Andreas Groll & Gerhard Tutz

Regularization for Generalized Additive Mixed Models by Likelihood-Based Boosting

Technical Report Number 110, 2011
Department of Statistics
University of Munich

http://www.stat.uni-muenchen.de

Regularization for Generalized Additive Mixed Models by Likelihood-Based Boosting

Andreas Groll [*] Gerhard Tutz [†]

June 29, 2011

Abstract

With the emergence of semi- and nonparametric regression the generalized linear mixed model has been expanded to account for additive predictors. In the present paper an approach to variable selection is proposed that works for generalized additive mixed models. In contrast to common procedures it can be used in high-dimensional settings where many covariates are available and the form of the influence is unknown. It is constructed as a componentwise boosting method and hence is able to perform variable selection. The complexity of the resulting estimator is determined by information criteria. The method is investigated in simulation studies for binary and Poisson responses and is illustrated by using real data sets.

Keywords: Generalized additive mixed model, Boosting, Smoothing, Variable selection, Penalized Quasi-Likelihood, Laplace approximation

[*] Department of Statistics, University of Munich, Akademiestrasse 1, D-80799, Munich, Germany, *email*: andreas.groll@stat.uni-muenchen.de

[†] Department of Statistics, University of Munich, Akademiestrasse 1, D-80799, Munich, Germany, *email*: tutz@stat.uni-muenchen.de

1 Introduction

General additive mixed models (GAMMs) are an extension of generalized additive models incorporating random effects. In the present article a boosting approach for the selection of additive predictors is proposed. Boosting originates in the machine learning community and turned out to be a successful and practical strategy to improve classification procedures by combining estimates with reweighted observations. The idea of boosting has become especially important in the last decade as the issue of estimating high-dimensional models has become more urgent. Since Freund and Schapire (1996) have presented their famous AdaBoost many extensions have been developed (e.g. gradient boosting by Friedman et al., 2000, generalized linear and additive regression based on the L_2-loss by Bühlmann and Yu, 2003).

In the following the concept of likelihood-based boosting is extended to GAMMs which are sketched in Section 2. The fitting procedure is outlined in Section 3 and a simulation study is reported in Section 4. Finally, two applications are considered in Section 5.

2 Generalized Additive Mixed Models - GAMMs

Let y_{it} denote observation t in cluster i, $i = 1, \ldots, n$, $t = 1, \ldots, T_i$, collected in $\mathbf{y}_i^T = (y_{i1}, \ldots, y_{iT_i})$. Let $\mathbf{x}_{it}^T = (1, x_{it1}, \ldots, x_{itp})$ be the covariate vector associated with fixed effects and $\mathbf{z}_{it}^T = (z_{it1}, \ldots, z_{itq})$ the covariate vector associated with random effects. It is assumed that the observations y_{it} are conditionally independent with means $\mu_{it} = E(y_{it}|\mathbf{b}_i, \mathbf{x}_{it}, \mathbf{z}_{it})$ and variances $var(y_{it}|\mathbf{b}_i) = \phi v(\mu_{it})$, where $v(.)$ is a known variance function and ϕ is a scale parameter.

In addition to parametric effects the model that is considered includes an additive term that depends on covariates $\mathbf{u}_{it}^T = (u_{it1}, \ldots, u_{itm})$. The generalized semiparametric mixed model that is assumed to hold is given by

$$\begin{aligned} g(\mu_{it}) &= \mathbf{x}_{it}^T\boldsymbol{\beta} + \sum_{j=1}^{m} \alpha_{(j)}(u_{itj}) + \mathbf{z}_{it}^T\mathbf{b}_i \qquad (1) \\ &= \eta_{it}^{\text{par}} + \eta_{it}^{\text{add}} + \eta_{it}^{\text{rand}}, \end{aligned}$$

where g is a monotonic differentiable link function, $\eta_{it}^{\text{par}} = \mathbf{x}_{it}^T\boldsymbol{\beta}$ is a linear parametric term with parameter vector $\boldsymbol{\beta}^T = (\beta_0, \beta_1, \ldots, \beta_p)$, including the intercept, $\eta_{it}^{\text{add}} = \sum_{j=1}^{m} \alpha_{(j)}(u_{itj})$ is an additive term with unspecified influence functions $\alpha_{(1)}, \ldots, \alpha_{(m)}$ and finally $\eta_{it}^{\text{rand}} = \mathbf{z}_{it}^T\mathbf{b}_i$ contains the cluster-specific random effects $\mathbf{b}_i \sim N(0, \mathbf{Q})$, where $\mathbf{Q}$ is a $q \times q$ dimensional known or unknown covariance matrix. An alternative form that we also use in the following is

$$\mu_{it} = h(\eta_{it}), \quad \eta_{it} = \eta_{it}^{\text{par}} + \eta_{it}^{\text{add}} + \eta_{it}^{\text{rand}},$$

where $h = g^{-1}$ is the inverse link function. If the functions $\alpha_{(j)}(\cdot)$ are strictly linear, the model reduces to the common generalized linear mixed model (GLMM). Versions of the additive model (1) have been considered by Zeger and Diggle (1994), Lin and Zhang (1999) and Zhang et al. (1998). While Lin and Zhang (1999) used natural cubic smoothing splines for the estimation of the unknown functions $\alpha_{(j)}(\cdot)$, in the following regression splines are used. In recent years regression splines have been widely used for the estimation of additive structures, see, for example, Marx and Eilers (1998), Wood (2004, 2006) and Wand (2000).

In regression spline methodology the unknown functions $\alpha_{(j)}(\cdot)$ are approximated by basis functions. A simple basis is known as the B-spline basis of degree d, yielding

$$\alpha_{(j)}(u) = \sum_{i=1}^{k} \alpha_i^{(j)} \, B_i^{(j)}(u; d),$$

where $B_i^{(j)}(u; d)$ denotes the i-th basis function for variable j. For an extensive discussion of smoothing by using splines, see for example Ruppert et al. (2003). More detailed information about the B-spline basis can be found for example in Eilers and Marx (1996).

In the following let $\boldsymbol{\alpha}_j^T = (\alpha_1^{(j)}, \dots, \alpha_k^{(j)})$ denote the unknown parameter vector of the j-th smooth function and let $\mathbf{B}_j^T(u) = (B_1^{(j)}(u; d), \dots, B_k^{(j)}(u; d))$ represent the vector-valued evaluations of the k basis functions. Then the parameterized model for (1) has the form

$$g(\mu_{it}) = \mathbf{x}_{it}^T \boldsymbol{\beta} + \mathbf{B}_1^T(u_{it1})\boldsymbol{\alpha}_1 + \dots + \mathbf{B}_m^T(u_{itm})\boldsymbol{\alpha}_m + \mathbf{z}_{it}^T \mathbf{b}.$$

By collecting observations within one cluster one obtains the design matrix $\mathbf{X}_i^T = (\mathbf{x}_{i1}, \dots, \mathbf{x}_{iT_i})$ for the i-th covariate, and analogously we set $\mathbf{Z}_i^T = (\mathbf{z}_{i1}, \dots, \mathbf{z}_{iT_i})$, so that the model has the simpler form

$$g(\boldsymbol{\mu}_i) = \mathbf{X}_i \boldsymbol{\beta} + \mathbf{B}_{i1}\boldsymbol{\alpha}_1 + \dots + \mathbf{B}_{im}\boldsymbol{\alpha}_m + \mathbf{Z}_i \mathbf{b}_i,$$

where $\mathbf{B}_{ij}^T = [\mathbf{B}_j(u_{i1j}), \dots, \mathbf{B}_j(u_{iT_ij})]$ denotes the transposed B-spline design matrix of the i-th cluster and variable j and g is understood componentwise. Furthermore, let $\mathbf{X}^T = [\mathbf{X}_1^T, \dots, \mathbf{X}_n^T]$, let $\mathbf{Z} = diag(\mathbf{Z}_1, \dots, \mathbf{Z}_n)$ be a block-diagonal matrix and let $\mathbf{b}^T = (\mathbf{b}_1^T, \dots, \mathbf{b}_n^T)$ be the vector collecting all random effects. Then one obtains the model in the matrix form

$$g(\boldsymbol{\mu}) = \mathbf{X}\boldsymbol{\beta} + \mathbf{B}_1\boldsymbol{\alpha}_1 + \ldots + \mathbf{B}_m\boldsymbol{\alpha}_m + \mathbf{Z}\mathbf{b}, \tag{2}$$

with $\mathbf{B}_j^T = [\mathbf{B}_{1j}^T, \ldots, \mathbf{B}_{nj}^T]$ representing the transposed B-spline design matrix of the j-th smooth function as in equation (13) in Appendix A. The model can be further reduced to

$$g(\boldsymbol{\mu}) = \mathbf{X}\boldsymbol{\beta} + \mathbf{B}\boldsymbol{\alpha} + \mathbf{Z}\mathbf{b},$$

where $\boldsymbol{\alpha}^T = (\boldsymbol{\alpha}_1^T, \ldots, \boldsymbol{\alpha}_m^T)$ and $\mathbf{B} = [\mathbf{B}_1, \ldots, \mathbf{B}_m]$.

The Penalized Likelihood Approach

Focusing on generalized mixed models we assume that the conditional density of y_{it}, given explanatory variables and the random effect $\mathbf{b}_i$, is of exponential family type

$$f(y_{it}|\mathbf{x}_{it}, \mathbf{u}_{it}, \mathbf{b}_i) = \exp\left\{\frac{(y_{it}\theta_{it} - \kappa(\theta_{it}))}{\phi} + c(y_{it}, \phi)\right\}, \tag{3}$$

where $\theta_{it} = \theta(\mu_{it})$ denotes the natural parameter, $\kappa(\theta_{it})$ is a specific function corresponding to the type of exponential family, $c(.)$ the log normalization constant and ϕ the dispersion parameter (for example Fahrmeir and Tutz, 2001).

