On

the one hand, the maximum number is bounded by the number of columns of the design matrix (more precisely by the rank of the design matrix). On the other hand, for rank-deficient penalties, the minimum number of degrees of freedom is given by the rank of the null space of the penalty matrix.

The interface of `bsignal()` is as follows:

```
R> bsignal(x, s, knots = 10, degree = 3, differences = 1,
+   df = 4, lambda = NULL, check.ident = FALSE)
```

The arguments `x` and `s` specify the name of the functional covariate and the name of its argument. `knots` gives the number of inner knots for the P-spline basis, `degree` the degree of the B-splines and `differences` the order of the differences that are used for the penalty. Thus, per default, 14 cubic P-splines with first order difference penalty are used. The argument `df` specifies the number of degrees of freedom for the effect and `lambda` the smoothing parameter. Only one of those two arguments can be supplied. If `check.ident = TRUE` identifiability checks proposed by Scheipl and Greven (2016) for functional linear effects are additionally performed.

The interface of `bfpc()` is:

```
R> bfpc(x, s, df = 4, lambda = NULL, pve = 0.99, npc = NULL)
```

The arguments `x, s, df` and `lambda` have the same meaning as in `bsignal()`. The two other arguments allow to control how many functional principal components are used as basis. Per default the number of functional principal components is chosen such that the proportion of the explained variance is 99%. This proportion can be changed using the argument `pve` (proportion variance explained). Alternatively, the number of components can be set to a specific value using `npc` (number principal components).

The interface of `bolsc()` is very similar to that of `bols()`, which is laid out in detail in Hofner *et al.* (2014). In contrast to `bols()`, `bolsc()` centers the design matrix such that the resulting linear effect is centered around zero. More details on `bolsc()` are given in Section 4.2.

```
R> bolsc(..., df = NULL, lambda = 0, K = NULL)
```

In the dots argument, `...`, one or more covariates can be specified. For factor variables `bolsc()` sets up a design matrix in dummy-coding. The arguments `df` and `lambda` have the same meaning as above. If `lambda` > 0 or `df` $<$ the number of columns of the design matrix a ridge-penalty is applied. Per default, `K = NULL`, the penalty matrix is the identity matrix. Setting the argument `K` to another matrix allows for customized penalty matrices.

Case study (ctd.): Fossil fuel data

For the heat values Y_i, $i = 1, \ldots, 129$, we fit the model

$$\mathbb{E}(Y|\boldsymbol{x}) = \beta_0 + f(z_{\text{H2O}}) + \int_{\mathcal{S}_{\text{NIR}}} x_{\text{NIR}}(s_{\text{NIR}})\beta_{\text{NIR}}(s_{\text{NIR}})\, ds_{\text{NIR}} + \int_{\mathcal{S}_{\text{UV}}} x_{\text{UV}}(s_{\text{UV}})\beta_{\text{UV}}(s_{\text{UV}})\, ds_{\text{UV}}, \qquad (12)$$

with water content z_{H2O} and centered spectral curves x_{NIR} and x_{UV}, which are observed over the wavelengths $s_{\text{NIR}} \in \mathcal{S}_{\text{NIR}}$ and $s_{\text{UV}} \in \mathcal{S}_{\text{UV}}$. We center the NIR and the UVVIS measurement per

wavelength such that $\sum_{i=1}^{N} x_{\text{NIR},i}(s_{\text{NIR}}) = 0 \, \forall s_{\text{NIR}}$ and analogously for UVVIS. Thus, the functional effects have mean zero, $\sum_{i=1}^{N} \int_{\mathcal{S}_{\text{NIR}}} x_{\text{NIR},i}(s_{\text{NIR}}) \beta(s_{\text{NIR}}) \, ds_{\text{NIR}} = 0$ and analogously for UVVIS. This does not affect the interpretation of $\beta_{\text{NIR}}(s_{\text{NIR}})$ and $\beta_{\text{UV}}(s_{\text{UV}})$, it only changes the interpretation of the intercept of the regression model. If all effects are centered, the intercept can be interpreted as overall mean and the other effects as deviations from the overall mean.

Note that the functional covariates have to be supplied as <number of curves> by <number of evaluation points> matrices. The non-linear effect of the scalar variable H2O is specified using the `bbs()` base-learner. For the linear functional effect of NIR and UVVIS, we use the base-learner `bsignal()`. The degrees of freedom are set to 4 for each base-learner. For the functional effects, we use a P-spline basis with 20 inner knots. Because of the scalar response `timeformula = NULL`.

```
R> fuelSubset$UVVIS <- scale(fuelSubset$UVVIS, scale = FALSE)
R> fuelSubset$NIR <- scale(fuelSubset$NIR, scale = FALSE)
R> sof <- FDboost(heatan ~ bbs(h2o, df = 4)
+   + bsignal(UVVIS, s = uvvis.lambda, knots = 20, df = 4)
+   + bsignal(NIR, s = nir.lambda, knots = 20, df = 4),
+   timeformula = NULL, data = fuelSubset)
```

♦

Model tuning and early stopping

Boosting iteratively selects base-learners to update the additive predictor. Fixing the base-learners and the step-length, the model complexity is controlled by the number of boosting iterations. With more boosting iterations the model becomes more complex (Bühlmann and Yu 2003). The step-length ν is chosen sufficiently small in the interval $(0, 1]$, usually as $\nu = 0.1$, which is also the default. For smaller step-length, more boosting iterations are required and vice versa (Friedman 2001). Note that the default number of boosting iterations is 100. This is arbitrary and in most cases not adequate. The number of boosting iterations and the step-length of the algorithm can be specified in the argument `control`. This argument must be supplied as a call to `boost_control()`. For example, `control = boost_control(mstop = 50, nu = 0.2)` implies 50 boosting iterations and step-length $\nu = 0.2$.

The most important tuning parameter is the number of boosting iterations. For regression with scalar response, the function `cvrisk.FDboost()` can be used to determine the optimal stopping iteration. This function directly calls `cvrisk.mboost()` from the **mboost** package, which performs an empirical risk estimation using a specified resampling method. The interface of `cvrisk.FDboost()` is:

```
R> cvrisk.FDboost(object,
+   folds = cvLong(id = object$id, weights = model.weights(object)),
+   grid = 1:mstop(object))
```

In the argument `object`, the fitted model object is specified. `grid` defines the grid on which the optimal stopping iteration is searched. Per default the grid from 1 to the current stopping iteration of the model object is used as search grid. But it is also possible to specify a larger grid, e.g., `1:5000`. The argument `folds` expects an integer weight matrix with dimension $N \times \kappa$ (<number of observations> times <number of folds>). Depending on the range of values in the

weight matrix, different types of resampling are performed. For example, if the weights sum to N for each column but also have values larger than one, the resampling scheme corresponds to bootstrap while a κ-fold cross-validation is employed by using an incidence matrix, for which the rows sum to $\kappa - 1$. If not manually specified, **mboost** and **FDboost** provide convenience functions – `cv()` and `cvLong()` – that construct such matrices on the basis of the given model object. The function `cvLong()` is suited for functional response and treats scalar response as the special case with one observation per curve. For scalar response, the function `cv()` from package **mboost** can be used, which has a simpler interface.

```
R> cv(weights, type = c("bootstrap", "kfold", "subsampling"),
+   B = ifelse(type == "kfold", 10, 25))
```

The argument `weights` is used to specify the weights of the original model, which can be extracted using `model.weights(object)`. Usually all model weights are one. Via argument `type` the resampling scheme is defined: `"bootstrap"` for non-parametric bootstrap, `"kfold"` for cross-validation and `"subsampling"` for resampling half of all observations for each fold. The number of folds is defined by `B`. Per default, 10 folds are used for cross-validation and 25 folds for bootstrap as well as for subsampling.

The function `cvLong()` is especially suited for functional response and has the additional argument `id`, which is used to specify which observations belong to the same response curve. For scalar response, `id = 1:N`.

Case study (ctd.): Fossil fuel data

To tune the scalar-on-function regression model (12), we search the optimal stopping iteration by 10-fold bootstrapping. First, the bootstrap folds are created using the function `cv()`. Second, for each bootstrap fold, the out-of-bag risk is computed for models with 1 to 1000 boosting iterations using the `cvrisk` function. The choice of the grid is independent of the number of boosting iterations of the fitted model object.

```
R> set.seed(123)
R> folds_sof <- cv(weights = model.weights(sof), type = "bootstrap", B = 10)
R> cvm_sof <- cvrisk(sof, folds = folds_sof, grid = 1:1000)
```

The object `cvm_sof` contains the out-of-bag risk of each fold for all 1000 iterations. ⧫

Methods to extract and visualize results from the resampling object

For a `cvrisk`-object as created by `cvrisk()`, the method `mstop()` extracts the estimated optimal number of boosting iterations, which corresponds to the number of boosting iterations yielding the minimal mean out-of-bag risk. `plot()` generates a plot of the estimated out-of-bag risk per stopping iteration in each fold. In addition, the mean out-of-bag risk per stopping iteration is displayed. The estimated optimal stopping iteration is marked by a dashed vertical line. In such a plot, the convergence behavior can be graphically examined.

Case study (ctd.): Fossil fuel data

We generate a plot that displays for each fold the estimated out-of-bag risk per stopping iteration for each fold; see Figure 3.

```
R> plot(cvm_sof, ylim = c(2, 15))
```

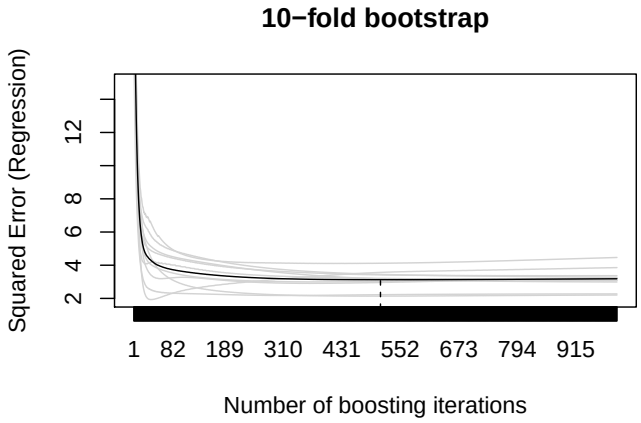

Figure 3: Bootstrapped out-of-bag risk for the model of the fossil fuels. For each fold, the out-of-bag risk is displayed as a gray line. The mean out-of-bag risk is visualized by a black line. The optimal number of boosting iterations is marked by a dashed vertical line.

For small numbers of boosting iterations, the out-of-bag risk declines sharply with a growing number of boosting iterations. With more and more iterations the model gets more complex and the out-of-bag risk starts to slowly increase. The dashed vertical line marks the estimated optimal stopping iteration of 511, which can be accessed using the function `mstop()`:

```
R> mstop(cvm_sof)

[1] 511
```

♦

Methods to extract and display results from the model object

Fitted `FDboost` objects inherit methods from class `mboost`. Thus, all methods available for `mboost` objects can also be applied to models fitted by `FDboost()`. The design and penalty matrices that are constructed by the base-learners can be extracted using the `extract()` function. For example, `extract(object, which = 1)` returns the design matrix of the first base-learner and `extract(object, which = 1, what = "penalty")` the corresponding penalty matrix. The number of boosting iterations for an `FDboost` object can be changed afterwards using the subset operator; e.g., `object[50]` sets the number of boosting iterations for `object` to 50. Note that the subset operator directly changes `object`, and hence no assignment is necessary.

One can access the estimated coefficients by the `coef()` function. The function takes a fitted `object` produced by `FDboost()` and returns estimated coefficient functions such as $\hat{\beta}(s)$, $\hat{\beta}(s,t)$, $\hat{g}(x)$ or other estimated effects. For smooth effects, `coef()` returns the smooth estimated effects evaluated on a regular grid. The resolution of the grid can be specified by the arguments `n1,`

`n2` and `n3` for 1-, 2- and 3-dimensional smooth terms, respectively, which define the number of equidistantly spaced grid points over the range of the covariate. The resulting object is a list containing an element for the offset and a named list with one entry for each further model term. The value of the offset for each observation can be accessed with `coef(object)$offset$value`. List entries for model terms in `coef(object)$smterms` are, in turn, lists with different entries, in particular, including `$x` (`$y, $z`) representing unique grid-points used to evaluate the coefficient function and `$value` representing a vector, matrix or list of matrices with the coefficient values. The estimated spline-coefficients $\hat{\boldsymbol{\theta}}_j$ of smooth effects can be obtained by `object$coef()`, which is equal to setting the argument `raw` to `TRUE` in the `coef` function.