A popular method to maximize generalized mixed models is penalized quasi-likelihood (PQL), which has been suggested by Breslow and Clayton (1993), Lin and Breslow (1996) and Breslow and Lin (1995). In the following we briefly sketch the PQL approach for the semiparametric model. As common in mixed models, we assume that the covariance matrix $\mathbf{Q}(\boldsymbol{\varrho})$ of the random effects $\mathbf{b}_i$ may depend on an unknown parameter vector $\boldsymbol{\varrho}$ which specifies the correlation. We specify the joint likelihood-function by the parameters of the covariance structure $\boldsymbol{\varrho}$ together with the dispersion parameter ϕ, which are collected in $\boldsymbol{\nu}^T = (\phi, \boldsymbol{\varrho}^T)$ and define the parameter vector $\boldsymbol{\delta}^T = (\boldsymbol{\beta}^T, \boldsymbol{\alpha}^T, \mathbf{b}^T)$. The corresponding log-likelihood is

$$l(\boldsymbol{\delta}, \boldsymbol{\nu}) = \sum_{i=1}^{n} \log\left(\int f(\mathbf{y}_i|\boldsymbol{\delta}, \boldsymbol{\nu}) p(\mathbf{b}_i, \boldsymbol{\nu}) d\mathbf{b}_i\right).$$

To avoid too severe restrictions on the form of the functions $\alpha_{(j)}(\cdot)$, we use many basis functions, say about 20 for each function $\alpha_{(j)}(.)$, and add a penalty term to the log-likelihood. Then one obtains the penalized log-likelihood

$$l^{\text{pen}}(\boldsymbol{\delta}, \boldsymbol{\nu}) = \sum_{i=1}^{n} \log\left(\int f(\mathbf{y}_i|\boldsymbol{\delta}, \boldsymbol{\nu}) p(\mathbf{b}_i, \boldsymbol{\nu}) d\mathbf{b}_i\right) - \frac{1}{2}\sum_{j=1}^{m} \lambda_j \boldsymbol{\alpha}_j^T \mathbf{K}_j \boldsymbol{\alpha}_j, \tag{4}$$

where $\mathbf{K}_j$ penalizes the parameters $\boldsymbol{\alpha}_j$ and λ_j are smoothing parameters which control the influence of the j-th penalty term. When using P-splines one penalizes the difference between adjacent categories in the form $\lambda_j \boldsymbol{\alpha}_j^T \mathbf{K}_j \boldsymbol{\alpha}_j = \lambda_j \boldsymbol{\alpha}_j^T (\boldsymbol{\Delta}^d)^T \boldsymbol{\Delta}^d \boldsymbol{\alpha}_j$, where $\boldsymbol{\Delta}^d$ denotes the difference operator matrix of degree d, for details see, for example, Eilers and Marx (1996). The log-likelihood (4) has also been considered by Lin and Zhang (1999) but with $\mathbf{K}_j$ referring to smoothing splines. For smoothing splines the dimension of $\boldsymbol{\alpha}_j$ increases with sample size whereas for the low rank smoother used here the dimension does not depend on n.

By approximating the likelihood in (4) along the lines of Breslow and Clayton (1993) one obtains the double penalized log-likelihood:

$$l^{\text{pen}}(\boldsymbol{\delta},\boldsymbol{\nu}) = \sum_{i=1}^{n} \log(f(\mathbf{y}_i|\boldsymbol{\delta},\boldsymbol{\nu})) - \frac{1}{2}\sum_{i=1}^{n} \mathbf{b}_i^T \mathbf{Q}(\boldsymbol{\varrho})^{-1}\mathbf{b}_i - \frac{1}{2}\sum_{j=1}^{m} \lambda_j \boldsymbol{\alpha}_j^T \mathbf{K}_j \boldsymbol{\alpha}_j, \tag{5}$$

where the first penalty term $\sum_{i=1}^{n} \mathbf{b}_i^T \mathbf{Q}(\boldsymbol{\varrho})^{-1}\mathbf{b}_i$ is due to the approximation based on the Laplace method and the second penalty term $\sum_{j=1}^{m} \lambda_j \boldsymbol{\alpha}_j^T \mathbf{K}_j \boldsymbol{\alpha}_j$ determines the smoothness of the functions $\alpha_{(j)}(.)$, depending on the chosen smoothing parameter λ_j.

PQL usually works within the profile likelihood concept. It is distinguished between the estimation of $\boldsymbol{\delta}$, given the plug-in estimate $\hat{\boldsymbol{\nu}}$, resulting in the profile-likelihood $l^{\text{pen}}(\boldsymbol{\delta},\hat{\boldsymbol{\nu}})$, and the estimation of $\boldsymbol{\nu}$. The PQL method for generalized additive mixed models is implemented in the `gamm` function of the R-package `mgcv` (Wood, 2006). Further aspects were discussed by Wolfinger and O'Connell (1993), Littell et al. (1996) and Vonesh (1996).

Note that the double penalized log-likelihood from equation (5) can also be derived by an EM-type algorithm, using posterior modes and curvatures instead of posterior means and covariances (see, for example, Fahrmeir and Tutz, 2001).

3 Boosted GAMMs - `bGAMM`

Boosting originates in the machine learning community and turned out to be a successful and practical strategy to improve classification procedures by combining estimates with reweighted observations. The idea of boosting has become more and more important in the last decade as the issue of estimating high-dimensional models has become more urgent. Since Freund and Schapire (1996) have presented their famous AdaBoost many other variants in the framework of functional gradient descent optimization have been developed (for example Friedman et al., 2000 or Friedman, 2001). Bühlmann and Yu (2003) further extended boosting to generalized linear and additive regression problems based on the L_2-loss.

Boosting is especially successful as a method to select relevant predictors in linear and generalized linear models. For extensions to GLMMs, see Tutz and Groll (2011). It works by iterative fitting of residuals using "weak learners". The boosting algorithm that is presented in the following extends the method to additive mixed models.

3.1 The Boosting Algorithm

The following algorithm uses componentwise boosting, that is, only one component of the additive predictor, in our case one weight vector $\boldsymbol{\alpha}_j$, is fitted at a time. That means that a model containing the linear term and only one smooth component is fitted in one iteration step. We use a reparametrization technique explained in more detail in Appendix A. The B-spline design matrices $\mathbf{B}_j$ from equation (2), corresponding to the difference penalty matrices $\mathbf{K}_j$ and spline coefficients $\boldsymbol{\alpha}_j$, can be transformed to new design matrices $\boldsymbol{\Phi}_j$ with spline coefficients $\tilde{\boldsymbol{\alpha}}_j$, which consist of an unpenalized and a penalized part and correspond to diagonal penalty matrices $\tilde{\mathbf{K}} := \tilde{\mathbf{K}}_j = diag(0, \ldots, 0, 1, \ldots, 1)$, which are equal for all $j = 1, \ldots, m$. We drop the first column of each matrix $\boldsymbol{\Phi}_j$, because we are in the semiparametric model context (see Appendix B).

The predictor containing all covariates associated with fixed effects and only the covariate vector of the r-th smooth effect yields for cluster i

$$\boldsymbol{\eta}_{i \cdot r} = \mathbf{X}_i \boldsymbol{\beta} + \boldsymbol{\Phi}_{ir} \tilde{\boldsymbol{\alpha}}_r + \mathbf{Z}_i \mathbf{b}_i,$$

where $\boldsymbol{\Phi}_{ir}$ is a sub-matrix of $\boldsymbol{\Phi}_r$, consisting of only the T_i rows from $\boldsymbol{\Phi}_r$ corresponding to cluster i. Altogether the predictor, considering only the r-th smooth effect, has the form

$$\boldsymbol{\eta}_{\cdot \cdot r} = \mathbf{X} \boldsymbol{\beta} + \boldsymbol{\Phi}_r \tilde{\boldsymbol{\alpha}}_r + \mathbf{Z}\mathbf{b}.$$

Moreover, we define $\boldsymbol{\Phi} := [\boldsymbol{\Phi}_1, \ldots, \boldsymbol{\Phi}_m]$ and introduce the new parameter vector $\boldsymbol{\gamma}^T := (\boldsymbol{\beta}^T, \tilde{\boldsymbol{\alpha}}^T, \mathbf{b}^T)$. The following boosting algorithm uses the EM-type algorithm given in Fahrmeir and Tutz (2001). We further want to introduce the vector $\boldsymbol{\gamma}_r^T := (\boldsymbol{\beta}^T, \tilde{\boldsymbol{\alpha}}_r^T, \mathbf{b}^T)$, containing only the spline coefficients of the r-th smooth component.

Algorithm `bGAMM`

---

1. *Initialization*

   Compute starting values $\hat{\boldsymbol{\beta}}^{(0)}, \hat{\tilde{\boldsymbol{\alpha}}}^{(0)}, \hat{\mathbf{b}}^{(0)}, \hat{\mathbf{Q}}^{(0)}$ and set $\hat{\boldsymbol{\eta}}^{(0)} = \mathbf{X}\hat{\boldsymbol{\beta}}^{(0)} + \boldsymbol{\Phi}\hat{\tilde{\boldsymbol{\alpha}}}^{(0)} + \mathbf{Z}\hat{\mathbf{b}}^{(0)}$.

2. *Iteration*

   For $l = 1, 2, \ldots$

   (a) *Refitting of residuals*

   (i.) Computation of parameters

For $r \in \{1, \dots, m\}$ the model

$$g(\boldsymbol{\mu}) = \hat{\boldsymbol{\eta}}^{(l-1)} + \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\Phi}_r\tilde{\boldsymbol{\alpha}}_r + \mathbf{Z}\mathbf{b}$$

is fitted, where $\hat{\boldsymbol{\eta}}^{(l-1)} = \mathbf{X}\hat{\boldsymbol{\beta}}^{(l-1)} + \boldsymbol{\Phi}\hat{\tilde{\boldsymbol{\alpha}}}^{(l-1)} + \mathbf{Z}\hat{\mathbf{b}}^{(l-1)}$ is considered a known off-set. Estimation refers to $\boldsymbol{\gamma}_r^T = (\boldsymbol{\beta}^T, \tilde{\boldsymbol{\alpha}}_r^T, \mathbf{b}^T)$. In order to obtain an additive correction of the already fitted terms, we use one-step Fisher scoring with starting value $\boldsymbol{\gamma}_r = \mathbf{0}$. Therefore Fisher scoring for the r-th component takes the simple form

$$\hat{\boldsymbol{\gamma}}_r^{(l)} = (\mathbf{F}_r^{\text{pen}}(\hat{\boldsymbol{\gamma}}^{(l-1)}))^{-1}\mathbf{s}_r(\hat{\boldsymbol{\gamma}}^{(l-1)}) \tag{6}$$

with penalized pseudo Fisher matrix $\mathbf{F}_r^{\text{pen}}(\boldsymbol{\gamma})$ and using the unpenalized version of the penalized score function $\mathbf{s}_r^{\text{pen}}(\boldsymbol{\gamma}) = \partial l^{\text{pen}}(\boldsymbol{\gamma})/\partial\boldsymbol{\gamma}_r$ (see Section 3.2.1). The variance-covariance components are replaced by their current estimates $\hat{\mathbf{Q}}^{(l-1)}$.