The estimated effects can be graphically displayed by the `plot()` function. The coefficient plots can be customized by various arguments. For example, coefficient surfaces can be displayed as image plots, setting `pers = FALSE`, or as perspective plots, setting `pers = TRUE`. To plot only some of the base-learners, the argument `which` can be used. For instance, `plot(object, which = c(1,3))` plots the estimated effects of the first and the third base-learner. The fitted values and predictions for new data can be obtained by the methods `fitted()` and `predict()`, respectively.

Case study (ctd.): Fossil fuel data

To better understand the penalization used in the `sof` model, we can exemplarily extract the marginal penalty matrix for UVVIS as follows:

```
R> marg_pen <- extract(sof, "penalty", which = 2)
R> marg_pen[[1]][1:5,1:5]

     [,1] [,2] [,3] [,4] [,5]
[1,]    1   -1    0    0    0
[2,]   -1    2   -1    0    0
[3,]    0   -1    2   -1    0
[4,]    0    0   -1    2   -1
[5,]    0    0    0   -1    2
```

In order to continue working with the optimal model, we set the number of boosting iterations to the estimated optimal value.

```
R> sof <- sof[mstop(cvm_sof)]
```

We can access estimated coefficients using `coef()`, e.g., by extracting the estimated coefficient function $\hat{\beta}_{\text{NIR}}(s_{\text{NIR}})$ contained in `$value` evaluated at grid points `$x`

```
R> coef_sof <- coef(sof)
R> str(coef_sof$smterms$`bsignal(NIR)`)
```

To display the estimated effects, `plot()` can be called on the fitted FDboost object.

Per default, `plot()` only displays effects of base-learners that were selected at least once. See Figure 4 for the resulting plots.

```
R> par(mfrow = c(1,3))
R> plot(sof, ask = FALSE, ylab = "")
```

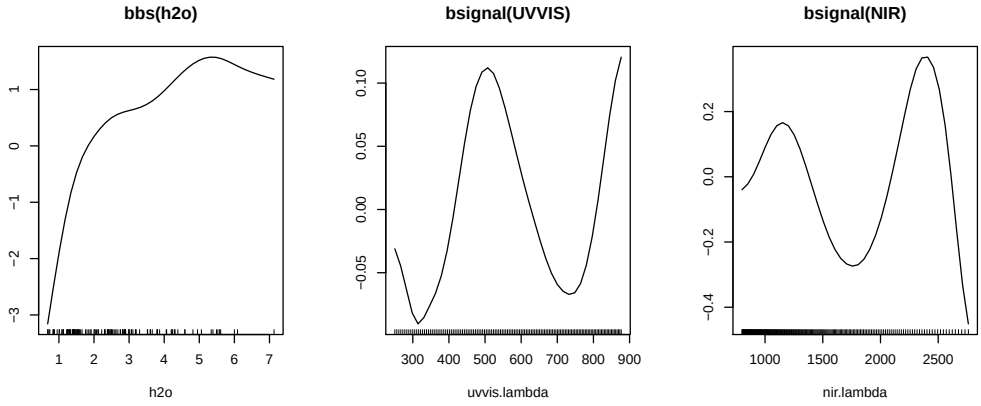

Figure 4: Coefficient estimates of the model for the heat value of the fossil fuels with optimal number of boosting iterations. The smooth effect of the water content (left), the linear effect of the UVVIS spectrum (center) and the NIR spectrum (right) are displayed.

The mean heat value is estimated to be higher for higher water content and lower for lower water content (see Figure 4 left). High values of the UVVIS spectrum at a wavelength of around 500 and 850 nm are associated with higher heat values. Higher values of the UVVIS spectrum at wavelength around 300 and 750 nm are associated with lower heat values (see Figure 4 middle). The effect of the NIR spectrum can be interpreted analogously. ⧫

Bootstrapped coefficient estimates

In order to get a measure for the uncertainty associated with the estimated coefficient functions, one can employ nested bootstrap. The optimal number of boosting iterations in each bootstrap fold, in turn, is estimated by an inner resampling procedure. The bootstrapped coefficients are shrunken towards zero as boosting shrinks coefficients towards zero due to early stopping. Thus, the resulting bootstrap "confidence" interval is biased towards zero but still captures the variability of the coefficient estimates. While they do not have proper coverage properties due to shrinkage bias, these bootstrap intervals capture all the sources of uncertainty (induced by the resampling, the model selection as well as the actual uncertainty of coefficients). They may be used to check, e.g., for the existence of certain effects by examining whether the resulting intervals contain the value zero, which was found to work well in Rügamer *et al.* (2018). Having no formal inference procedure clearly is a limitation of the model-based boosting framework in general and users who want to formally test pre-specified hypotheses are referred to alternative software packages such as `refund` (Huang, Scheipl, Goldsmith, Gellar, Harezlak, McLean, Swihart, Xiao, Crainiceanu, and Reiss 2016) for cases where these are applicable and the particular strengths of model-based boosting (high-dimensional data and models, model selection, general loss-functions) are not needed. In **FDboost** the function `bootstrapCI()` can be used to conveniently compute bootstrapped coefficients:

```
R> bootstrapCI(object, B_outer = 100, B_inner = 25, ...)
```

The argument `object` is the fitted model object. The maximal number of boosting iterations for each bootstrap fold is the number of boosting iterations of the model-object. Per default bootstrap is used with `B_outer = 100` outer folds and `B_inner = 25` inner folds. The dots argument, `...` can be used to pass further arguments to `applyFolds()`, which is used for the outer bootstrap. In particular, setting the argument `mc.cores` to an integer greater 1 will run the outer bootstrap in parallel on the number of cores that are specified via `mc.cores` (this does not work under Windows, as the parallelization is based on the function `mclapply()`). As for the resampling scheme, which determines the number of iterations, the bootstrap which is done to quantify uncertainty of coefficient estimates should be conducted on the level of independent observations. This is particularly relevant for functional responses, where both resampling procedures should be done on the level of curves. Additional dependence in the data, such as observations sampled from clusters or in a longitudinal fashion, should also be taken into account for scalar-on-function models. To this end, observations should be sampled on the levels of clusters, subjects, or in nested designs, by a nested sampling for each of the levels. This yields a limitation of our method in cases, in which observations can not be separated into independent units (e.g., for spatially correlated observations with a strong dependence among all observations). However, costumized solutions such as a block-wise bootstrap (cf. Brockhaus *et al.* 2018) for time-series data can be employed as in the scalar case.

Case study (ctd.): Fossil fuel data

We recompute the model on 100 bootstrap samples to compute bootstrapped coefficient estimates. In each bootstrap fold the optimal number of boosting iterations is estimated by an inner bootstrap with 10 folds. In contrast to other methods and analytic inference concepts, employing bootstrap for coefficient uncertainty is much more time consuming but can be easily parallelized. See the help page of `bootstrapCI()` for example code. The resulting estimated coefficients can be seen in Figure 5.

```
R> set.seed(123)
R> sof_bootstrapCI <- bootstrapCI(sof[1000], B_outer = 100, B_inner = 10,
+   mc.cores = 10)
R> par(mfrow = c(1,3))
R> plot(sof_bootstrapCI, ask = FALSE, commonRange = FALSE, ylab = "")
```

♦

4.2. Functional response

In this subsection, we explain how to fit models with functional response like model (4). Models with scalar and functional covariates are treated, thus covering function-on-scalar and function-on-function regression models.

Specification of functional response

If a functional variable is observed on one common grid, its observations can be represented by a matrix. In **FDboost**, such functional variables have to be supplied as <number of curves> by

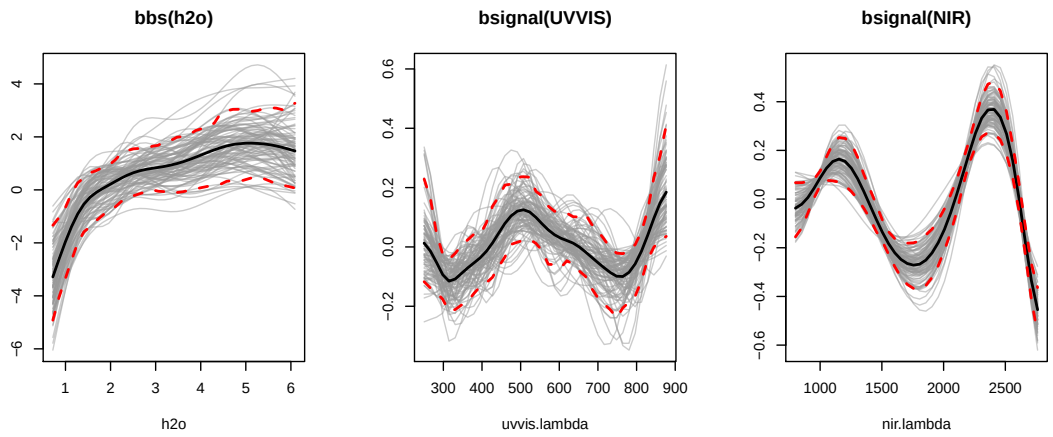

Figure 5: Bootstrapped coefficient estimates of the model for the heat value of the fossil fuels. The coefficient estimates in the bootstrap samples for the smooth effect of the water content (left), the linear effect of the UVVIS spectrum (middle) and the NIR spectrum (right) are displayed. The pointwise 5% and the 95% quantiles are marked with dashed red lines. The pointwise 50% quantile is marked by a black line.

<number of evaluation points> matrices. That is, a functional response $y_i(t_g)$, with $i = 1, \ldots, N$ curves and $g = 1, \ldots, G$ evaluation points, is stored in an $N \times G$ matrix with cases in rows and evaluation points in columns. This corresponds to a data representation in wide format. The t variable must be given as vector $(t_1, \ldots, t_G)^\top$.

For the functional response, curve-specific observation grids are possible, i.e., the ith response curve is observed at evaluation points $(t_{ig}, \ldots, t_{iG_i})^\top$ specific for each curve i. In this case, three pieces of information must be supplied: the values of the response, the evaluation points and the curve to which each of the observations belongs. The response is supplied as the vector $(y_1(t_{11}), \ldots, y_N(t_{NG_N}))^\top$. This vector has length $n = \sum_{i=1}^{N} G_i$. The t variable contains all evaluation points $(t_{11}, \ldots, t_{NG_N})^\top$. The argument `id` contains the information on which observation corresponds to which response curve. The argument `id` must be supplied as a right-sided formula `id = ~ idvariable`.

Case study (ctd.): Emotion components data

In the following, we give an example for a model fit with a functional response. In the first model fit, the response is stored in the matrix `EMG`, in the second in the vector `EMG_long`. We fit an intercept model by defining the `formula` as `y ~ 1` and the `timeformula` as `~ bbs(t)`.

```
R> # fit intercept model with response matrix
R> fos_intercept <- FDboost(EMG ~ 1,
+   timeformula = ~ bbs(t, df = 3),
+   data = emotionHGL)
```

The corresponding mathematical formula is

$$\mathbb{E}(Y_{\text{EMG}}(t)) = \beta_0(t),$$

i.e., we simply estimate the mean curve $\beta_0(t)$ of the functional EMG signal.