(ii.) Selection step

Select from $r \in \{1, \dots, m\}$ the component j that leads to the smallest $AIC_r^{(l)}$ or $BIC_r^{(l)}$ as given in Section 3.2.3 and select the corresponding vector $(\hat{\boldsymbol{\gamma}}_j^{(l)})^T = \left((\hat{\boldsymbol{\beta}}^*)^T, (\hat{\tilde{\boldsymbol{\alpha}}}_j^*)^T, (\hat{\mathbf{b}}^*)^T\right)$.

(iii.) Update

Set

$$\hat{\boldsymbol{\beta}}^{(l)} = \hat{\boldsymbol{\beta}}^{(l-1)} + \hat{\boldsymbol{\beta}}^*, \qquad \hat{\mathbf{b}}^{(l)} = \hat{\mathbf{b}}^{(l-1)} + \hat{\mathbf{b}}^*$$

and for $r = 1, \dots, m$ set

$$\hat{\tilde{\boldsymbol{\alpha}}}_r^{(l)} = \begin{cases} \hat{\tilde{\boldsymbol{\alpha}}}_r^{(l-1)} & \text{if } r \neq j \\ \hat{\tilde{\boldsymbol{\alpha}}}_r^{(l-1)} + \hat{\tilde{\boldsymbol{\alpha}}}_r^* & \text{if } r = j, \end{cases}$$

$$(\hat{\boldsymbol{\gamma}}^{(l)})^T = \left((\hat{\boldsymbol{\beta}}^{(l)})^T, (\hat{\tilde{\boldsymbol{\alpha}}}_1^{(l)})^T, \dots, (\hat{\tilde{\boldsymbol{\alpha}}}_m^{(l)})^T, (\hat{\mathbf{b}}^{(l)})^T\right).$$

With $\mathbf{A} := [\mathbf{X}, \boldsymbol{\Phi}, \mathbf{Z}]$ update

$$\hat{\boldsymbol{\eta}}^{(l)} = \mathbf{A}\hat{\boldsymbol{\gamma}}^{(l)}$$

(b) *Computation of variance-covariance components*

Estimates of $\hat{\mathbf{Q}}^{(l)}$ are obtained as approximate REML-type estimates or alternative methods (see Section 3.2.2)

Note that the EM-type algorithm may be viewed as an approximate EM algorithm, where the posterior of b_i is approximated by a normal distribution. In the case of linear random effects models, the EM-type algorithm corresponds to an exact EM algorithm since the posterior of b_i is normal, and so posterior mode and mean coincide, as do posterior covariance and curvature.

3.2 Computational details of `bGAMM`

In the following we give a more detailed description of the single steps of the `bGAMM` algorithm. First the derivation of the score function and the Fisher matrix are described. Then we present two estimation techniques for the variance-covariance components, give the details of the computation of the starting values and explain the selection procedure.

3.2.1 Score Function and Fisher Matrix

In this section we specify more precisely the single components which are derived in step 2 (a) of the `bGAMM` algorithm. For $r \in \{1,\dots,p\}$ the penalized score functions $\mathbf{s}_r^{\text{pen}}(\boldsymbol{\gamma})$ are obtained by differentiating the penalized log-likelihood from equation (5) with respect to $\boldsymbol{\gamma}_r$, that is $\mathbf{s}_r^{\text{pen}}(\boldsymbol{\gamma}) = \partial l^{\text{pen}}(\boldsymbol{\gamma})/\partial \boldsymbol{\gamma}_r$. To keep the notation simple, we omit the argument $\boldsymbol{\gamma}$ in the following and write $\mathbf{s}_r^{\text{pen}\,(l-1)} = \Big((\mathbf{s}_{\boldsymbol{\beta}r}^{\text{pen}\,(l-1)})^T, (\mathbf{s}_{\tilde{\boldsymbol{\alpha}}_r r}^{\text{pen}\,(l-1)})^T, (\mathbf{s}_{1r}^{\text{pen}\,(l-1)})^T, \dots, (\mathbf{s}_{nr}^{\text{pen}\,(l-1)})^T\Big)^T = \mathbf{s}_r^{\text{pen}}(\hat{\boldsymbol{\gamma}}^{(l-1)})$ for the r-th evaluated penalized score function at $(l-1)$-th iteration. For given $\mathbf{Q}$, it has single components

$$
\begin{aligned}
\mathbf{s}_{\boldsymbol{\beta}r}^{\text{pen}\,(l-1)} &= \sum_{i=1}^{n} \mathbf{X}_i^T \mathbf{D}_i \boldsymbol{\Sigma}_i^{-1} (\mathbf{y}_i - \hat{\boldsymbol{\mu}}_i), \\
\mathbf{s}_{\tilde{\boldsymbol{\alpha}}_r r}^{\text{pen}\,(l-1)} &= \sum_{i=1}^{n} \boldsymbol{\Phi}_{ir}^T \mathbf{D}_i \boldsymbol{\Sigma}_i^{-1} (\mathbf{y}_i - \hat{\boldsymbol{\mu}}_i) - \lambda \tilde{\mathbf{K}} \hat{\tilde{\boldsymbol{\alpha}}}_r^{(l-1)}, \\
\mathbf{s}_{ir}^{\text{pen}\,(l-1)} &= \mathbf{Z}_i^T \mathbf{D}_i \boldsymbol{\Sigma}_i^{-1} (\mathbf{y}_i - \hat{\boldsymbol{\mu}}_i) - \mathbf{Q}^{-1} \hat{\mathbf{b}}_i^{(l-1)}, \quad i = 1,\dots,n,
\end{aligned}
$$

with $\mathbf{D}_i = \partial h(\hat{\boldsymbol{\eta}}_i)/\partial \boldsymbol{\eta}, \boldsymbol{\Sigma}_i = cov(\mathbf{y}_i)$, and $\hat{\boldsymbol{\mu}}_i = h(\hat{\boldsymbol{\eta}}_i)$ evaluated at previous fit $\hat{\boldsymbol{\eta}}_i = \mathbf{A}_i \hat{\boldsymbol{\gamma}}^{(l-1)}$, whereas $\mathbf{A}_i := [\mathbf{X}_i, \boldsymbol{\Phi}_i, \mathbf{Z}_i]$. One should keep in mind that actually, $\mathbf{D}_i, \boldsymbol{\Sigma}_i, \boldsymbol{\mu}_i$ and $\boldsymbol{\eta}_i$ are depending on $\hat{\boldsymbol{\gamma}}^{(l-1)}$ and thus on the current iteration, which is suppressed here to keep the notation simple. The vector $\mathbf{s}_{\boldsymbol{\beta}r}^{\text{pen}\,(l-1)}$ has dimension $p+1$, the vector $\mathbf{s}_{\tilde{\boldsymbol{\alpha}}_r r}^{\text{pen}\,(l-1)}$ has dimension k corresponding to the number of basis functions, while the vectors $\mathbf{s}_{ir}^{\text{pen}\,(l-1)}$ are of dimension s. Note that $\mathbf{s}_r^{\text{pen}\,(l-1)}$ could be seen as penalized score function because of the terms $\lambda \tilde{\mathbf{K}} \hat{\tilde{\boldsymbol{\alpha}}}_r^{(l-1)}$ and $\mathbf{Q}^{-1} \hat{\mathbf{b}}_i^{(l-1)}$.

Let $\tilde{\boldsymbol{\beta}}_r^T := (\boldsymbol{\beta}^T, \tilde{\boldsymbol{\alpha}}_r^T)$. Then the penalized pseudo Fisher matrix $\mathbf{F}_r^{\text{pen}\,(l-1)}$, $r \in \{1,\dots,m\}$, which is partitioned into

$$\mathbf{F}_r^{\text{pen}\,(l-1)} = \begin{bmatrix} \mathbf{F}_{\tilde{\boldsymbol{\beta}}_r\tilde{\boldsymbol{\beta}}_r r} & \mathbf{F}_{\tilde{\boldsymbol{\beta}}_r 1r} & \mathbf{F}_{\tilde{\boldsymbol{\beta}}_r 2r} & \cdots & \mathbf{F}_{\tilde{\boldsymbol{\beta}}_r nr} \\ \mathbf{F}_{1\tilde{\boldsymbol{\beta}}_r r} & \mathbf{F}_{11r} & & & 0 \\ \mathbf{F}_{2\tilde{\boldsymbol{\beta}}_r r} & & \mathbf{F}_{22r} & & \\ \vdots & & & \ddots & \\ \mathbf{F}_{n\tilde{\boldsymbol{\beta}}_r r} & 0 & & & \mathbf{F}_{nnr} \end{bmatrix}, \quad \text{with} \quad \mathbf{F}_{\tilde{\boldsymbol{\beta}}_r\tilde{\boldsymbol{\beta}}_r r} = \begin{bmatrix} \mathbf{F}_{\boldsymbol{\beta}\boldsymbol{\beta}r} & \mathbf{F}_{\boldsymbol{\beta}\tilde{\boldsymbol{\alpha}}_r r} \\ \mathbf{F}_{\tilde{\boldsymbol{\alpha}}_r\boldsymbol{\beta}r} & \mathbf{F}_{\tilde{\boldsymbol{\alpha}}_r\tilde{\boldsymbol{\alpha}}_r r} \end{bmatrix},$$

has single components

$$\begin{aligned}
\mathbf{F}_{\boldsymbol{\beta}\boldsymbol{\beta}r} &= -E\left(\frac{\partial^2 l^{\text{pen}}}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^T}\right) = \sum_{i=1}^{n} \mathbf{X}_i^T\mathbf{D}_i\boldsymbol{\Sigma}_i^{-1}\mathbf{D}_i\mathbf{X}_i \\
\mathbf{F}_{\boldsymbol{\beta}\tilde{\boldsymbol{\alpha}}_r r} &= \mathbf{F}_{\tilde{\boldsymbol{\alpha}}_r\boldsymbol{\beta}r}^T = -E\left(\frac{\partial^2 l^{\text{pen}}}{\partial\boldsymbol{\beta}\partial\tilde{\boldsymbol{\alpha}}_r^T}\right) = \sum_{i=1}^{n} \mathbf{X}_i^T\mathbf{D}_i\boldsymbol{\Sigma}_i^{-1}\mathbf{D}_i\boldsymbol{\Phi}_{ir}, \\
\mathbf{F}_{\tilde{\boldsymbol{\alpha}}_r\tilde{\boldsymbol{\alpha}}_r r} &= -E\left(\frac{\partial^2 l^{\text{pen}}}{\partial\tilde{\boldsymbol{\alpha}}_r\partial\tilde{\boldsymbol{\alpha}}_r^T}\right) = \sum_{i=1}^{n} \boldsymbol{\Phi}_{ir}^T\mathbf{D}_i\boldsymbol{\Sigma}_i^{-1}\mathbf{D}_i\boldsymbol{\Phi}_{ir} - \lambda\tilde{\mathbf{K}}, \\
\mathbf{F}_{\tilde{\boldsymbol{\beta}}_r ir} &= \mathbf{F}_{i\tilde{\boldsymbol{\beta}}_r r}^T = -E\left(\frac{\partial^2 l^{\text{pen}}}{\partial\tilde{\boldsymbol{\beta}}_r\partial\mathbf{b}_i^T}\right) = [\mathbf{X}_i, \boldsymbol{\Phi}_{ir}]^T\mathbf{D}_i\boldsymbol{\Sigma}_i^{-1}\mathbf{D}_i\mathbf{Z}_i, \\
\mathbf{F}_{iir} &= -E\left(\frac{\partial^2 l^{\text{pen}}}{\partial\mathbf{b}_i\partial\mathbf{b}_i^T}\right) = \mathbf{Z}_i^T\mathbf{D}_i\boldsymbol{\Sigma}_i^{-1}\mathbf{D}_i\mathbf{Z}_i + \mathbf{Q}^{-1}.
\end{aligned}$$

whereas $\mathbf{D}_i = \partial h(\hat{\boldsymbol{\eta}}_i)/\partial\boldsymbol{\eta}$ and $\boldsymbol{\Sigma}_i = cov(\mathbf{y}_i)$ again are evaluated at the previous fit $\hat{\boldsymbol{\eta}}_i = \mathbf{A}_i\hat{\boldsymbol{\gamma}}^{(l-1)}$.

3.2.2 Variance-Covariance Components

In this section we present two different ways how to perform the update of the variance-covariance matrix $\mathbf{Q}$ from step 2. (b) of the our `bGAMM` algorithm.

Breslow and Clayton (1993) recommend to estimate the variance by maximizing the profile likelihood that is associated with the normal theory model. By replacing $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ with $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\alpha}}$ we maximize

$$\begin{aligned}
l(\mathbf{Q_b}) &= -\frac{1}{2}\log(|\mathbf{V}(\hat{\boldsymbol{\gamma}})|) - \frac{1}{2}\log(|[\mathbf{X}, \boldsymbol{\Phi}]^T\mathbf{V}^{-1}(\hat{\boldsymbol{\gamma}})[\mathbf{X}, \boldsymbol{\Phi}]|) \\
&\quad -\frac{1}{2}(\tilde{\boldsymbol{\eta}}(\hat{\boldsymbol{\gamma}}) - \mathbf{X}\hat{\boldsymbol{\beta}} - \boldsymbol{\Phi}\hat{\boldsymbol{\alpha}})^T\mathbf{V}^{-1}(\hat{\boldsymbol{\gamma}})(\tilde{\boldsymbol{\eta}}(\hat{\boldsymbol{\gamma}}) - \mathbf{X}\hat{\boldsymbol{\beta}} - \boldsymbol{\Phi}\hat{\boldsymbol{\alpha}})
\end{aligned}$$

with respect to $\mathbf{Q_b}$, using the pseudo-observations $\tilde{\boldsymbol{\eta}}(\boldsymbol{\gamma}) = \mathbf{A}\boldsymbol{\gamma} + \mathbf{D}^{-1}(\boldsymbol{\gamma})(\mathbf{y} - \boldsymbol{\mu}(\boldsymbol{\gamma}))$ and with matrices $\mathbf{V}(\boldsymbol{\gamma}) = \mathbf{W}^{-1}(\boldsymbol{\gamma}) + \mathbf{Z}\mathbf{Q_b}\mathbf{Z}^T$, $\mathbf{W}(\boldsymbol{\gamma}) = \mathbf{D}(\boldsymbol{\gamma})\boldsymbol{\Sigma}^{-1}(\boldsymbol{\gamma})\mathbf{D}(\boldsymbol{\gamma})^T$ and with block-diagonal matrices $\mathbf{Q_b} = diag(\mathbf{Q}, \ldots, \mathbf{Q})$, $\mathbf{D} = diag(\mathbf{D}_1, \ldots, \mathbf{D}_n)$ and $\boldsymbol{\Sigma} = diag(\boldsymbol{\Sigma}_1, \ldots \boldsymbol{\Sigma}_n)$. Having calculated $\hat{\boldsymbol{\gamma}}^{(l)}$ in the l-th boosting iteration, we obtain the estimator $\hat{\mathbf{Q}}_{\mathbf{b}}^{(l)}$, which is an approximate REML-type estimate for $\mathbf{Q_b}$.

An alternative estimate, that can be derived as an approximate EM algorithm, uses the posterior mode estimates and posterior curvatures. One derives $(\mathbf{F}^{\text{pen}\,(l)})^{-1}$, the inverse of the penalized pseudo Fisher matrix of the full model corresponding to the l-th iteration using the posterior mode estimates $\hat{\boldsymbol{\gamma}}^{(l)}$ to obtain the posterior curvatures $\hat{\mathbf{V}}_{ii}^{(l)}$. Now compute $\hat{\mathbf{Q}}^{(l)}$ by

$$\hat{\mathbf{Q}}^{(l)} = \frac{1}{n}\sum_{i=1}^{n}(\hat{\mathbf{V}}_{ii}^{(l)} + \hat{\mathbf{b}}_i^{(l)}(\hat{\mathbf{b}}_i^{(l)})^T).$$

In general, the $\mathbf{V}_{ii}$ are derived via the formula

$$\mathbf{V}_{ii} = \mathbf{F}_{ii}^{-1} + \mathbf{F}_{ii}^{-1}\mathbf{F}_{i\tilde{\boldsymbol{\beta}}}(\mathbf{F}_{\tilde{\boldsymbol{\beta}}\tilde{\boldsymbol{\beta}}} - \sum_{i=1}^{n}\mathbf{F}_{\tilde{\boldsymbol{\beta}}i}\mathbf{F}_{ii}^{-1}\mathbf{F}_{i\tilde{\boldsymbol{\beta}}})^{-1}\mathbf{F}_{\tilde{\boldsymbol{\beta}}i}\mathbf{F}_{ii}^{-1},$$

whereas $\tilde{\boldsymbol{\beta}}^T := (\boldsymbol{\beta}, \boldsymbol{\alpha}_{J_1}, \dots, \boldsymbol{\alpha}_{J_s})$ and $J = \{j : \text{sign}(\boldsymbol{\alpha}_j) \neq 0, j = 1, \dots, m\}$ is the index set of "active" covariates, corresponding to the $s := \#J \leq m$ non-zero spline coefficient vectors. $\mathbf{F}_{\tilde{\boldsymbol{\beta}}\tilde{\boldsymbol{\beta}}}, \mathbf{F}_{i\tilde{\boldsymbol{\beta}}}, \mathbf{F}_{ii}$ are the elements of the penalized pseudo Fisher matrix $\mathbf{F}^{\text{pen}}$ of the full model corresponding to the l-th iteration, for details see for example Tutz and Hennevogl (1996) or Fahrmeir and Tutz (2001).

3.2.3 Starting Values, Hat Matrix and Selection in bGAMM

We compute the starting values $\hat{\boldsymbol{\beta}}^{(0)}, \hat{\tilde{\boldsymbol{\alpha}}}^{(0)}, \hat{\mathbf{b}}^{(0)}, \hat{\mathbf{Q}}^{(0)}$ from step 1 of the `bGAMM` algorithm by setting $\hat{\tilde{\boldsymbol{\alpha}}}^{(0)} = \mathbf{0}$ and then fitting a GLMM given by

$$g(\mu_{it}) = \mathbf{x}_{it}^T\boldsymbol{\beta} + \mathbf{z}_{it}^T\mathbf{b}_i, \quad i = 1, \dots, n;\ t = 1, \dots, T_i. \tag{7}$$

This model can be fitted e.g. by using the R-function `glmmPQL` (Wood, 2006) from the `MASS` library (Venables and Ripley, 2002).