To fit a model with response in long format, we first have to convert the data into the corresponding format. We therefore construct a dataset `data_emotion_long` that contains the response in long format. Usually, the long format specification is only necessary for responses that are observed on curve specific grids. We here provide this version for illustrative purposes, but in this example the following model specification is equivalent to the previous model fit `fos_intercept`.

```
R> emotion_long <- emotionHGL
R> emotion_long$EMG_long <- as.vector(emotion_long$EMG)
R> emotion_long$time_long <- rep(emotionHGL$t, each = nrow(emotionHGL$EMG))
R> emotion_long$curveid <- rep(1:nrow(emotionHGL$EMG), ncol(emotionHGL$EMG))


R> fos_intercept_long <- FDboost(EMG_long ~ 1,
+   timeformula = ~ bbs(time_long, df = 3),
+   id = ~ curveid, data = emotion_long)
```

♦

Effects in the formula that are combined with the timeformula

Many covariate effects can be represented by the Kronecker product of two marginal bases as in Equation 6. The response and the bases in covariate direction $\boldsymbol{b}_j(x)$ are specified in `formula` as `Y ~ b_1 + ...+ b_J`. The base-learner for the expansion along t is specified in `timeformula` as `~ b_Y`. Each base-learner in `formula` is combined with the base-leaner in `timeformula` using the operator `%O%`. This operator implements the Kronecker product of two basis vectors as in Equation 6. Consider, for example, `formula = Y ~ b_1 + b_2`. If, `b_1` is defined by `bols(z)` with covariate z and a scalar response is given, using `timeformula = NULL` specifies a model with linear effect $z\beta$. In the case of a functional response, we usually want the effect $z\beta$ to vary for each time-point $t \in \mathcal{T}$ of the response, i.e., $z\beta(t)$. This can be done by defining `timeformula = ~ b_Y`, where the base-learner `b_Y` defines the form of variation in t-direction. Assuming a linear effect in t, `b_Y` is set to `bols(t)`. The combination of `timeformula` and `formula` yields `Y ~ b_1 %O% b_Y + b_2 %O% b_Y`. For the particular example, `b_1 %O% b_Y` is equal to `bols(z) %O% bols(t)` yielding $z\beta(t)$.

If marginal base-learners are specified with a penalty, the Kronecker product of the two basis vectors is defined with an isotropic penalty matrix as in 8. If the effect should only be penalized in t direction, the operator `%A0%` can be used as it sets up the penalty as Equation 9. If `formula` contains base-learners that are composed of two base-learners by `%O%` or `%A0%`, those effects are not expanded with `timeformula`, allowing for model specifications with different effects in t direction. This can be used, for example, to model some effects linearly and others non-linearly in t or to construct effects using `%A0%`. For further details on these operators and their use, we refer to Appendix C.

We start with base-learners for the `timeformula`. Theoretically, it is possible to use any base-learner which models the effect of a continuous variable. Usually, the effects are assumed to be smooth along t. In this case, the base-learner `bbs()` can be used, which represents the smooth effect by P-splines (Schmid and Hothorn 2008a). Thus, `bbs()` uses a B-spline representation for the design matrix and a squared difference matrix as penalty matrix. Using the `bbs()` base-leaner in the `timeformula` corresponds to using a marginal basis $\boldsymbol{b}_Y$ as described in Equation 10.

Base-learners that can be used in `formula` are listed in Table 4. In this table, a selection of additive predictors that can be represented within the array framework are listed in the left column. In the right column, the corresponding `formula` is given. The `timeformula` is set to `~ bbs(t)` to model all effects as smooth effects in t. Thus, the specified effects in `formula` are combined with `timeformula` using the Kronecker product.

additive predictor $h(\boldsymbol{x}, t) = \sum_j h_j(\boldsymbol{x}, t)$	call
$\beta_0(t)$	`y ~ 1`
$\beta_0(t) + z_1\beta_1(t)$	`y ~ 1 + bolsc(z1)`
$\beta_0(t) + f_1(z_1, t)$	`y ~ 1 + bbsc(z1)`
$\beta_0(t) + z_1\beta_1(t) + z_2\beta_2(t) + z_1z_2\beta_3(t)$	`y ~ 1 + bolsc(z1) + bolsc(z2) + bols(z1) %Xc% bols(z2)`
$\beta_0(t) + z_1\beta_1(t) + f_2(z_2, t) + z_1f_3(z_2, t)$	`y ~ 1 + bolsc(z1) + bbsc(z2) + bols(z1) %Xc% bbs(z2)`
$\beta_0(t) + f_1(z_1, t) + f_2(z_2, t) + f_3(z_1, z_2, t)$	`y ~ 1 + bbsc(z1) + bbsc(z2) + bbs(z1) %Xc% bbs(z2)`
$\beta_0(t) + \int_{\mathcal{S}} x(s)\beta_1(s, t)\, ds$	`y ~ 1 + bsignal(x, s = s)`
	`y ~ 1 + bfpc(x, s = s)`
$\beta_0(t) + z\beta_1(t) + \int_{\mathcal{S}} x(s)\beta_2(s, t)\, ds$ $+z \int_{\mathcal{S}} x(s)\beta_3(s, t)\, ds$	`y ~ 1 + bolsc(z) + bsignal(x, s = s) + bsignal(x, s = s) %X% bolsc(z)`

Table 4: Additive predictors that can be represented within the array framework.

For `offset = NULL`, the model contains a smooth offset $\beta_0^*(t)$. The smooth offset is computed prior to the model fit as smoothed population minimizer of the loss. For mean regression, the smooth offset is the smoothed mean over t. The specification `offset = "scalar"` yields a constant offset β_0^*. The resulting intercept in the final model is the sum of the offset and the smooth intercept $\tilde{\beta}_0(t)$ specified in the `formula` as `1`, i.e., $\beta_0(t) = \beta_0^*(t) + \tilde{\beta}_0(t)$.

The upper part of Table 4 gives examples for linear predictors with scalar covariates. A linear effect of a scalar covariate is specified using the base-learner `bolsc()`. This base-learner works for continuous and for factor variables. A smooth effect of a continuous covariate is obtained by using the base-learner `bbsc()`. The base-learners `bolsc()` and `bbsc()` are similar to the base-learners `bols()` and `bbs()` from the **mboost** package, but enforce pointwise sum-to-zero constraints to ensure identifiability for models with functional response (the suffix 'c' refers to 'constrained'). Since, for example, the effect $f_1(z_1, t)$ contains a smooth intercept as special case, the model would not be identifiable without constraints, see Appendix A for more details. We use the constraint $\sum_{i=1}^{N} h_j(\boldsymbol{x}_i, t) = 0$ for all t, which centers each effect for each point t (Scheipl *et al.* 2015). This implies that effects varying over t can be interpreted as deviations from the smooth intercept and that the intercept can be interpreted as global mean if all effects are centered in this way. It is possible to check whether all covariate effects sum to zero for all points t by setting `check0 = TRUE` in the `FDboost()` call. To specify interaction effects of two scalar covariates, the base-learners for each of the covariates are combined using the operator `%Xc%` that applies the sum-to-zero constraint to the interaction effect.

The lower part of Table 4 gives examples for linear predictors with functional covariates. In analogy to models with scalar response, the linear effect $\int_{\mathcal{S}} x(s)\beta(s,t)\,ds$ can be fitted by `bsignal()` or `bfpc()` and the interaction effect is formed using the operator `%X%` (see the explanations for Table 3).

Case study (ctd.): Emotion components data

For the emotion components data with the EMG signal as functional response, $Y_{\text{EMG}}(t)$, $t \in [0, 1560]ms$, we fit models with scalar and functional covariate effects in the following.

Function-on-scalar regression

We specify a model for the conditional expectation of the EMG signal using a random intercept curve for each subject and a linear effect for the study setting `power`:

$$\mathbb{E}(Y_{\text{EMG}}(t)|\boldsymbol{x}) = \beta_0(t) + \sum_{k=1}^{23} I(x_{\text{subject}} = k)\beta_{\text{subject},k}(t) + x_{\text{power}}\beta_{\text{power}}(t), \tag{13}$$

with `subject` having values 1 to 23 for the participants of the study, and x_{power} taking values $\{-1, 1\}$ for low and high power. Both covariate effects in the model are specified by using a centered base-learner. The linear effect of the factor variable `subject` and the effect of `power` are both specified using the `bolsc()` base-learner. Therefore, the effects sum up to zero for each time-point t over all observations $i = 1, \ldots, N = 184$, i.e., $\sum_{i=1}^{N}\sum_{k=1}^{23} I(x_{\text{subject},i} = k)\beta_{\text{subject},k}(t) = 0$ for all t.

```
R> fos_random_power <- FDboost(EMG ~ 1 + bolsc(subject, df = 2)
+   + bolsc(power, df = 1) %A0% bbs(t, df = 6),
+   timeformula = ~ bbs(t, df = 3),
+   data = emotion)
```

As described in Section 3, it is important that all base-learners have the same number of degrees of freedom. In this model the degrees of freedom for each base-learner are $2*3 = 6$. By specifying the `bolsc`-baselearner with `df = 2` for `subject`, the subject effect is estimated with a Ridge penalty similar to a random effect, whereas the `power` effect is estimated unpenalized due to the use of the `%A0%`-operator.

Analogously, a model with response in long format as in `fos_intercept_long` could be specified by changing the formula to the formula of `fos_random_power`.

Function-on-function regression

For the data subset for one specific game condition, we use the effect of the EEG signal to model the EMG signal:

$$\mathbb{E}(Y_{\text{EMG}}(t)|\boldsymbol{x}) = \beta_0(t) + \int_{\mathcal{S}} x_{\text{EEG}}(s)\beta_{\text{EEG}}(s,t)\,ds. \tag{14}$$

In this model each time-point of the covariate $x_{\text{EEG}}(s)$ potentially influences each time-point of the response $Y_{\text{EMG}}(t)$. We center the EEG signal per time point such that $\sum_{i=1}^{N} x_{\text{EEG},i}(s) = 0$ for each s to center its effect per time-point.

```
R> emotionHGL$EEG <- scale(emotionHGL$EEG, scale = FALSE)
```

```
R> fof_signal <- FDboost(EMG ~ 1 + bsignal(EEG, s = s, df = 2),
+   timeformula = ~ bbs(t, df = 3),
+   data = emotionHGL)
```

We will show and interpret plots of the estimated coefficients later on. Assuming that the brain activity (measured via the EEG) triggers the muscle activity (measured via the EMG), it is reasonable to assume that EMG signals are only influenced by past EEG signals. Such a relationship can be represented using a historical effect $\int_{T_1}^{t} x(s)\beta(s,t)\,ds$, which will be discussed in the following paragraph. ♦

Effects in the formula comprising both the effect in covariate and t-direction

If the covariate varies with t, the effect cannot be separated into a marginal basis depending on the covariate and a marginal basis depending only on t. In this case the effects are represented as in Equation 5. Examples for such effects are historical and concurrent functional effects, as discussed in Brockhaus *et al.* (2017). In Table 5 we give an overview of possible additive predictors containing such effects.

additive predictor $h(x,t) = \sum_j h_j(x,t)$	call
$\beta_0(t) + x(t)\beta(t)$	`y ~ 1 + bconcurrent(x, s = s, time = t)`
$\beta_0(t) + \int_{T_1}^{t} x(s)\beta(s,t)\,ds$	`y ~ 1 + bhist(x, s = s, time = t)`
$\beta_0(t) + \int_{t-\delta}^{t} x(s)\beta(s,t)\,ds$	`y ~ 1 + bhist(x, s = s, time = t, limits = limitsLag)`*
$\beta_0(t) + \int_{T_1}^{t-\delta} x(s)\beta(s,t)\,ds$	`y ~ 1 + bhist(x, s = s, time = t, limits = limitsLead)`*
$\int_{l(t)}^{u(t)} x(s)\beta(s,t)\,ds$	`y ~ 1 + bhist(x, s = s, time = t, limits = mylimits)`
$\beta_0(t) + z\beta_1(t) + \int_{T_1}^{t} x(s)\beta_2(s,t)\,ds + z\int_{T_1}^{t} x(s)\beta_3(s,t)\,ds$	`y ~ 1 + bolsc(z) + bhist(x, s = s, time = t) + bhistx(x) %X% bolsc(z)`

Table 5: Additive predictors that contain effects that cannot be separated into an effect in covariate direction and an effect in t direction. These effects in `formula` are not expanded by the `timeformula`. We give examples for general limit functions `mylimits` in this section. In `bhistx()`, the variable `x` has to be of class `hmatrix`, please see the manual of `bhistx()` for details.

The concurrent effect $\beta(t)x(t)$ is only meaningful if the functional response and the functional covariate are observed over the same domain. Models with concurrent effects can be seen as varying-coefficient models (Hastie and Tibshirani 1993), where the effect varies over t. The base-learner `bconcurrent()` expands the smooth concurrent effect $\beta(t)$ in P-splines. The historical effect $\int_{T_1}^{t} x(s)\beta(s,t)\,ds$ uses only covariate information up to the current observation point of the response. The base-learner `bhist()` expands the coefficient surface $\beta(s,t)$ in s and in t direction

using P-splines to fit the historical effect. In Appendix B, details on the representation of functional effects are given.

The interface of `bhist()` is:

```
R> bhist(x, s, time, limits = "s<=t", knots = 10, degree = 3, differences = 1,
+   df = 4, lambda = NULL, check.ident = FALSE)
```

Most arguments of `bhist()` are analogous to those of `bsignal()`. `bhist()` has the additional argument `time` to specify the observation points of the response. Via the argument `limits` in `bhist()` the user can specify integration limits depending on t. Per default a historical effect with limits $s \leq t$ is used. Other integration limits can be specified by using a function with arguments `s` and `t`, which returns `TRUE` for combinations of `s` and `t` that lie within the integration interval and `FALSE` otherwise. In the following, we give three examples for functions that can be used for `limits` resulting in a classical historical effect, a lag effect or a lead effect, respectively:

```
R> limitsHist <- function(s, t) {
+   s <= t
+ }
R> limitsLag <- function(s, t, delta = 5) {
+   s >= t - delta &  s <= t
+ }
R> limitsLead <- function(s, t, delta = 5) {
+   s <= t - delta
+ }
```

The base-learner `bhistx()` is especially suited to form interaction effects such as factor-specific historical effects (Rügamer *et al.* 2018), as `bhist()` cannot be used in combination with the row-wise tensor product operator `%X%` to form interaction effects. `bhistx()` requires the data to be supplied as an object of type `hmatrix`; see the manual of `bhistx()` for its setup.

Case study (ctd.): Emotion components data

Again, we use the subset of the data for one specific game condition. We start with a simple function-on-function regression model by specifying a concurrent effect of the EEG signal on the EMG signal:

$$\mathbb{E}(Y_{\text{EMG}}(t)|\boldsymbol{x}) = \beta_0(t) + x_{\text{EEG}}(t)\beta(t).$$

A concurrent effect is obtained by the base-learner `bconcurrent()`, which is not expanded by the base-learner in `timeformula`. In this model, `timeformula` is only used to expand the smooth intercept.

```
R> fof_concurrent <- FDboost(EMG ~ 1 + bconcurrent(EEG, s = s, time = t, df = 6),
+   timeformula = ~ bbs(t, df = 3), data = emotionHGL,
+   control = boost_control(mstop = 300))
```

Assuming that the activity in the muscle can be completely traced back to previous activity in the brain, a more appropriate model seems to be a historical model including a historical effect

$$\mathbb{E}(Y_{\text{EMG}}(t)|\boldsymbol{x}) = \beta_0(t) + \int_{l(t)}^{u(t)} x_{\text{EEG}}(s)\beta_{\text{EEG}}(s,t)\,ds. \tag{15}$$

From a neuro-anatomy perspective, the signal from the brain requires time to reach the muscle. We therefore set $l(t) = 0$ and $u(t) = t - 3$, which is in line with Rügamer *et al.* (2018).

```
R> fof_historical <- FDboost(EMG ~ 1 + bhist(EEG, s = s, time = t,
+   limits = function(s, t) s <= t - 3, df = 6),
+   timeformula = ~ bbs(t, df = 3), data = emotionHGL,
+   control = boost_control(mstop = 300))
```

More complex historical models are discussed in Rügamer *et al.* (2018). In particular, a model containing random effects for the participants, effects for the game conditions and game condition- as well as subject-specific historical effects of the EEG signal. ♦

It is also possible to combine effects listed in Table 4 and Table 5 to form more complex models. In particular, base-learners with and without array structure can be combined within one model. As in the component-wise boosting procedure each base-learner is evaluated separately, the array structure of the Kronecker product base-learners can still be exploited in such hybrid models.

Model tuning and early stopping

For a fair selection of base-learner, additional care is needed for functional responses as only some of the base-learners in the `formula` are expanded by the base-learner in `timeformula`. In particular, all base-learners listed in Table 4 are expanded by `timeformula`, whereas base-learners given in Table 5 are not expanded by the `timeformula`. For the row-wise tensor product and the Kronecker product of two base-learners, the degrees of freedom for the combined base-learner is computed as product of the two marginally specified degrees of freedom. For instance, `formula = y ~ bbsc(z, df = 3) + bhist(x, s = s, df = 12)` and `timeformula = ~ bbs(t, df = 4)` implies $3 \cdot 4 = 12$ degrees of freedom for the first combined base-learner and 12 degrees of freedom for the second base-learner. The call `extract(object, "df")` displays the degrees of freedom for each base-learner in an `FDboost` object. For other tuning options such as the number of iterations and the specification of the step-length see Section 4.1.

To find the optimal number of boosting iterations for a model fit with functional response, **FDboost** provides two resampling functions. Depending on the specified model, some parameters are computed from the data prior to the model fit: per default a smooth functional offset $\beta_0^*(t)$ is computed (`offset = NULL` in `FDboost()`) and for linear and smooth effects of scalar variables, defined by `bolsc()` and `bbsc()`, transformation matrices for the sum-to-zero constraints are computed. The function `cvrisk.FDboost()` uses the smooth functional offset and the transformation matrices from the original model fit in all folds. Thus, these parameters are treated as fixed and the uncertainty induced by their estimation is not considered in the resampling. On the other hand, `applyFolds()` recomputes the whole model in each fold. The two resampling methods are equal if no smooth offset is used and if the model does not contain any base-learner with a sum-to-zero constraint (i.e., neither `bolsc()` nor `bbsc()`). In general, we recommend to use the function `applyFolds()` to determine the optimal number of boosting iterations for a model with functional response. The interface of `applyFolds()` is:

```
R> applyFolds(object,
+   folds = cv(rep(1, length(unique(object$id))), type = "bootstrap"),
+   grid = 1:mstop(object))
```

The interface is in analogy to the interface of `cvrisk()`. In the argument `object`, the fitted model object is specified. `grid` defines the grid on which the optimal stopping iteration is searched. Via the argument `folds` the resampling folds are defined by suitable weights. The function `applyFolds()` expects resampling weights that are defined on the level of curves, $i = 1, \ldots, N$. That means that the folds must contain weights w_i, $i = 1, \ldots, N$, which can be done easily using the function `cv()`.

Methods to extract and display results

Methods to extract and visualize results are the same irrespective of scalar or functional response. Thus, we refer to the corresponding paragraphs at the end of Section 4.1.

Case study (ctd.): Emotion components data

Exemplarily, the penalty matrix for the historical effect can be extracted as follows:

```
R> kron_pen <- extract(fof_historical, "penalty")
R> as.matrix(kron_pen[[1]][1:5,1:5])
```

This is equal to the kronecker sum of two marginal B-Spline penalties with isotropic penalization (as defined by Equation 7 with $\lambda_j = \lambda_Y$):

```
R> margPen <- extract(with(emotionHGL,
+   bbs(s, knots=10, differences = 1)), "penalty")
R> (kronecker(margPen, diag(ncol(margPen))) +
+   kronecker(diag(ncol(margPen)), margPen))[1:5,1:5]

     [,1] [,2] [,3] [,4] [,5]
[1,]    2   -1    0    0    0
[2,]   -1    3   -1    0    0
[3,]    0   -1    3   -1    0
[4,]    0    0   -1    3   -1
[5,]    0    0    0   -1    3
```

As for scalar response, the `plot`-function can be used to access the estimated effects in a function-on-function regression. In the following, we compare the three basic types of functional covariate effects, which can be used in conjunction with a functional response. We first determine the optimal number of stopping iterations for all three presented models.

```
R> set.seed(123)
R> folds_bs <- cv(weights = rep(1, fof_signal$ydim[1]),
+  type = "kfold", B = 5)


R> cvm_concurrent <- applyFolds(fof_concurrent, folds = folds_bs, grid = 1:300)
R> ms_conc <- mstop(cvm_concurrent)
R> fof_concurrent <- fof_concurrent[ms_conc]
```

```
R> cvm_signal <- applyFolds(fof_signal, folds = folds_bs, grid = 1:300)
R> ms_signal <- mstop(cvm_signal)
R> fof_signal <- fof_signal[ms_signal]


R> cvm_historical <- applyFolds(fof_historical, folds = folds_bs, grid = 1:300)
R> ms_hist <- mstop(cvm_historical)
R> fof_historical <- fof_historical[ms_hist]
```

Then, we plot the estimated effects into one figure:

```
R> par(mfrow = c(1,3))
R> plot(fof_concurrent, which = 2, main = "Concurrent EEG effect")
R> plot(fof_signal, which = 2, main = "Signal EEG effect",
+   n1 = 80, n2 = 80, zlim = c(-0.02, 0.025),
+   col = terrain.colors(20))
R> plot(fof_historical, which = 2, main = "Historical EEG effect",
+   n1 = 80, n2 = 80, zlim = c(-0.02, 0.025),
+   col = terrain.colors(20))
```

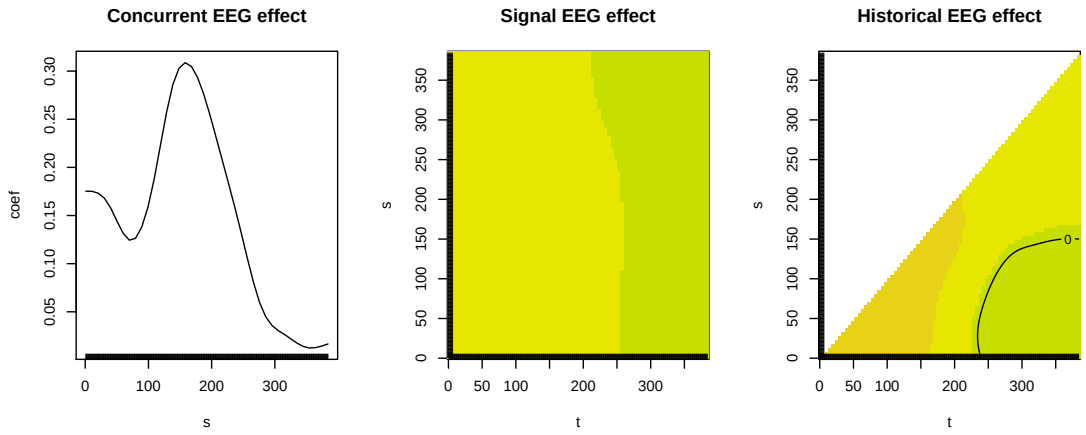

Figure 6: Visualization of estimated concurrent EEG effect (left panel), signal EEG effect (center panel) and historical EEG effect (right panel).

The concurrent effect corresponds to the diagonal of the other two surfaces in Figure 6 and assumes that off-diagonal time-points have no association. Due to the temporal lag between EEG and EMG discussed for model (15), there is no meaningful interpretation for this model and the effect is only shown for demonstrative purposes. The historical effect corresponds to the assumption that the upper triangle in the signal EEG effects should be zero, as future brain activity should not influence the present muscle activity. The results in Figure 6 (right panel) can be interpreted in the same manner as results of a scalar-on-function regression when keeping a certain time point t fixed. For the time point $t = 180$ of the EMG signal, for example, time points $s \approx 100$ to $s \approx 177$ of the EEG

signal do not show an effect, but for $s < 100$ the estimated effect on the expected EMG signal is positive. For a detailed description of the interpretation of historical effect surfaces as shown in Figure 6, we refer to the online appendix of Rügamer *et al.* (2018).