To find the appropriate complexity of our model we use the effective degrees of freedom, which corresponds to the trace of the hat matrix (Hastie and Tibshirani, 1990). In the following we derive the hat matrix corresponding to the l-th boosting step for the r-th smooth component (compare Tutz and Groll, 2011). Let $\mathbf{A}_{..r} := [\mathbf{X}, \boldsymbol{\Phi}_r, \mathbf{Z}]$ and $\boldsymbol{\Lambda} = diag(0, \dots, 0, \tilde{\mathbf{K}}, \mathbf{Q}^{-1}, \dots, \mathbf{Q}^{-1})$ be a block diagonal penalty matrix with a diagonal consisting of $p+1$ zeros corresponding to the fixed effects at the beginning, followed by $\tilde{\mathbf{K}}$ corresponding to the r-th smooth effect and finally n times the matrix $\mathbf{Q}^{-1}$. Then the Fisher matrix $\mathbf{F}_r^{\text{pen}\,(l-1)}$ and the score vector $\mathbf{s}_r^{\text{pen}\,(l-1)}$ are given in closed form as

$$\mathbf{F}_r^{\text{pen}\,(l-1)} = \mathbf{A}_{..r}^T\mathbf{W}_l\mathbf{A}_{..r} + \boldsymbol{\Lambda}$$

and

$$\mathbf{s}_r^{\text{pen}\,(l-1)} = \mathbf{A}_{..r}^T \mathbf{W}_l \mathbf{D}_l^{-1}(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(l-1)}) - \boldsymbol{\Lambda}\hat{\boldsymbol{\gamma}}_r^{(l-1)}$$

where $\mathbf{W}_l, \mathbf{D}_l, \boldsymbol{\Sigma}_l$ and $\hat{\boldsymbol{\mu}}^{(l-1)}$ are evaluated at the previous fit $\hat{\boldsymbol{\eta}}^{(l-1)} = \mathbf{A}\hat{\boldsymbol{\gamma}}^{(l-1)}$. For $r = 1, \ldots, p$ the refit in the l-th iteration step by Fisher scoring (6) is given by

$$\begin{aligned}\hat{\boldsymbol{\gamma}}_r^{(l)} &= (\mathbf{F}_r^{\text{pen}\,(l-1)})^{-1}\mathbf{s}_r^{(l-1)} \\ &= \left(\mathbf{A}_{..r}^T \mathbf{W}_l \mathbf{A}_{..r} + \boldsymbol{\Lambda}\right)^{-1} \mathbf{A}_{..r}^T \mathbf{W}_l \mathbf{D}_l^{-1}(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(l-1)}).\end{aligned}$$

We define the predictor corresponding to the r-th refit in the l-th iteration step as

$$\begin{aligned}\hat{\boldsymbol{\eta}}_{..r}^{(l)} &:= \hat{\boldsymbol{\eta}}^{(l-1)} + \mathbf{A}_{..r}\hat{\boldsymbol{\gamma}}_r^{(l)}, \\ \hat{\boldsymbol{\eta}}_{..r}^{(l)} - \hat{\boldsymbol{\eta}}^{(l-1)} &= \mathbf{A}_{..r}\hat{\boldsymbol{\gamma}}_r^{(l)} \\ &= \mathbf{A}_{..r}\left(\mathbf{A}_{..r}^T \mathbf{W}_l \mathbf{A}_{..r} + \boldsymbol{\Lambda}\right)^{-1} \mathbf{A}_{..r}^T \mathbf{W}_l \mathbf{D}_l^{-1}(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(l-1)}).\end{aligned}$$

Taylor approximation of first order $h(\hat{\boldsymbol{\eta}}) \approx h(\boldsymbol{\eta}) + \frac{\partial h(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}^T}(\hat{\boldsymbol{\eta}} - \boldsymbol{\eta})$ yields

$$\begin{aligned}\hat{\boldsymbol{\mu}}_{..r}^{(l)} &\approx \hat{\boldsymbol{\mu}}^{(l-1)} + \mathbf{D}_l(\hat{\boldsymbol{\eta}}_{..r}^{(l)} - \hat{\boldsymbol{\eta}}^{(l-1)}), \\ \hat{\boldsymbol{\eta}}_{..r}^{(l)} - \hat{\boldsymbol{\eta}}^{(l-1)} &\approx \mathbf{D}_l^{-1}(\hat{\boldsymbol{\mu}}_{..r}^{(l)} - \hat{\boldsymbol{\mu}}^{(l-1)}),\end{aligned}$$

and therefore

$$\mathbf{D}_l^{-1}(\hat{\boldsymbol{\mu}}_{..r}^{(l)} - \hat{\boldsymbol{\mu}}^{(l-1)}) \approx \mathbf{A}_{..r}\left(\mathbf{A}_{..r}^T \mathbf{W}_l \mathbf{A}_{..r} + \boldsymbol{\Lambda}\right)^{-1} \mathbf{A}_{..r}^T \mathbf{W}_l \mathbf{D}_l^{-1}(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(l-1)}).$$

Multiplication with $\mathbf{W}_l^{1/2}$ and using $\mathbf{W}^{1/2}\mathbf{D}^{-1} = \boldsymbol{\Sigma}^{-1/2}$ yields

$$\boldsymbol{\Sigma}_l^{-1/2}(\hat{\boldsymbol{\mu}}_{..r}^{(l)} - \hat{\boldsymbol{\mu}}^{(l-1)}) \approx \tilde{\mathbf{H}}_r^{(l)}\boldsymbol{\Sigma}_l^{-1/2}(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(l-1)}),$$

where $\tilde{\mathbf{H}}_r^{(l)} := \mathbf{W}_l^{1/2}\mathbf{A}_{..r}\left(\mathbf{A}_{..r}^T \mathbf{W}_l \mathbf{A}_{..r} + \boldsymbol{\Lambda}\right)^{-1} \mathbf{A}_{..r}^T \mathbf{W}_l^{1/2}$ denotes the usual generalized ridge regression hat-matrix. Defining $\mathbf{M}_r^{(l)} := \boldsymbol{\Sigma}_l^{1/2}\tilde{\mathbf{H}}_r^{(l)}\boldsymbol{\Sigma}_l^{-1/2}$ yields the approximation

$$\begin{aligned}\hat{\boldsymbol{\mu}}_{..r}^{(l)} &\approx \hat{\boldsymbol{\mu}}^{(l-1)} + \mathbf{M}_r^{(l)}(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(l-1)}) \\ &= \hat{\boldsymbol{\mu}}^{(l-1)} + \mathbf{M}_r^{(l)}[(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(l-2)}) - (\hat{\boldsymbol{\mu}}^{(l-1)} - \hat{\boldsymbol{\mu}}^{(l-2)})] \\ &\approx \hat{\boldsymbol{\mu}}^{(l-1)} + \mathbf{M}_r^{(l)}[(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(l-2)}) - \mathbf{M}_{j_{l-1}}^{(l-1)}(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(l-2)})],\end{aligned}$$

whereas $j_{l-1} \in \{1, \ldots, p\}$ denotes the index of the component selected in boosting step $l-1$.

The hat matrix corresponding to the fixed effects model from equation (7) is

$$\mathbf{M}^{(0)} = \mathbf{A}_0(\mathbf{A}_0^T\mathbf{W}_1\mathbf{A}_0 + \mathbf{K}_0)^{-1}\mathbf{A}_0^T\mathbf{W}_1,$$

with $\mathbf{A}_0 := [\mathbf{X}, \mathbf{Z}]$ and block diagonal penalty matrix $\mathbf{K}_0 := diag(0, \ldots, 0, \mathbf{Q}^{-1}, \ldots, \mathbf{Q}^{-1})$ whereas the first $p+1$ zeros correspond to the fixed effects. As the approximation $\hat{\boldsymbol{\mu}}^{(0)} \approx \mathbf{M}^{(0)}\mathbf{y}$ holds, one obtains

$$\begin{aligned} \hat{\boldsymbol{\mu}}^{(1)}_{..r} &\approx \hat{\boldsymbol{\mu}}^{(0)} + \mathbf{M}^{(1)}_r(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(0)}) \\ &\approx \mathbf{M}^{(0)}\mathbf{y} + \mathbf{M}^{(1)}_r(\mathbf{I} - \mathbf{M}^{(0)})\mathbf{y}. \end{aligned}$$

In the following, to indicate that the hat matrices of the former steps have been fixed, let $j_k \in \{1, \ldots, p\}$ denote the index of the component selected in boosting step k. Then we can abbreviate $\mathbf{M}_{j_k} := \mathbf{M}^{(k)}_{j_k}$ for the matrix corresponding to the component that has been selected in the k-th iteration. Further, in a recursive manner, we get

$$\hat{\boldsymbol{\mu}}^{(l)}_{..r} \approx \mathbf{H}^{(l)}_r\mathbf{y},$$

where

$$\begin{aligned} \mathbf{H}^{(l)}_r &= \mathbf{I} - (\mathbf{I} - \mathbf{M}^{(l)}_r)(\mathbf{I} - \mathbf{M}_{j_{l-1}})(\mathbf{I} - \mathbf{M}_{j_{l-2}}) \cdot \ldots \cdot (\mathbf{I} - \mathbf{M}^{(0)}) \\ &= \mathbf{M}^{(l)}_r \prod_{i=0}^{l-1}(\mathbf{I} - \mathbf{M}_{j_i}) + \sum_{k=0}^{l-1} \mathbf{M}_{j_k} \prod_{i=0}^{k-1}(\mathbf{I} - \mathbf{M}_{j_i}) \\ &= \sum_{k=0}^{l} \mathbf{M}_{j_k} \prod_{i=0}^{k-1}(\mathbf{I} - \mathbf{M}_{j_i}), \end{aligned}$$

is the hat matrix corresponding to the l-th boosting step considering the r-th component, whereas $\mathbf{M}_{j_l} := \mathbf{M}^{(l)}_r$ is not fixed yet.

For a given hat matrix $\mathbf{H}$, we can determine the complexity of our model by the following information criteria:

$$AIC = -2\, l(\hat{\boldsymbol{\mu}}) + 2\, \text{trace}\, (\mathbf{H}), \tag{8}$$

$$BIC = -2\, l(\hat{\boldsymbol{\mu}}) + 2\, \text{trace}\, (\mathbf{H})\, log(n), \tag{9}$$

where

$$l(\boldsymbol{\mu}) = \sum_{i=1}^{n} l_i(\hat{\boldsymbol{\mu}}_i) = \sum_{i=1}^{n} \log f(\mathbf{y}_i|\hat{\boldsymbol{\mu}}_i) \tag{10}$$

denotes the non-penalized version of the log-likelihood from equation (5) and $l_i(\hat{\boldsymbol{\mu}}_i)$ the log-likelihood contributions of $(\mathbf{y}_i, \mathbf{X}_i, \mathbf{\Phi}_i, \mathbf{Z}_i)$. Note that the log-likelihood can be written with $\boldsymbol{\mu}$ instead of $\boldsymbol{\delta}$ in the argument, considering the definition of the natural parameter $\theta = \theta(\boldsymbol{\mu})$ in (3) and using $\boldsymbol{\mu} = h(\boldsymbol{\eta})$ and $\boldsymbol{\eta} = \mathbf{A}\boldsymbol{\gamma}$.