Careful interpretation has to take into account that this data set has a rather small signal-to-noise ratio due to the oscillating nature of both signals. In such cases, it is recommended to check the uncertainty of estimated effects via bootstrap, e.g., by using the `bootstrapCI()` function as exemplarily shown in Figure 7.

```
R> fof_historical_bci <- bootstrapCI(fof_historical, mc.cores = 2,
+   B_inner = 10, type_inner = "kfold")
R> par(mfrow=c(1,3))
R> plot(fof_historical_bci, which = 2, ask = FALSE, pers = FALSE,
+   col = terrain.colors(20), probs = c(0.05, 0.5, 0.95))
```

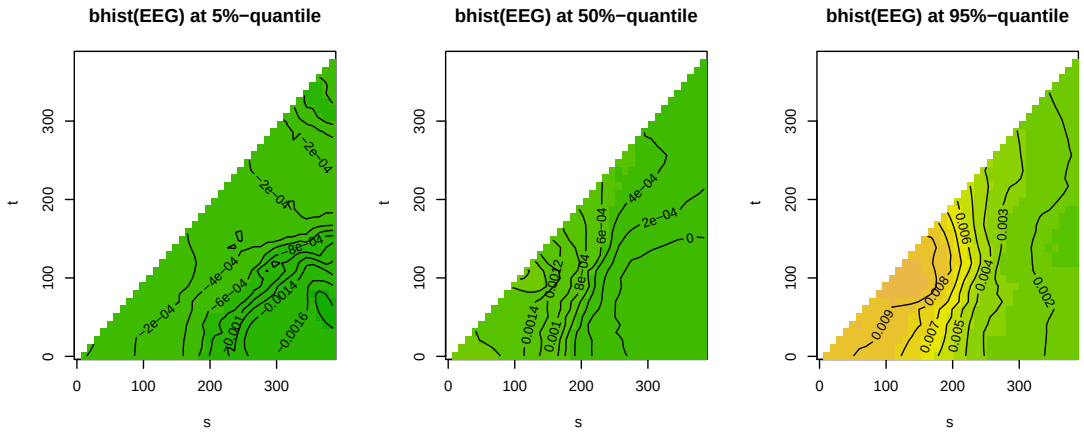

Figure 7: Visualization of three bootstrap quantiles for the historical EEG effect based on 100 bootstrap samples and a 10-fold cross-validation to optimize the stopping iteration for each bootstrap sample.

♦

4.3. Functional regression models beyond the mean

Using boosting for model estimation it is possible to optimize other loss functions than the squared error loss. This allows to fit, e.g., generalized linear models (GLMs) and quantile regression models (Koenker 2005). It is also possible to fit models for several parameters of the conditional response distribution in the framework of generalized additive models for location, scale and shape (GAMLSS, Rigby and Stasinopoulos 2005).

For the estimation of these more general models, a suitable loss function in accordance with the modeled characteristic of the response distribution is defined and optimized. The absolute error loss (L_1 loss), for instance, implies median regression, and minimizing the L_2-loss yields mean regression.

In `FDboost()`, the regression type is specified by the `family` argument. The `family` argument expects an object of class `Family`, which implements the respective loss function with its corre-

sponding negative gradient and link function. The default is `family = Gaussian()` which yields L_2-boosting (Bühlmann and Yu 2003). This means that the mean squared error loss is minimized, which is equivalent to maximizing the log-likelihood of the normal distribution. Table 6 lists some loss functions currently implemented in **mboost**, which can be directly used in **FDboost** (see Hofner *et al.* 2014, for more families). Hofner *et al.* (2014) also give an example on how to implement new families via the function `Family()`. See also the help page `?Family` for more details on all families.

response type	regression type	loss	call
continuous response	mean regression	L_2 loss	`Gaussian()`
	median regression	L_1 loss	`Laplace()`
	quantile regression	check function	`QuantReg()`
	expectile regression	asymmetric L_2	`ExpectReg()`
	robust regression	Huber loss	`Huber()`
non-negative response	gamma regression	$-l_{\text{gamma}}$	`GammaReg()`
binary response	logistic regression	$-l_{\text{Bernoulli}}$	`Binomial()`
	AdaBoost classification	exponential loss	`AdaExp()`
count response	Poisson model	$-l_{\text{Poisson}}$	`Poisson()`
	neg. binomial model	$-l_{\text{neg. binomial}}$	`NBinomial()`
scalar ordinal response	proportional odds model	$-l_{\text{proportional odds model}}$	`ProppOdds()`
scalar categorical response	multinomial model	$-l_{\text{multinomial}}$	`Multinomial()`
scalar survival time	Cox model	$-l_{\text{cox}}$	`CoxPH()`

Table 6: Overview of some families that are implemented in **mboost**. $-l_F$ denotes the negative log-likelihood of the distribution or model F.

For a continuous response, several model types are available (Bühlmann and Hothorn 2007): L_2-boosting yields mean regression; a more robust alternative is median regression, which optimizes the absolute error loss; the Huber loss is a combination of L_1 and L_2 loss (Huber 1964); quantile regression can be used to model a certain quantile of the conditional response distribution (Fenske *et al.* 2011); and expectile regression for modeling an expectile (Newey and Powell 1987; Sobotka and Kneib 2012). For a non-negative continuous response, models assuming the gamma distribution can be useful. A binary response can be modeled in a GLM framework with a logit model or by minimizing the exponential loss, which corresponds to the first boosting algorithm 'AdaBoost' (Friedman 2001; Bühlmann and Hothorn 2007). Count data can be modeled assuming a Poisson or negative binomial distribution (Schmid, Potapov, Pfahlberg, and Hothorn 2010).

For functional response, we compute the loss point-wise and integrate over the domain of the response.

The following models can only be applied for scalar and not for functional response. For ordinal response, a proportional odds model can be used (Schmid, Hothorn, Maloney, Weller, and Potapov 2011). For categorical response, the multinomial logit model is available. For survival models, boosting Cox proportional hazard models and accelerated failure time models have been introduced by Schmid and Hothorn (2008b).

Case study (ctd.): Emotion components data

So far, we fitted a model for the conditional mean of the response. As a more robust alternative, we consider median regression by setting `family = QuantReg(tau = 0.5)`. We use the `update` function, to update the functional model with the new family.

```
R> fof_signal_med <- update(fof_signal, family = QuantReg(tau = 0.5))
```

For median regression, the smooth intercept is the estimated median at each time-point and the effects are deviations from the median.

Similarily, if a certain quantile of the functional response is of interest, for example the 90% quantile, the model can be updated as follows

```
R> fof_historical_q90 <- update(fof_historical, family = QuantReg(tau = 0.9))
```

which is equivalent to the following initial model specification:

```
R> fof_historical_q90 <- FDboost(EMG ~ 1 + bhist(EEG, s = s, time = t,
+   limits = function(s, t) s <= t - 3, df = 6),
+   timeformula = ~ bbs(t, df = 3), data = emotionHGL,
+   control = boost_control(mstop = 300),
+   family = QuantReg(tau = 0.9))
```

To illustrate an example for scalar-on-function regression with binary response, consider the case, in which the goal is to predict the `game_outcome` in the case study for the emotions component data using only the muscle activity measured via the EMG. Consider the model

$$g(\mathbb{P}(Y_{i,j}|\boldsymbol{x}_{i,j})) = \beta_0 + \gamma_j + \int_{\mathcal{S}} x_{\text{EMG},i,j}(s)\beta_{\text{EMG}}(s)ds + \int_{\mathcal{S}} x_{\text{EMG},i,j}(s)\gamma_{\text{EMG},j}(s)ds,$$

for observation $i = 1, \ldots, 8$ of subject $j = 1, \ldots, 23$, where g is the inverse of the logit function, $Y_{i,j} \in \{0, 1\}$ determines the game outcome (*gain* and *loss*, respectively) for participant j in game i, γ_j is a subject effect and the EMG is modeled using a global EMG effect β_{EMG} as well as a subject-specific EMG effect $\gamma_{\text{EMG},j}$. We first center the EMG-signal as it is now used as covariate

```
R> emotion$EMG <- scale(emotion$EMG, center = TRUE, scale = FALSE)
```

and specifiy the model in `FDboost` as follows

```
R> sof_binary <- FDboost(
+   game_outcome ~ 1 +
+   brandom(subject, df = 4) +
+   bsignal(EMG, s = s, df = 4) +
+   brandom(subject, df = 2) %X% bsignal(EMG, s = s, df = 2),
+   data = emotion,
+   family = Binomial(),
+   control = boost_control(mstop = 5000),
+   timeformula = NULL)
```

Note that the row-wise tensor product operator `%X%` in this case is used to specify a subject specific functional effect of the EMG-signal and the resulting degrees of freedom of this base learner are determined as the product of the `df`s of both base learners. To get a measure of the performance of this model, we could, e.g., compute predictions and look at the confusion matrix when simply rounding the predictions:

```
R> predictions <- predict(sof_binary, type = "response")
R> round_preds <- round(predictions)
R> table(round_preds, as.numeric(emotion$game_outcome))
```

```
   0  1
0 77 12
1 15 80
```

♦

The combination of GAMLSS with functional variables is discussed in Brockhaus *et al.* (2018) and Stöcker *et al.* (2017). For GAMLSS models, **FDboost** builds on the package **gamboostLSS** (Hofner, Mayr, Fenske, Thomas, and Schmid 2017), in which families are implemented to fit GAMLSS. For details on the boosting algorithm to fit GAMLSS, see Mayr *et al.* (2012) and Thomas, Mayr, Bischl, Schmid, Smith, and Hofner (2018). The families in **gamboostLSS** need to model at least two distribution parameters. For an overview of currently implemented response distributions for GAMLSS, we refer to Hofner, Mayr, and Schmid (2016). In **FDboost**, the function `FDboostLSS()` implements GAMLSS with functional data. The interface of `FDboostLSS()` is:

```
R> FDboostLSS(formula, timeformula, data = list(), families = GaussianLSS(), ...)
```

In `formula` a named list of formulas is supplied. Each list entry in the `formula` specifies the potential covariate effects for one of the distribution parameters. The names of the list are the names of the distribution parameters. The argument `families` is used to specify the assumed response distribution with its modeled distribution parameters. The default `families = GaussianLSS()` yields a Gaussian location scale model. In the dots-argument further arguments passed to `FDboost()` can be supplied. The model object which is fitted by `FDboostLSS()` is a list of `FDboost` model objects. It is not possible to automatically fit a smooth offset within `FDboostLSS()`. Per default, a scalar offset value is used for each distribution parameter. For functional response, it can thus be useful to center the response prior to the model fit. All integration weights for the loss function are set to one, corresponding to the negative log-likelihood of the observation points.

For model objects fitted by `FDboostLSS()`, methods to estimate the optimal stopping iterations, as well as methods for plotting and prediction exist. For more details on boosting GAMLSS models, we refer to Hofner *et al.* (2016), which is a tutorial for the package **gamboostLSS**.

Case study (ctd.): Fossil fuel data

We fit a Gaussian location scale model for the heat value. Such a model is obtained by setting `families = GaussianLSS()`, where the expectation is modeled using the identity link and the

standard deviation by a log-link. Mean and standard deviation of the heat value are modeled by different covariates:

$$
\begin{aligned}
Y_i|\boldsymbol{x}_i &\sim N(\mu_i, \sigma_i^2), \\
\mu_i &= \beta_0 + f(z_{\text{H2O},i}) + \int_{\mathcal{S}_{\text{NIR}}} x_{\text{NIR},i}(s_{\text{NIR}})\beta_{\text{NIR}}(s_{\text{NIR}})\, ds_{\text{NIR}} + \int_{\mathcal{S}_{\text{UV}}} x_{\text{UV},i}(s_{\text{UV}})\beta_{\text{UV}}(s_{\text{UV}})\, ds_{\text{UV}} \\
\log \sigma_i &= \alpha_0 + \alpha_1 z_{\text{H2O},i}.
\end{aligned}
$$

The mean is modeled depending on the water content as well as depending on the NIR and the UVVIS spectrum. The standard deviation is modeled using a log-link and a linear predictor based on the water content. The `formula` has to be specified as a list of two formulas with names `mu` and `sigma` for mean and standard deviation of the normal distribution. We use the noncyclic fitting method that is introduced by Thomas *et al.* (2018).

```
R> fuelSubset$h2o_center <- fuelSubset$h2o - mean(fuelSubset$h2o)
R> library("gamboostLSS")
R> sof_ls <- FDboostLSS(list(mu = heatan ~ bbs(h2o, df = 4)
+   + bsignal(UVVIS, uvvis.lambda, knots = 40, df = 4)
+   + bsignal(NIR, nir.lambda, knots = 40, df = 4),
+   sigma = heatan ~ 1 + bols(h2o_center, df = 2)),
+   timeformula = NULL, data = fuelSubset,
+   families = GaussianLSS(), method = "noncyclic")
R> names(sof_ls)

[1] "mu"    "sigma"
```

The optimal number of boosting iterations is searched on a grid of 1 to 2000 boosting iterations. The algorithm updates in each boosting iteration the base-learner that best fits the negative gradient. Thus, in each iteration the additive predictor for only one of the distribution parameters is updated.

```
R> set.seed(123)
R> cvm_sof_ls <- cvrisk(sof_ls, folds = cv(model.weights(sof_ls[[1]]), B = 5),
+   grid = 1:2000, trace = FALSE)
```

The estimated coefficients for the expectation are similar to the effects resulting from the pure mean model. The water content has a negative effect on the standard deviation, with higher water content being associated with lower variability.

4.4. Variable selection by stability selection

Variable selection can be refined using stability selection (Meinshausen and Bühlmann 2010; Shah and Samworth 2013). Stability selection is a procedure to select influential variables while controlling false discovery rates and maximal model complexity. For component-wise gradient boosting, it is implemented in **mboost** in the function `stabsel()` (Hofner, Boccuto, and Göker 2015), which can also be used for model objects fitted by `FDboost()`. Brockhaus *et al.* (2017) compute

function-on-function regression models with more functional covariates than observations and perform variable selection by stability selection. Thomas *et al.* (2018) discuss stability selection for GAMLSS estimated by boosting.

4.5. Computational Characteristics and Costs

In order to give rough estimates on how `FDboost` scales up with increasing number of obervations N, observation points per response curve G, number of base-learners J as well as other data and run-time related setups, this section provides some further insights into the algorithm and bottlenecks to bear in mind.

Estimating the run-time of `FDboost` is not straightforward as it depends on the number of boosting iterations, the size of the data set, the number and complexity of base-learners, as well as the type and parallelization of resampling. Different loss-functions, i.e., different types of regression should not change the run-time directly, but may require a smaller step-length as explained before which in turn induces a higher number of boosting iterations. In the following simulation study, we use the default value $\nu = 0.1$. `FDboost` scales linearly in the number of iterations, which is why we use a fixed number $m_{\text{stop}} = 50$ in the following. However, note that the initialization of the model can get computationally very expensive, if very complex base-learners are defined (see, e.g. Rügamer *et al.* 2018). This is due to a singular-value decomposion of the design matrix of each base-learner, which is needed to compute the smoothing parameter corresponding to the pre-defined degrees of freedom and which has cubic run-time in the number of columns of the design matrix. For smooth effects, the number of columns of the design matrix of a base-learner is defined by the number of knots. For the simulation study, we use 20 knots for a historical or unrestricted functional effect base-learner for function-on-function and scalar-on-function models, respectively. This corresponds to the number of knots used in the `fuelSubset` data and yields rather flexible estimates of functions. For applications where less flexibility is needed, this simulation study can be seen as a worst-case scenario estimate of run-times.

Furthermore, we define the number of observations to be $N \in \{10, 100, 1000\}$, the number of time-points to be $G \in \{1, 10, 100, 1000\}$ and the number of base-learners to be $J \in \{5, 10, 20\}$. For $G = 1$ scalar-on-function regression is performed, the other settings correspond to function-on-function regression. Due to computational burden, we exclude settings, in which $N = 1000$ and $G = 1000$ at the same time. The simulation was conducted on a Linux server with *Intel(R) Xeon(R) CPU E5-4620 0* with *2.20GHz*, *64 cores* and *512 GB RAM*.

We do not consider resampling or validation here as resampling on k folds should approximately yield a k-multiple of the original run-time if not parallelized, i.e. run-times scale linearly in the number of folds. With parallelization the run-time can be reduced to the run-time of a single model fit.

The results of the simulation study are visualized in the following, indicating a roughly linear increase in run-time and total allocation of memory by the number of observations (note that both are plotted against $\log_{10}(N)$), a linear increase by the number of observed time points per curve G as well as by the number of base-learners J. The $m_{\text{stop}} = 50$ iterations play a comparatively minor role in time and memory consumption after the model has been initialized. Note that the total amount of allocated memory can only be interpreted in relative terms and does not correspond to the maximum amount of consumed memory at one time-point, which is considerably smaller.

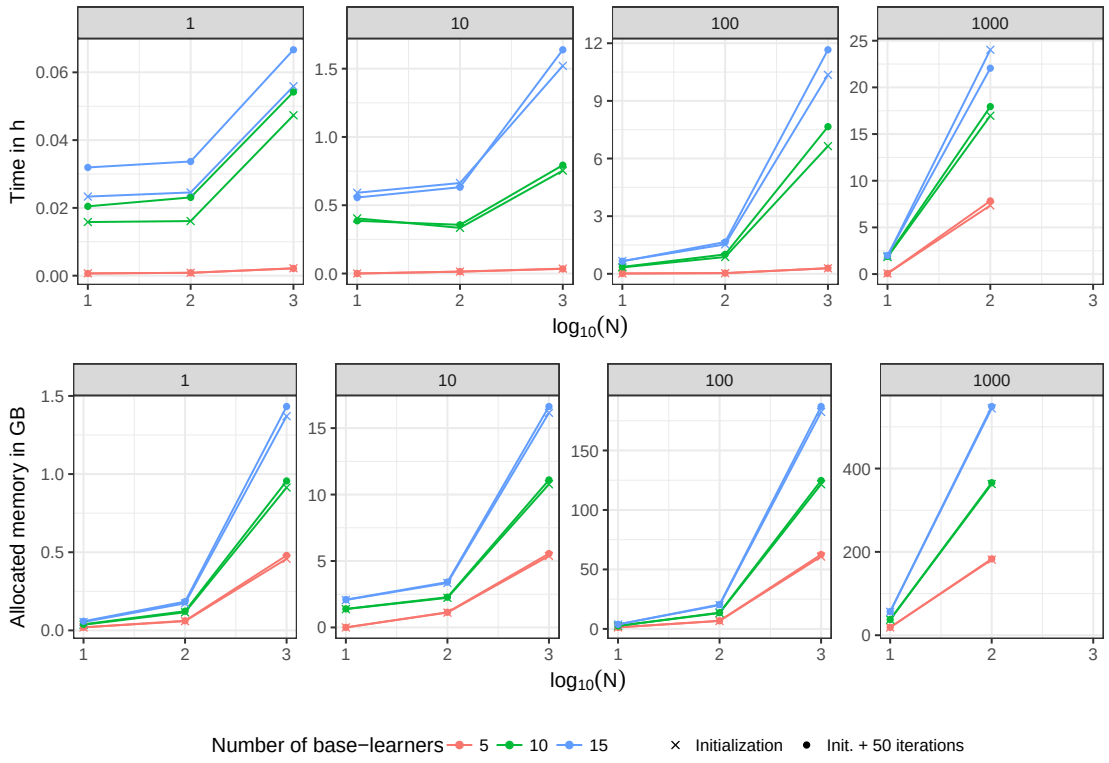

Figure 8: Estimated computational costs of FDboost in the simulation study. Different columns correspond to different numbers of observed time points per curve (G) and the number of base learners (J) is visualized by different colors.

5. Discussion

The R add-on package **FDboost** provides a comprehensive implementation to fit functional regression models by gradient boosting. The implementation allows to fit regression models with scalar or functional response depending on many covariate effects. The framework includes mean, mean with link function, median and quantile regression models as well as GAMLSS. Various covariate effects are implemented including linear and smooth effects of scalar covariates, linear effects of functional covariates and interaction effects, also between scalar and functional covariates (Rügamer *et al.* 2018). The linear functional effects can have flexible integration limits, for example, to form historical or lag effects (Brockhaus *et al.* 2017). Whenever possible, the effects are represented in the structure of linear array models (Currie *et al.* 2006) to increase computational efficiency (Brockhaus *et al.* 2015). Component-wise gradient boosting allows to fit models in high-dimensional data situations and performs data-driven variable selection. **FDboost** builds on the well tested and modular implementation of **mboost** (Hothorn *et al.* 2016). This facilitates the implementation of further base-learners in order to fit new covariate effects and that of families modeling other characteristics of the conditional response distribution.

A. Constraints for effects of scalar covariates

Consider a model for functional response with smooth intercept and an effect that contains a smooth intercept as special case, $\mathbb{E}(Y_i(t)) = \beta_0(t) + h_j(\boldsymbol{x}_i, t)$, and define the mean effect at each point t as $\bar{h}_j(\boldsymbol{x}, t) = \mathbb{E}_X(h_j(\boldsymbol{X}, t))$. This model can be parametrized in different ways, e.g., as

$$\begin{aligned}\mathbb{E}(Y_i(t)) &= \beta_0(t) + h_j(\boldsymbol{x}_i, t) \\ &= \left[\beta_0(t) + \bar{h}_j(\boldsymbol{x}, t)\right] + \left[h_j(\boldsymbol{x}_i, t) - \bar{h}_j(\boldsymbol{x}, t)\right] \\ &= \tilde{\beta}_0(t) + \tilde{h}_j(\boldsymbol{x}, t).\end{aligned}$$

The problem arises as $\bar{h}_j(\boldsymbol{x}, t)$ (or any other smooth function in t) can be shifted between the intercept and the covariate effect. At the level of the design matrices of these effects, this can be explained by the fact that the columns of the design matrix $\boldsymbol{B}_{jY}$ and the columns of the design matrix of the functional intercept are linearly dependent. To obtain identifiable effects, Scheipl *et al.* (2015) propose to center such effects $h_j(\boldsymbol{x}, t)$ at each point t. The centering is achieved by setting the point-wise expectation over the covariate effects to zero on $\mathcal{T}$, i.e., $\mathbb{E}_X(h_j(\boldsymbol{X}, t)) = 0$ for all t, approximated by the sum-to-zero constraint $\sum_{i=1}^{N} h_j(\boldsymbol{x}_i, t) = 0$ for all t. How to enforce such constraints is described in Appendix A of Brockhaus *et al.* (2015). Other constraints to obtain identifiable models are possible. However, this sum-to-zero constraint for each point t yields an intuitive interpretation: the intercept can be interpreted as global mean and the covariate effects can be interpreted as deviations from the smooth intercept.

The constraint is enforced by a basis transformation of the design and penalty matrix. As shown in Brockhaus *et al.* (2015), it is sufficient to apply the constraint on the covariate-part of the design and the penalty matrix. Thus, it is not necessary to transform the basis in t direction.

B. Base-learners for functional covariates

The base-learner `bsignal()` sets up a linear effect of a functional variable $\int_{\mathcal{S}} x_j(s)\beta_j(s)\,ds \approx \boldsymbol{b}_j(\boldsymbol{x})^\top \boldsymbol{\theta}_j$ using P-splines. We approximate the integral numerically as a weighted sum using integration weights $\Delta(s)$ (Wood 2011), see Equation 3:

$$\begin{aligned}\boldsymbol{b}_j(\boldsymbol{x}_i)^\top &= \left[\sum_{r=1}^{R} \Delta(s_r) x_i(s_r)\phi_1(s_r) \;\cdots\; \sum_{r=1}^{R} \Delta(s_r) x_i(s_r)\phi_{K_j}(s_r)\right] \\ &\approx \left[\int_{\mathcal{S}} x_i(s)\phi_1(s)\,ds \;\cdots\; \int_{\mathcal{S}} x_i(s)\phi_{K_j}(s)\,ds\right],\end{aligned}$$

where $\phi_k(s_r)$, $k = 1, \ldots, K_j$ are B-splines evaluated at s_r. The corresponding penalty matrix $\boldsymbol{P}_j$ is a squared difference matrix and thus, the smooth effect $\beta_j(s)$ in s is represented by P-splines.

Using the base-learner `bfpc()` the linear functional effect $\int_{\mathcal{S}} x_j(s)\beta_j(s)\,ds$ is specified using an FPC basis. The functional covariate $x_j(s)$ and the coefficient $\beta_j(s)$ are both represented in the basis that is spanned by the functional principal components (FPCs, see, e.g., Ramsay and Silverman 2005, Chap. 8 and 9) of $x_j(s)$. Let $X_j(s)$ be a zero-mean stochastic process in the space of all square-integrable functions $L^2(\mathcal{S})$. Let $x_{ij}(s)$ be the observations of the copies $X_{ij}(s)$ of this process. We denote the eigenvalues of the auto-covariance of $X_j(s)$ as $\zeta_1 \geq \zeta_2 \geq \cdots \geq 0$ and the corresponding eigenfunctions as $e_k(s)$, $k \in \mathbb{N}$. The eigenfunctions $\{e_k(s), k \in \mathbb{N}\}$ form an orthonormal basis for the $L^2(\mathcal{S})$. Using the Karhunen-Loève theorem, the functional covariate

can be represented as weighted sum

$$X_{ij}(s) = \sum_{k=1}^{\infty} Z_{ik} e_k(s),$$

where Z_{ik} are uncorrelated mean zero random variables with variance ζ_k and realizations z_{ik}. In practice, the infinite sum is truncated at a certain value K_j. Representing the functional covariate and the coefficient function by this truncated basis with weights θ_l and z_{ik}, respectively, the effect simplifies to

$$\int_{\mathcal{S}} x_{ij}(s)\beta_j(s)\, ds \approx \sum_{k,l=1}^{K_j} \int_{\mathcal{S}} z_{ik} e_k(s) e_l(s) \theta_l \, ds = \sum_{k=1}^{K_j} z_{ik}\theta_k,$$

as the eigenfunctions $e_k(s)$ are orthonormal. Thus, this approach is equivalent to using the (estimated) first K_j FPC scores z_{ik} as linear covariates. The number of eigenfunctions is usually chosen such that the truncated basis explains a fixed proportion of the total variability of the covariate, for example 99% (cf., Morris 2015). This truncation achieves regularized effects, as the effect can only lie in the space spanned by the first K_j eigenfunctions. For the penalty matrix $\boldsymbol{P}_j$ the identity matrix is used in `bfpc()`.