For exponential family distributions $\log f(\mathbf{y}_i|\hat{\boldsymbol{\mu}}_i)$ has a well-known form. For example in the case of binary responses, one obtains

$$\log f(\mathbf{y}_i|\hat{\boldsymbol{\mu}}_i) = \sum_{t=1}^{T_i} y_{it} \log \hat{\mu}_{it} + (1 - y_{it}) \log (1 - \hat{\mu}_{it}),$$

whereas in the case of Poisson responses, one has

$$\log f(\mathbf{y}_i|\hat{\boldsymbol{\mu}}_i) = \sum_{t=1}^{T_i} y_{it} \log \hat{\mu}_{it} - \hat{\mu}_{it}.$$

Based on (10), the information criteria (8) and (9) used in the l-th boosting step, considering the r-th component, have the form $AIC_r^{(l)} = -2\, l(\hat{\boldsymbol{\mu}}_{..r}^{(l)}) + 2\,\text{trace}\,(\mathbf{H}_r^{(l)})$, $BIC_r^{(l)} = -2\, l(\hat{\boldsymbol{\mu}}_{..r}^{(l)}) + 2\,\text{trace}\,(\mathbf{H}_r^{(l)})\, log(n)$ with $l(\hat{\boldsymbol{\mu}}_{..r}^{(l)}) = \sum_{i=1}^{n} \log f(\mathbf{y}_i|\hat{\boldsymbol{\mu}}_{i.r}^{(l)})$.

3.2.4 Stopping Criterion

In the l-th step one selects from $r \in \{1, \ldots, p\}$ the component j_l that minimizes $AIC_r^{(l)}$ or $BIC_r^{(l)}$ and obtains $AIC^{(l)} := AIC_{j_l}^{(l)}$. We choose a number l_{max} of maximal boosting steps, e.g. $l_{max} = 1000$, and stop the algorithm at iteration l_{max}. Then we select from $\mathcal{L} := \{1, 2, \ldots, l_{max}\}$ the component l_{opt}, where $AIC^{(l)}$ or $BIC^{(l)}$ is smallest, that is

$$\begin{aligned} l_{opt} &= \arg\min_{l \in \mathcal{L}} AIC^{(l)}, \\ l_{opt} &= \arg\min_{l \in \mathcal{L}} BIC^{(l)}. \end{aligned}$$

Finally, we obtain the parameter estimates $\hat{\boldsymbol{\gamma}}^{(l_{opt})}, \hat{\mathbf{Q}}^{(l_{opt})}$ and the corresponding fit $\hat{\boldsymbol{\mu}}^{(l_{opt})}$.

4 Simulation study

In the following we present two simulation studies to investigate the performance of the `bGAMM` algorithm, one with Bernoulli data and one with Poisson data. We also compare the algorithm to alternative approaches. The optimal smoothing parameter λ chosen as the value λ_{opt} which leads to the smallest AIC or BIC from (8) and (9), which are computed on a fine grid. Also general cross validation could be used, with the negative effect of expanding computational time.

4.1 Bernoulli Data with Logit-Link

The underlying model is the random intercept additive Bernoulli model

$$\begin{aligned}\eta_{it} &= \sum_{j=1}^{p} f_j(u_{itj}) + b_i, \quad i = 1, \dots, 40, \quad t = 1, \dots, 10 \\ E[y_{it}] &= \frac{\exp(\eta_{it})}{1 + \exp(\eta_{it})} := \pi_{it} \qquad y_{it} \sim \mathrm{B}(1, \pi_{it})\end{aligned}$$

with smooth effects given by

$$\begin{aligned}
&f_1(u) = 6\sin(u) && \text{with} \quad u \in [-\pi, \pi], \\
&f_2(u) = 6\cos(u) && \text{with} \quad u \in [-\pi, 2\pi], \\
&f_3(u) = u^2 && \text{with} \quad u \in [-\pi, \pi], \\
&f_4(u) = 0.4u^3 && \text{with} \quad u \in [-\pi, \pi], \\
&f_5(u) = -u^2 && \text{with} \quad u \in [-\pi, \pi], \\
&f_j(u) = 0 && \text{with} \quad u \in [-\pi, \pi], \quad \text{for} \quad j = 6, \dots, 50.
\end{aligned}$$

We choose the different settings $p = 5, 10, 15, 20, 50$. For $j = 1, \dots, 50$ the vectors $\mathbf{u}_{it}^T = (u_{it1}, \dots, u_{it50})$ have been drawn independently with components following a uniform distribution within the specified interval. The number of observations is fixed as $n = 40, T_i := T = 10, \forall i = 1, \dots, n$. The random effects are specified by $b_i \sim N(0, \sigma_b^2)$ with three different scenarios $\sigma_b \in \{0.4, 0.8, 1.6\}$.

The performance of estimators is evaluated separately for the structural components and the variance. We compare the results of our `bGAMM` algorithm with the results that one achieves by using the **R** function `gamm` recommended in Wood (2006), which is providing a penalized quasi-likelihood approach for the generalized additive mixed model. It is supplied with the `mgcv` library.

By averaging across 100 data sets we consider mean squared errors for the smooth components and σ_b given by

$$\mathrm{mse}_f := \sum_{t=1}^{N} \sum_{j=1}^{p} (f_j(v_{tj}) - \hat{f}_j(v_{tj}))^2, \qquad \mathrm{mse}_{\sigma_b} := ||\sigma_b - \hat{\sigma}_b||^2,$$

where $v_{tj}, t = 1, \dots, N$ denote fine and evenly spaced grids on the different predictor spaces for $j = 1, \dots, p$.

Additional information on the stability of the algorithms was collected in *notconv* (n.c.), which indicates the sum over the datasets, where numerical problems occurred during estimation. Moreover, *falseneg* (f.n.) reflects the mean over all 100 simulations of the number of functions $f_j, j = 1,2,3,4,5$, that were not selected while *falsepos* (f.p.) reflects the mean over the number of functions $f_j, j = 6,\dots,p$, that were wrongly selected. As the gamm function is not able to perform variable selection it always estimates all functions $f_j, j = 1,\dots,p$.

The results of all quantities for different scenarios of σ_b and for varying number of noise variables can be found in Table 1. It should be noted that, in order to obtain a better comparability, the quantities mse_f and mse_{σ_b} are only averaged across those cases, where the gamm function yields reasonable results, while the quantities *notconv*, *falseneg* and *falsepos* are averaged across all 100 simulations. Also the following boxplots include only those cases, where no numerical problems occurred for the gamm function, see Figures 1 and 2.. For completeness we give the results of the bGAMM algorithm averaged over all 100 simulations in the Table 2.

		gamm			bGAMM (EM)				bGAMM (REML)			
σ_b	p	mse_f	mse_{σ_b}	n.c.	mse_f	mse_{σ_b}	f.p.	f.n.	mse_f	mse_{σ_b}	f.p.	f.n.
0.4	5	54809.28	0.188	64	34017.24	0.884	0	0	41002.12	0.223	0	0.05
0.4	10	54826.50	0.112	85	34486.28	0.654	0	0	41220.06	0.122	0	0.05
0.4	15	51605.63	0.151	93	34465.05	1.442	0	0	40695.23	0.322	0	0.05
0.4	20	54706.54	0.149	96	36361.86	0.160	0	0	44823.88	0.104	0	0.05
0.4	50	-	-	100	33648.53	1.359	0	0	41606.17	0.282	0	0.05
0.8	5	52641.67	0.470	55	34058.04	1.432	0	0	44332.94	0.474	0	0.08
0.8	10	53384.37	0.462	88	36665.52	1.257	0	0	43772.60	0.407	0	0.08
0.8	15	53842.01	0.272	95	32970.83	1.638	0	0	38868.70	0.445	0	0.08
0.8	20	55771.45	0.320	96	41776.10	1.254	0	0	41876.68	0.526	0	0.08
0.8	50	-	-	100	34581.50	1.584	0	0	42755.58	0.545	0	0.08
1.6	5	53909.80	1.683	58	32268.83	1.689	0	0	39505.94	0.828	0	0.36
1.6	10	54376.56	2.160	86	34677.94	1.646	0	0	40186.27	0.806	0	0.36
1.6	15	53100.51	2.110	93	32380.74	1.410	0	0	40496.85	0.953	0	0.36
1.6	20	-	-	100	32844.44	1.891	0	0	40306.13	0.927	0	0.36
1.6	50	-	-	100	32884.22	1.897	0	0	40449.15	0.935	0	0.36

Table 1: Generalized additive mixed model with gamm and boosting (bGAMM) on Bernoulli data

		bGAMM (EM)		bGAMM (REML)	
σ_b	p	mse_f	mse_{σ_b}	mse_f	mse_{σ_b}
0.4	5	33563.44	1.382	41671.53	0.280
0.4	10	33563.44	1.382	41671.53	0.280
0.4	15	33563.44	1.382	41671.53	0.280
0.4	20	33530.58	1.395	41624.79	0.282
0.4	50	33648.53	1.359	41606.17	0.282
0.8	5	34581.50	1.584	42755.58	0.545
0.8	10	34581.50	1.584	42755.58	0.545
0.8	15	34581.50	1.584	42755.58	0.545
0.8	20	34581.50	1.584	42755.58	0.545
0.8	50	34581.50	1.584	42755.58	0.545
1.6	5	32844.44	1.891	40306.13	0.927
1.6	10	32844.44	1.891	40306.13	0.927
1.6	15	32844.44	1.891	40306.13	0.927
1.6	20	32844.44	1.891	40306.13	0.927
1.6	50	32884.22	1.897	40449.15	0.935

Table 2: Generalized additive mixed model with boosting (bGAMM) on bernoulli data averaged over all 100 simulations

It is seen that the gamm function is very unstable when the number of predictors grows and for all numbers of predictors estimates are hard to find. The boosting algorithms are much more stable and mse_f is even better if evaluated for all simulations instead of the subset favored by gamm. So for binary data boosting procedures dominate gamm in terms of mse_f. In terms of mse$_{\sigma_b}$ gamm dominates but the REML version of boosting comes close.

Exemplarily for the case $p = 5$ and $\sigma_b = 0.4$ the estimates of the smooth functions are presented in Figure 3 for those 36 simulations, where the gamm function estimated without numerical problems. It becomes obvious that the two boosting approaches can reproduce the true feature of the influence functions much more precisely, with the EM version leading to slightly better results.