For scalar response, the base-learners `bsignal()` and `bfpc()` yield the effect $\int_{\mathcal{S}} x_j(s)\beta_j(s)\, ds$. Combining them with a smooth effect in t using `bbs()`, they can be used to fit effects for function-on-function regression $\int_{\mathcal{S}} x_j(s)\beta_j(s,t)\, ds$.

The base-learner `bhist()` allows to specify functional linear effects with integration limits depending on t, $\int_{l(t)}^{u(t)} x(s)\beta(s,t)\, ds$. Per default, a historical effects with limits $[l(t), u(t)] = [T_1, t]$ is fitted. The integral is approximated by a numerical integration scheme (Scheipl *et al.* 2015). We transform the observations of the functional covariate $x_j(s_r)$ such that they contain the integration limits and the weights for numerical integration. We define $\tilde{x}_j(s_r, t) = I\left(l(t) \leq s_r \leq u(t)\right) \Delta(s_r) x_j(s_r)$, with indicator function $I(\cdot)$ and integration weights $\Delta(s_r)$. The marginal basis over the covariates $\boldsymbol{x}$, which in this case also depends on t, is:

$$\begin{aligned} \boldsymbol{b}_{jY}(\boldsymbol{x}_i, t)^\top &= \left[\sum_{r=1}^{R} \tilde{x}_j(s_r, t)\phi_1(s_r) \;\cdots\; \sum_{r=1}^{R} \tilde{x}_j(s_r, t)\phi_{K_j}(s_r)\right] \otimes \left[\phi_1(t_g) \;\cdots\; \phi_{K_Y}(t_g)\right] \\ &\approx \left[\int_{l(t)}^{u(t)} x_i(s)\phi_1(s)\, ds \;\cdots\; \int_{l(t)}^{u(t)} x_i(s)\phi_{K_j}(s)\, ds\right] \otimes \left[\phi_1(t_g) \;\cdots\; \phi_{K_Y}(t_g)\right]. \end{aligned}$$

The isotropic penalty in Equation 8 is used with squared difference matrices as marginal penalties to form P-splines bases for the s and t direction of $\beta(s,t)$.

For a concurrent effect $x(t)\beta(t)$, the base-learner `bconcurrent()` can be used. The smooth effect $\beta(t)$ in t is expanded by P-splines.

C. Row tensor product and Kronecker product bases

In the R package **mboost** (Hothorn *et al.* 2016), the Kronecker product of two base-learners is implemented as `%O%`. The row-wise tensor product of two base-learners is implemented in the operator `%X%`. The row-wise tensor product of two marginal design matrices, $\boldsymbol{B}_j \in \mathbb{R}^{n \times K_j}$ and

$\boldsymbol{B}_Y \in \mathbb{R}^{n \times K_Y}$, is defined as $n \times K_j K_Y$ matrix

$$\boldsymbol{B}_j \odot \boldsymbol{B}_Y = (\boldsymbol{B}_j \otimes \mathbf{1}_{K_Y}^\top) \cdot (\mathbf{1}_{K_j}^\top \otimes \boldsymbol{B}_Y),$$

where $\cdot$ denotes entry-wise multiplication and $\mathbf{1}_K$ is the K-dimensional vector of ones. The operators `%X%` and `%O%` use the Kronecker product or the row-wise tensor product to compute the design matrix. The penalty is computed according to Equation 7. When `%X%` or `%O%` is called with specified argument `df` in both marginal base-learners, the degrees of freedom of the composed effect are computed as the product of the two specified degrees of freedom. Then, only one smoothing parameter is computed for an isotropic penalty like in Equation 8. Consider, for example, the composed base-learner `bols(z1, df = df1) %O% bbs(t, df = df2)`. The base-learner `bols()` specifies a linear effect. The base-learner `bbs()` specifies a smooth effect represented by P-splines. Thus, the composed base-learner yields the effect $z_1\beta_j(t)$, which is linear in z_1 and smooth in t. The global degrees of freedom for the composed base-learner are computed as $\text{df}_j =$ `df1 * df2`. The corresponding smoothing parameter λ_j is computed by Demmler-Reinsch orthogonalization (Ruppert, Wand, and Carroll 2003, Appendix B.1.1).

For array models, `FDboost()` connects the effects of `formula` and `timeformula` by the operator `%O%`, yielding `b_1 %O% b_Y + ...+ b_J %O% b_Y`. The operator `%O%` uses the array framework of Currie *et al.* (2006) to efficiently implement such effects in boosting (Hothorn, Kneib, and Bühlmann 2013). If it is not possible to use the array framework, e.g., if the response is observed on curve-specific grids or for historical effects, the design matrix is computed as row-wise tensor product basis, i.e., using the operator `%X%`. Within the function `FDboost()` the appropriate operator is used automatically. When the marginal base-learners are supplied with specified degrees of freedom (argument `df`), `%O%` and `%X%` use the isotropic penalty (8).

The anisotropic penalty (7) is obtained if the smoothing parameter is specified in both marginal base-learners; for instance, as `bols(z1, lambda = lambda1) %O% bbs(t, lambda = lambda2)`. However, it is hard to control the degrees of freedom in this case such that each base-learner in the model has the same number of degrees of freedom. Thus, specifying the smoothing parameter λ in both marginal base-learners is hardly applicable in practice.

In some cases, one only wants to penalize the basis in t direction. In this case, the penalty in Equation 9 can be used. Such a penalty is obtained using the operators `%A0%` or `%Xa0%`, for the Kronecker and the row-wise tensor product basis, respectively. When `%A0%` or `%Xa0%` are used to form an effect with penalty (9), the number of degrees of freedom in the first base-learner has to be equal to the number of its columns. Consider, `bols(z1, df = 1, intercept = FALSE) %A0% bbs(t, df = df2)`, with a metric variable `z1`. This specification implies $\boldsymbol{b}_j(\boldsymbol{x}_i) = z_{i1}$ and $\boldsymbol{P}_j = \mathbf{0}$ for the `bols()` base-learner. The `bbs()` base-learner sets up a design matrix of B-spline evaluations in t and a squared difference matrix as penalty matrix.

Linking `formula` and `timeformula` in `FDboost()` to representation (6), the J base-learners in `formula` correspond to the J marginal bases $\boldsymbol{b}_j$ and the base-learners in `timeformula` corresponds to the marginal basis $\boldsymbol{b}_Y$. If it is possible to represent the effects as Kronecker product, the base-learners are combined by `%O%`. Otherwise, the row-wise tensor product `%X%` is used to combine the marginal bases.

Consider, for example, `formula = Y ~ b_1 + b_2 + ...+ b_J`, and the `timeformula = ~ b_Y`. For an array model, this yields `Y ~ b_1 %O% b_Y + b_2 %O% b_Y + ... + b_J %O% b_Y`. If `formula` contains base-learners that are composed of two base-learners by `%O%` or `%A0%`, those effects are not expanded with `timeformula`, allowing for model specifications with different effects in t direction. For example, `formula = Y ~ b_1 + b_2 %A0% b_Y0`, and `timeformula = ~ b_Y`,

with non-linear base-learner `b_Y` and linear base-learner `b_Y0`, yield `Y ~ b_1 %O% b_Y + b_2 %A0% b_Y0`.

D. Example code for resampling with repeated measurements

In the following, we search the optimal stopping iteration for model (13), which contains a linear effect for the game condition power and a person-specific effect.

We search the optimal stopping iteration by a 5-fold cross-validation. The resampling is done on the level of curves, assuming that the observations per subject are independent conditional on the subject specific effects. We use the function `applyFolds()` for the resampling.

```
R> set.seed(123)
R> folds_bs <- cv(weights = rep(1, fos_random_power$ydim[1]),
+   type = "kfold", B = 5)
R> cvm <- applyFolds(fos_random_power, folds = folds_bs, grid = 1:200)
```

The optimal stopping iteration is estimated to be 200, which is the upper limit of the searched grid. Thus, the resampling has to be rerun with a higher maximal number of boosting iterations.

To resample the observations on the level of independent observation units, the folds can be set up on the level of subjects. The corresponding folds for a leave-on-subject out cross-validation, which are then passed to `applyFolds()`, could be constructed as follows:

```
R> set.seed(123)
R> folds_bs_long_subject <- sapply(levels(emotion$subject),
+   function(x) as.numeric(x != emotion$subject))
```

E. Fitting factor-specific historical models

In this section we provide code to fit a more complex and realistic model to the emotion component data. As the EMG signal might depend on all three study settings (`power`, `game_outcome`, `control`) as well as their interactions, and the influence of the EEG signal might also be specific for each setting as well as for each subject, we assume the following model (cf. Rügamer *et al.*

2018):

$$
\begin{aligned}
\mathbb{E}(Y_{\text{EMG},i,j}(t)|\boldsymbol{x}_{i,j}) = \beta_0(t) &+ \gamma_{\text{subject},j}(t) \\
&+ I(x_{\text{power},i,j} = 1)\beta_{\text{power}}(t) \\
&+ I(x_{\text{outcome},i,j} = 1)\beta_{\text{outcome}}(t) \\
&+ I(x_{\text{control},i,j} = 1)\beta_{\text{control}}(t) \\
&+ I(x_{\text{power},i,j} = 1, x_{\text{outcome},i,j} = 1)\beta_{\text{power,outcome}}(t) \\
&+ I(x_{\text{outcome},i,j} = 1, x_{\text{control},i,j} = 1)\beta_{\text{outcome,control}}(t) \\
&+ I(x_{\text{power},i,j} = 1, x_{\text{control},i,j} = 1)\beta_{\text{power,control}}(t) \\
&+ I(x_{\text{power},i,j} = 1, x_{\text{outcome},i,j} = 1, x_{\text{control},i,j} = 1) \cdot \\
&\quad \beta_{\text{power,outcome,control},i}(t) \\
&+ \int_0^{t-3} x_{\text{EMG},i,j}(s)\beta_{\text{EMG}}(s,t)ds \\
&+ \int_0^{t-3} x_{\text{EMG},i,j}(s)\gamma_{\text{EMG},i}(s,t)ds \\
&+ \int_0^{t-3} x_{\text{EMG},i,j}(s)\zeta_{\text{EMG},j}(s,t)ds + \varepsilon_{i,j}(t)
\end{aligned}
\tag{16}
$$

for observation $i = 1, \ldots, 8$ corresponding to the 8 different game conditions of subject $j = 1, \ldots, 23$. The model was proposed in Rügamer *et al.* (2018), which extended historical models by allowing for factor-specific historical effects. To our knowledge, **FDboost** so far is the only software capable of fitting such effects.

To this end, we have to define the 3 two-way interactions `power.outcome`, `outcome.control`, `power.control`, 1 three-way interaction `gamecondition` and an `hmatrix`-object `X1h`. The object is needed for the function `bhistx`, which in turn allows to combine historical effects with factor variables using the row-wise tensor product operator `%X%`. To construct a `hmatrix`-object, the time and an identifier for each curve in long format must be supplied along with the original response. The corresponding model fit in R takes around 75 minutes to fit the model with 5000 iterations and needs approximately a maximum of 15GB RAM at once. We further allow for an anisotropic penalty for all factor effects that are time-dependent, which is achieved by using the `%A%`-operator.

This example also demonstrates how the degrees of freedom can be defined to be equal across all base-learners (in this case $\text{df}_j = 20$), which is explained in Appendix C.

```
R> N <- nrow(emotion$EEG)
R> G <- ncol(emotion$EEG)
R>
R> emotion$id_repeated = rep(1:N, G)
R>
R> emotion$EEG <- scale(emotion$EEG, center = TRUE, scale = FALSE)
R>
R> X1h <- hmatrix(time = rep(emotion$t, each = N),
+   id = emotion$id_repeated,
+   x = emotion$EEG)
```

```
R> emotion$power.outcome <- interaction(emotion$power, emotion$game_outcome)
R> emotion$outcome.control <- interaction(emotion$game_outcome, emotion$control)
R> emotion$power.control <- interaction(emotion$power, emotion$control)
R> emotion$gamecondition <- interaction(emotion$power, emotion$game_outcome,
+                                       emotion$control)
R>
R> emotion$X1h <- I(X1h)
R>
R> mod <- FDboost(
+   EMG ~ 1 + brandomc(subject, df = 5) %A% bbs(t, df = 4) +
+   bolsc(power, df = 2, intercept = TRUE) %A% bbs(t, df = 10) +
+   bolsc(game_outcome, df = 2, intercept = TRUE) %A% bbs(t, df = 10) +
+   bolsc(control, df = 2, intercept = TRUE)  %A% bbs(t, df = 10)+
+   bolsc(power.outcome, intercept = TRUE, df = 2) %A% bbs(t, df = 10) +
+   bolsc(outcome.control, intercept = TRUE, df = 2) %A% bbs(t, df = 10) +
+   bolsc(power.control, intercept = TRUE, df = 2) %A% bbs(t, df = 10) +
+   bolsc(gamecondition, intercept = TRUE, df = 2) %A% bbs(t, df = 10) +
+   bhistx(X1h,
+     limits = function(s,t){ s < t - 3 },
+     df = 20, knots = 10,
+     differences = 2,
+     standard = "length"
+   ) +
+   bhistx(X1h,
+     limits = function(s,t){ s < t - 3 },
+     df = 5, knots = 10,
+     differences = 2,
+     standard = "length") %X%
+   bolsc(gamecondition, df = 4, intercept = TRUE,
+     index = id_repeated) +
+   bhistx(X1h,
+     limits = function(s,t){ s < t - 3 },
+     df = 5, knots = 10,
+     differences = 2,
+     standard = "length") %X%
+   brandomc(subject, df = 4, index = id_repeated),
+   control = boost_control(mstop = 5000, trace = TRUE),
+   timeformula = ~ bbs(t),
+   data = emotion
+ )
```

References

Brockhaus S, Fuest A, Mayr A, Greven S (2018). "Signal Regression Models for Location, Scale

and Shape with an Application to Stock Returns." *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, (3), 665–686.

Brockhaus S, Melcher M, Leisch F, Greven S (2017). "Boosting Flexible Functional Regression Models with a High Number of Functional Historical Effects." *Statistics and Computing*, **27**(4), 913–926.

Brockhaus S, Rügamer D (2017). ***FDboost**: Boosting Functional Regression Models*. R package version 0.3-0, URL http://CRAN.R-project.org/package=FDboost/.

Brockhaus S, Scheipl F, Hothorn T, Greven S (2015). "The Functional Linear Array Model." *Statistical Modelling*, **15**(3), 279–300.

Bühlmann P, Hothorn T (2007). "Boosting Algorithms: Regularization, Prediction and Model Fitting (with discussion)." *Statistical Science*, **22**(4), 477–505.

Bühlmann P, Yu B (2003). "Boosting with the L_2 Loss: Regression and Classification." *Journal of the American Statistical Association*, **98**(462), 324–339.

Currie ID, Durban M, Eilers PHC (2006). "Generalized Linear Array Models with Applications to Multidimensional Smoothing." *Journal of the Royal Statistical Society B*, **68**(2), 259–280.

Efron B (1979). "Bootstrap Methods: Another Look at the Jackknife." *The Annals of Statistics*, **7**(1), 1–26.

Eilers PHC, Marx BD (1996). "Flexible Smoothing with B-splines and Penalties (with comments and rejoinder)." *Statistical Science*, **11**(2), 89–121.

Fenske N, Kneib T, Hothorn T (2011). "Identifying Risk Factors for Severe Childhood Malnutrition by Boosting Additive Quantile Regression." *Journal of the American Statistical Association*, **106**(494), 494–510.

Friedman JH (2001). "Greedy Function Approximation: a Gradient Boosting Machine." *The Annals of Statistics*, **29**(5), 1189–1232.

Fuchs K, Scheipl F, Greven S (2015). "Penalized Scalar-on-Functions Regression with Interaction Term." *Computational Statistics & Data Analysis*, **81**, 38–51.

Gentsch K, Grandjean D, Scherer KR (2014). "Coherence Explored Between Emotion Components: Evidence from Event-Related Potentials and Facial Electromyography." *Biological Psychology*, **98**, 70 – 81.

Greven S, Scheipl F (2017). "A General Framework for Functional Regression Modelling." *Statistical Modelling*, **17**(1-2), 1–35.

Hastie TJ, Tibshirani RJ (1993). "Varying-Coefficient Models." *Journal of the Royal Statistical Society B*, **55**(4), 757–796.

Hofner B, Boccuto L, Göker M (2015). "Controlling False Discoveries in High-Dimensional Situations: Boosting with Stability Selection." *BMC Bioinformatics*, **16**(1), 1–17.

Hofner B, Hothorn T, Kneib T, Schmid M (2011). "A Framework for Unbiased Model Selection Based on Boosting." *Journal of Computational and Graphical Statistics*, **20**(4), 956–971.

Hofner B, Mayr A, Fenske N, Thomas J, Schmid M (2017). ***gamboostLSS**: Boosting Methods for GAMLSS Models*. R package version 2.0-0, URL https://CRAN.R-project.org/package=gamboostLSS.

Hofner B, Mayr A, Robinzonov N, Schmid M (2014). "Model-based Boosting in R: a Hands-on Tutorial using the R Package **mboost**." *Computational Statistics*, **29**(1), 3–35.

Hofner B, Mayr A, Schmid M (2016). "**gamboostLSS**: An R Package for Model Building and Variable Selection in the GAMLSS Framework." *Journal of Statistical Software*, **74**, 1–31.

Hothorn T, Bühlmann P, Kneib T, Schmid M, Hofner B (2016). ***mboost**: Model-Based Boosting*. R package version 2.8-0, URL http://CRAN.R-project.org/package=mboost.

Hothorn T, Kneib T, Bühlmann P (2013). "Conditional Transformation Models." *Journal of the Royal Statistical Society B*, **76**(1), 3–27.

Huang L, Scheipl F, Goldsmith J, Gellar JE, Harezlak J, McLean MW, Swihart B, Xiao L, Crainiceanu CM, Reiss PT (2016). ***refund**: Regression with Functional Data*. R package version 0.1-14, Available at https://cran.r-project.org/package=refund.

Huber PJ (1964). "Robust Estimation of a Location Parameter." *The Annals of Mathematical Statistics*, **35**(1), 73–101.

Kneib T, Hothorn T, Tutz G (2009). "Variable Selection and Model Choice in Geoadditive Regression Models." *Biometrics*, **65**(2), 626–634. ISSN 1541-0420.

Koenker R (2005). *Quantile Regression*. Cambridge University Press, Cambridge.

Mayr A, Fenske N, Hofner B, Kneib T, Schmid M (2012). "Generalized Additive Models for Location, Scale and Shape for High Dimensional Data – a Flexible Approach Based on Boosting." *Journal of the Royal Statistical Society C*, **61**(3), 403–427.

Meinshausen N, Bühlmann P (2010). "Stability Selection (with discussion)." *Journal of the Royal Statistical Society B*, **72**(4), 417–473.

Meyer MJ, Coull BA, Versace F, Cinciripini P, Morris JS (2015). "Bayesian Function-on-Function Regression for Multilevel Functional Data." *Biometrics*, **71**(3), 563–574. ISSN 1541-0420.

Morris JS (2015). "Functional Regression." *Annual Review of Statistics and Its Application*, **2**(1), 321–359.

Morris JS (2017). "Comparison and Contrast of Two General Functional Regression Modelling Frameworks." *Statistical Modelling*, **17**(1-2), 59–85.

Morris JS, Carroll RJ (2006). "Wavelet-Based Functional Mixed Models." *Journal of the Royal Statistical Society B*, **68**(2), 179–199.

Newey WK, Powell JL (1987). "Asymmetric Least Squares Estimation and Testing." *Econometrica: Journal of the Econometric Society*, **55**(4), 819–847.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. R 3.4.0, URL http://www.R-project.org/.

Ramsay JO, Silverman BW (2005). *Functional Data Analysis*. Springer-Verlag, New York.

Ridgeway G (1999). "The state of boosting." *Computing Science and Statistics*, pp. 172–181.

Ridgeway G (2017). ***gbm**: Generalized Boosted Regression Models*. R package version 2.1.3, URL https://CRAN.R-project.org/package=gbm.

Rigby RA, Stasinopoulos DM (2005). "Generalized Additive Models for Location, Scale and Shape (with discussion)." *Journal of the Royal Statistical Society C*, **54**(3), 507–554.

Rügamer D, Brockhaus S, Gentsch K, Scherer K, Greven S (2018). "Boosting factor-specific functional historical models for the detection of synchronization in bioelectrical signals." *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **67**(3), 621–642. doi:10.1111/rssc.12241.

Ruppert D, Wand MP, Carroll RJ (2003). *Semiparametric Regression*. Cambridge University Press.

Scheipl F, Gertheiss J, Greven S (2016). "Generalized Functional Additive Mixed Models." *Electronic Journal of Statistics*, **10**(1), 1455–1492.

Scheipl F, Greven S (2016). "Identifiability in Penalized Function-On-Function Regression Models." *Electronic Journal of Statistics*, **10**(1), 495–526.

Scheipl F, Staicu AM, Greven S (2015). "Functional Additive Mixed Models." *Journal of Computational and Graphical Statistics*, **24**(2), 477–501.

Schmid M, Hothorn T (2008a). "Boosting Additive Models Using Component-Wise P-splines." *Computational Statistics & Data Analysis*, **53**(2), 298–311.

Schmid M, Hothorn T (2008b). "Flexible Boosting of Accelerated Failure Time Models." *BMC Bioinformatics*, **9**(1), 1–13. ISSN 1471-2105.

Schmid M, Hothorn T, Maloney KO, Weller DE, Potapov S (2011). "Geoadditive Regression Modeling of Stream Biological Condition." *Environmental and Ecological Statistics*, **18**(4), 709–733.

Schmid M, Potapov S, Pfahlberg A, Hothorn T (2010). "Estimation and Regularization Techniques for Regression Models with Multidimensional Prediction Functions." *Statistics and Computing*, **20**(2), 139–150.

Shah RD, Samworth RJ (2013). "Variable Selection with Error Control: another Look at Stability Selection." *Journal of the Royal Statistical Society B*, **75**(1), 55–80.

Sobotka F, Kneib T (2012). "Geoadditive Expectile Regression." *Computational Statistics & Data Analysis*, **56**(4), 755–767.

Stöcker A, Brockhaus S, Schaffer S, von Bronk B, Opitz M, Greven S (2017). "Boosting Functional Response Models for Location, Scale and Shape with an Application to Bacterial Competition." Unpublished working paper.

Thomas J, Mayr A, Bischl B, Schmid M, Smith A, Hofner B (2018). "Gradient boosting for distributional regression: faster tuning and improved variable selection via noncyclical updates." *Statistics and Computing*, **28**(3), 673–687. Doi:10.1007/s11222-017-9754-6.

Ullah S, Finch CF (2013). "Applications of Functional Data Analysis: a Systematic Review." *BMC Medical Research Methodology*, **13**(43), 1–12.

Wood SN (2006). *Generalized Additive Models: An Introduction with R*. Chapman & Hal/CRC, Boca Raton, Florida.

Wood SN (2011). "Fast Stable Restricted Maximum Likelihood and Marginal Likelihood Estimation of Semiparametric Generalized Linear Models." *Journal of the Royal Statistical Society B*, **73**(1), 3–36.

Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12. Juli 2011, §8 Abs. 2 Pkt. 5)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

München, den 08.05.2018 David Rügamer