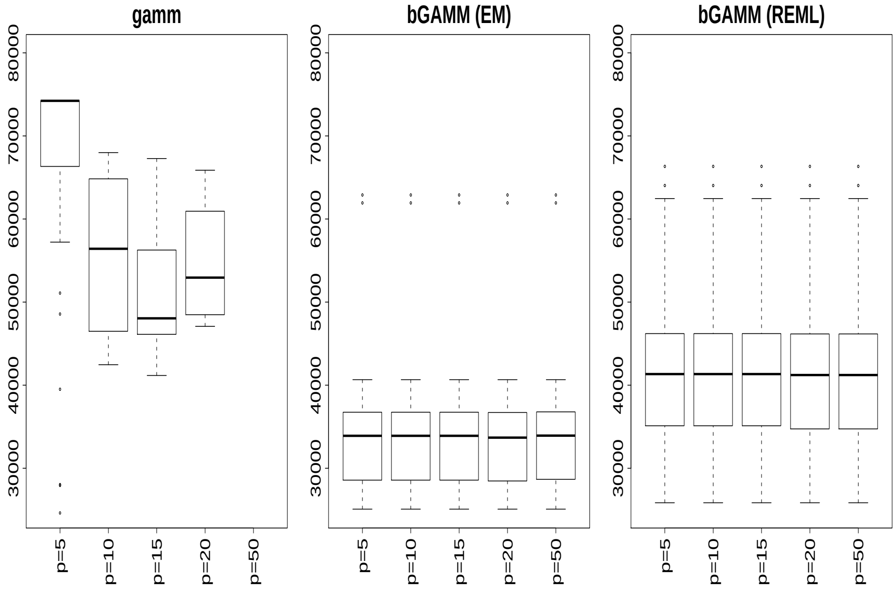

Figure 1: Boxplots of mse$_f$ for gamm* (left), bGAMM EM(middle) and bGAMM REML (right) for $p = 5, 10, 15, 20, 50$ (* only those cases, where gamm did converge)

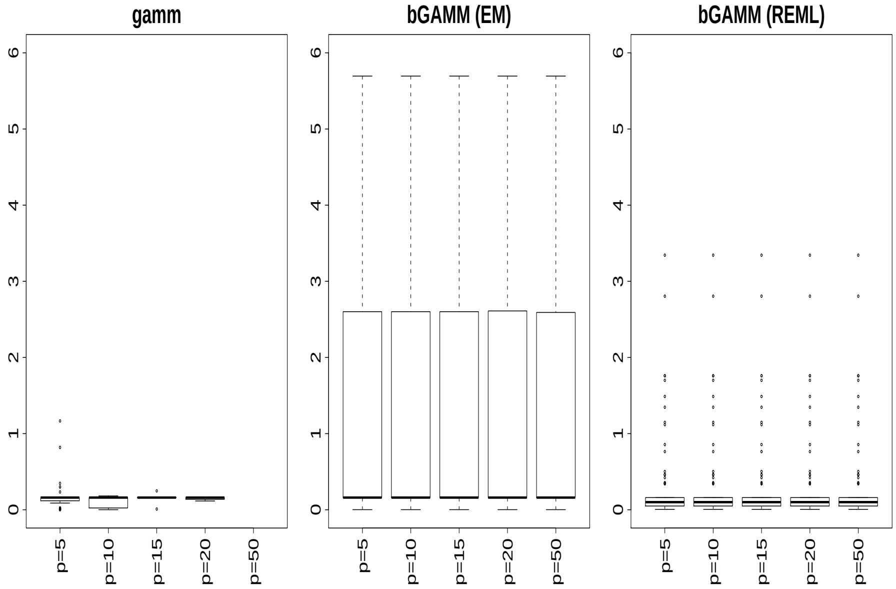

Figure 2: Boxplots of mse$_\sigma$ for the gamm model (left), the bGAMM EM model (middle) and the bGAMM REML model (right) for $p = 5, 10, 15, 20, 50$

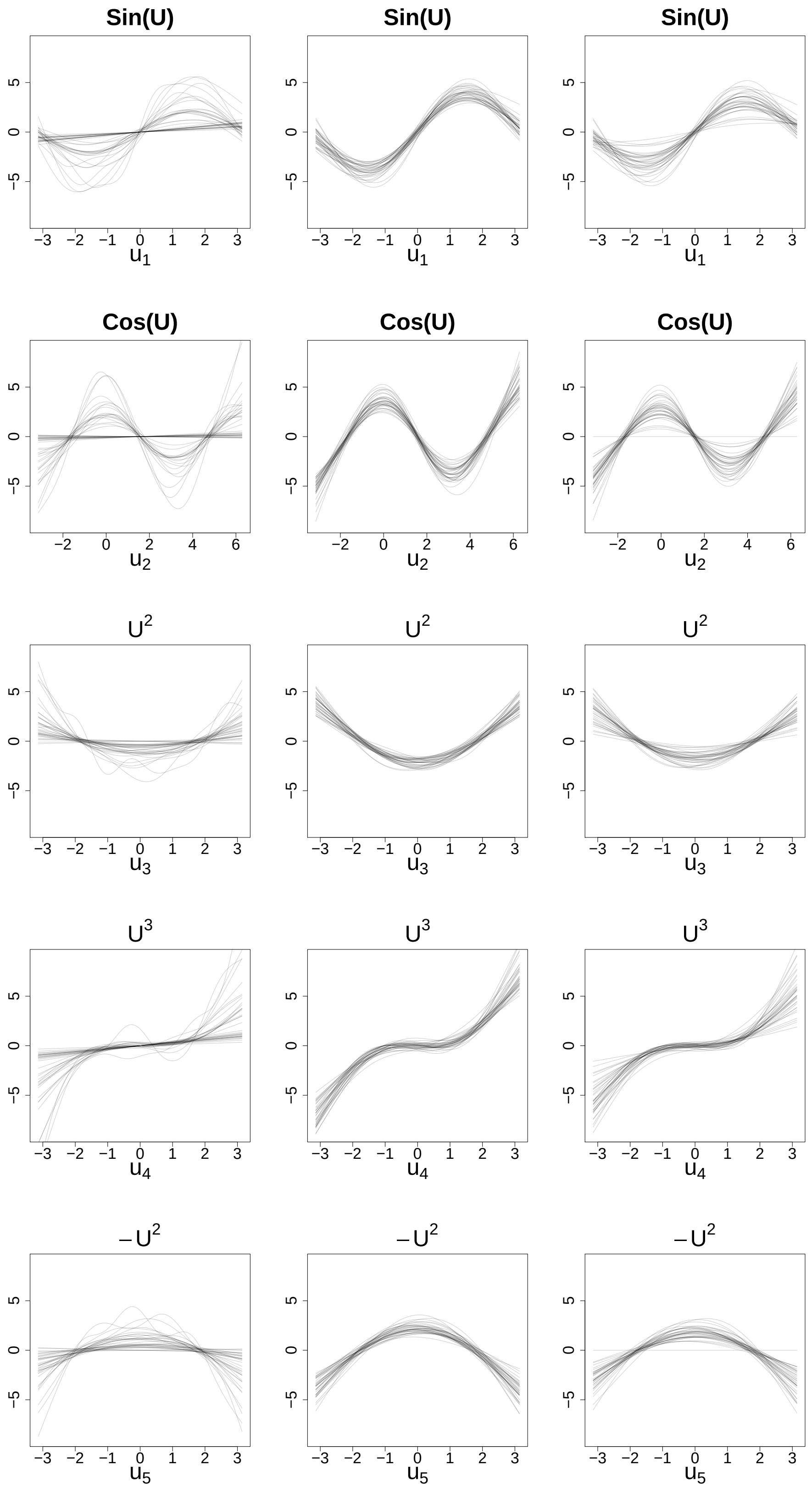

Figure 3: Smooth functions computed with the `gamm` model (left), the `bGAMM` EM model (middle) and the `bGAMM` REML model (right) for $p = 5, \sigma_b = 0.4$

4.2 Poisson Data with Log-Link

The underlying model is the random intercept additive Poisson model

$$\begin{aligned} \eta_{it} &= \sum_{j=1}^{p} f_j(u_{itj}) + b_i, \quad i = 1, \dots, 40, \quad t = 1, \dots, 10, \\ E[y_{it}] &= \exp(\eta_{it}) := \lambda_{it} \qquad y_{it} \sim \text{Pois}(\lambda_{it}) \end{aligned}$$

with smooth effects given by

$$\begin{aligned} &f_1(u) = \sin(u) && \text{with} \quad u \in [-\pi, \pi], \\ &f_2(u) = \cos(u) && \text{with} \quad u \in [-\pi, 3\pi], \\ &f_3(u) = u^2 && \text{with} \quad u \in [-1, 1], \\ &f_4(u) = u^3 && \text{with} \quad u \in [-1, 1], \\ &f_5(u) = -u^2 && \text{with} \quad u \in [-1, 1], \\ &f_j(u) = 0 && \text{with} \quad u \in [-\pi, \pi], \quad \text{for} \quad j = 6, \dots, 50. \end{aligned}$$

Again we choose the different settings $p = 5, 10, 15, 20, 50$. For $j = 1, \dots, 50$ the vectors $\mathbf{u}_{it}^T = (u_{it1}, \dots, u_{it50})$ have been drawn independently with components following a uniform distribution within the specified interval. The number of observations is fixed as $n = 40, T_i := T = 10, \forall i = 1, \dots, n$. The random effects are specified by $b_i \sim N(0, \sigma_b^2)$ with same three scenarios as in the Poisson case.

We also use the same goodness-of-fit criteria as for the Bernoulli case and compare the results of our `bGAMM` algorithm with the results achieved by using the `gamm` function (Wood, 2006), see Table 3.

		`gamm`			`bGAMM` (EM)				`bGAMM` (REML)			
σ_b	p	mse$_f$	mse$_{\sigma_b}$	n.c.	mse$_f$	mse$_{\sigma_b}$	f.p.	f.n.	mse$_f$	mse$_{\sigma_b}$	f.p.	f.n.
0.4	5	21.220	0.004	0	28.617	0.050	0	0	28.598	0.005	0	0
0.4	10	26.059	0.004	39	28.158	0.033	0.01	0	28.158	0.005	0.02	0
0.4	15	27.819	0.003	89	23.927	0.100	0.04	0	23.968	0.007	0.04	0
0.4	20	33.050	0.001	95	28.259	0.027	0.04	0	28.278	0.004	0.04	0
0.4	50	79.245	0.006	89	32.522	0.029	0.09	0	30.899	0.005	0.08	0
0.8	5	19.398	0.010	0	24.293	0.122	0	0	24.310	0.009	0	0
0.8	10	21.859	0.011	48	23.827	0.097	0.01	0	23.836	0.007	0.01	0
0.8	15	36.088	0.001	96	26.524	0.151	0.01	0	26.560	0.002	0.01	0
0.8	20	36.311	0.007	95	25.704	0.015	0.02	0	25.652	0.007	0.02	0
0.8	50	75.365	0.015	95	25.258	0.177	0.06	0	23.526	0.009	0.06	0
1.6	5	11.823	0.038	2	15.301	1.224	0	0	15.283	0.042	0	0
1.6	10	14.869	0.036	57	16.229	1.287	0.14	0	16.283	0.040	0.14	0
1.6	15	14.098	0.070	99	4.478	7.212	0.22	0	4.481	0.127	0.23	0
1.6	20	-	-	100	16.762	1.139	0.28	0	16.818	0.042	0.28	0
1.6	50	2043.006	2.543	99	34.449	0.963	0.46	0	27.338	0.044	0.47	0

Table 3: Generalized additive mixed model with `gamm` and boosting (`bGAMM`) on Poisson data

For completeness we give the results of the `bGAMM` algorithm averaged over all 100 simulations in the Table 4. For Poisson data it is seen again that the `gamm` function is very unstable when the number of predictors grows. Already for ten predictors estimates are hard to find. The boosting algorithms are much more stable and mse_f is again better if evaluated for all simulations instead of the subset favored by `gamm`.

		`bGAMM` (EM)		`bGAMM` (REML)	
σ_b	p	mse_f	mse_{σ_b}	mse_f	mse_{σ_b}
0.4	5	28.617	0.050	28.598	0.005
0.4	10	28.597	0.050	28.732	0.005
0.4	15	28.888	0.050	28.927	0.005
0.4	20	28.863	0.050	28.869	0.005
0.4	50	29.391	0.050	28.848	0.005
0.8	5	24.293	0.122	24.310	0.009
0.8	10	24.346	0.121	24.364	0.009
0.8	15	24.360	0.121	24.377	0.009
0.8	20	24.465	0.118	24.456	0.009
0.8	50	24.899	0.113	24.464	0.009
1.6	5	15.301	1.219	15.287	0.042
1.6	10	15.666	1.184	15.688	0.042
1.6	15	16.399	1.163	16.449	0.042
1.6	20	16.762	1.139	16.818	0.042
1.6	50	18.140	0.963	17.075	0.044

Table 4: Generalized additive mixed model with boosting (`bGAMM`) on Poisson data averaged over all 100 simulations

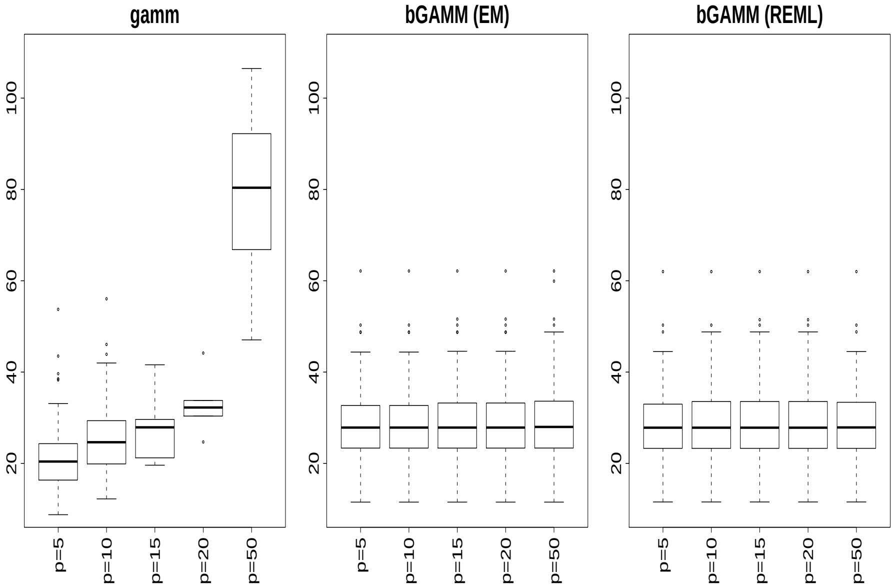

Figure 4: Boxplots of mse_f for the `gamm` model (left), the `bGAMM` EM model (middle) and the `bGAMM` REML model (right) for $p = 5, 10, 15, 20, 50$

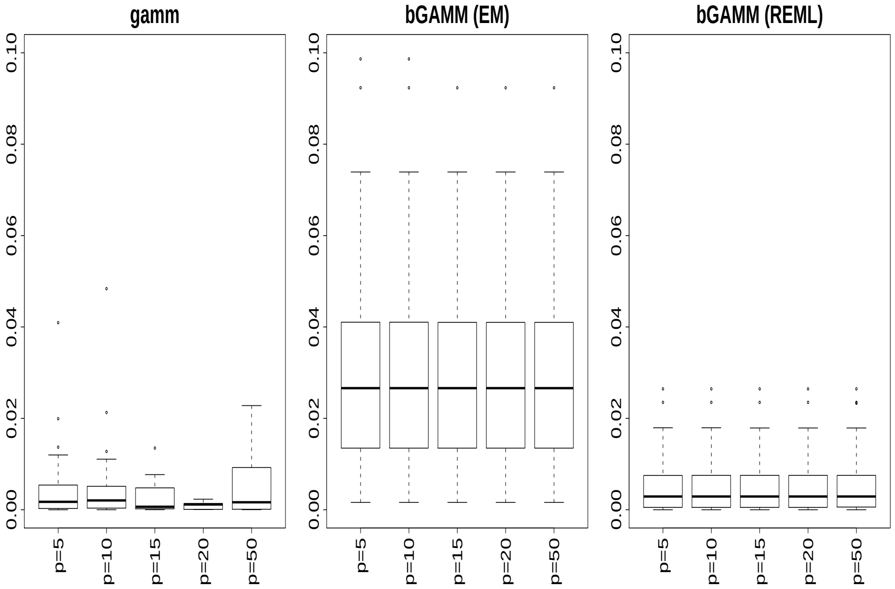

Figure 5: Boxplots of mse_σ for the `gamm` model (left), the `bGAMM` EM model (middle) and the `bGAMM` REML model (right) for $p = 5, 10, 15, 20, 50$

5 Applications to Real Data

In the following sections we will apply our boosting method on different real data sets and compare the results of our method with other approaches. The identification of the optimal smoothing parameter λ has been carried out using 5-fold cross validation.

5.1 AIDs study

The data were collected within the Multicenter AIDS Cohort Study (MACS), which has followed nearly 5000 gay or bisexual men from Baltimore, Pittsburgh, Chicago and Los Angeles since 1984 (see Kaslow et al., 1987; Zeger and Diggle, 1994). The study includes 1809 men who were infected with HIV when the study began and another 371 men who were seronegative at entry and seroconverted during the followup. In our application 369 seroconverters with 2376 measurements over time are used. The interesting response variable is the number of CD4 cells by which progression of disease may be assessed. Covariates include years since seroconversion, packs of cigarettes a day, recreational drug use (yes/no), number of sexual partners, age and a mental illness score (cesd). The data has been already examined in Tutz and Reithinger (2007).

Since the forms of the effects are not known, time since seroconversion, age and the mental illness score may be considered as unspecified additive effects. We consider the semi-parametric mixed model with linear predictor $g(\mu_{it}) = \eta_{it} = \eta_{it}^{par} + \eta_{it}^{add} + b_i$, where μ_{it} denotes the expected

CD4 number of cells for subject i on measurement t (taken at irregular time intervals). The parametric and nonparametric terms are

$$\eta_{it}^{\text{par}} = \beta_0 + \text{drugs}_{it}\beta_1 + \text{partners}_{it}\beta_2 + \text{packs}_{it}\beta_3, \qquad \eta_{it}^{\text{add}} = \alpha_1(\text{time}_{it}) + \alpha_2(\text{age}_{it}) + \alpha_3(\text{cesd}_{it}).$$

We fit an overdispersed Poisson model with natural link. The overdispersion parameter Φ is estimated by use of Pearson residuals $\hat{r}_{it} = (y_{it} - \hat{\mu}_{it})/(v(\hat{\mu}_{it}))^{\frac{1}{2}}$ as

$$\hat{\Phi} = \frac{1}{N - \text{df}} \sum_{i=1}^{n} \sum_{t=1}^{T_i} \hat{r}_{it}^2, \qquad N = \sum_{i=1}^{n} T_i, \tag{11}$$

where the degrees of freedom (df) correspond to the trace of the hat-matrix. The results for the estimation of fixed effects, overdispersion parameter $\hat{\Phi}$ and $\hat{\sigma}_b$ for the `gamm` function (Wood, 2006) and for the `bGAMM` algorithm are given in Table 5.

	`gamm`		`bGAMM` (EM)	`bGAMM` (REML)
Intercept	6.485	(0.026)	6.460	6.460
Drugs	0.034	(0.023)	0.009	0.009
Partners	0.003	(0.003)	0.006	0.006
Packs of Cigarettes	0.040	(0.009)	0.005	0.005
$\hat{\sigma}_b$	0.299		0.345	0.346
$\hat{\Phi}$	69.929		69.473	69.473

Table 5: Estimates for the AIDS Cohort Study MACS with `gamm` function (standard deviations in brackets) and `bGAMM` algorithm

The main interest is in the typical time course of CD4 cell decay and the variability across subjects (see also Zeger and Diggle, 1994). Figure 6 shows the data together with an estimated overall smooth effect of time on CD4 cell decay derived by the `gamm` function. In Figure 7 the smooth effects of time, the mental illness score and age are given for both `gamm` function and `bGAMM` algorithm. It is seen that there is a decease in CD4 cells with time and with higher values of the mental illness score. The `gamm` function estimates a very slight increase for age, while for the `bGAMM` algorithm age is not selected and therefore has no effect at all.

5.2 The German Bundesliga

In the study the effect of team specific influence variables on the sportive success of the 18 soccer clubs of Germany's first soccer division, the Bundesliga, has been investigated for the last three seasons 2007/2008 to 2009/2010. The response variable is the number of points, on which the league's form table is based. Each team gets three points for wins, one point for every draw and no points for defeats. A brief description of the team specific covariates in the

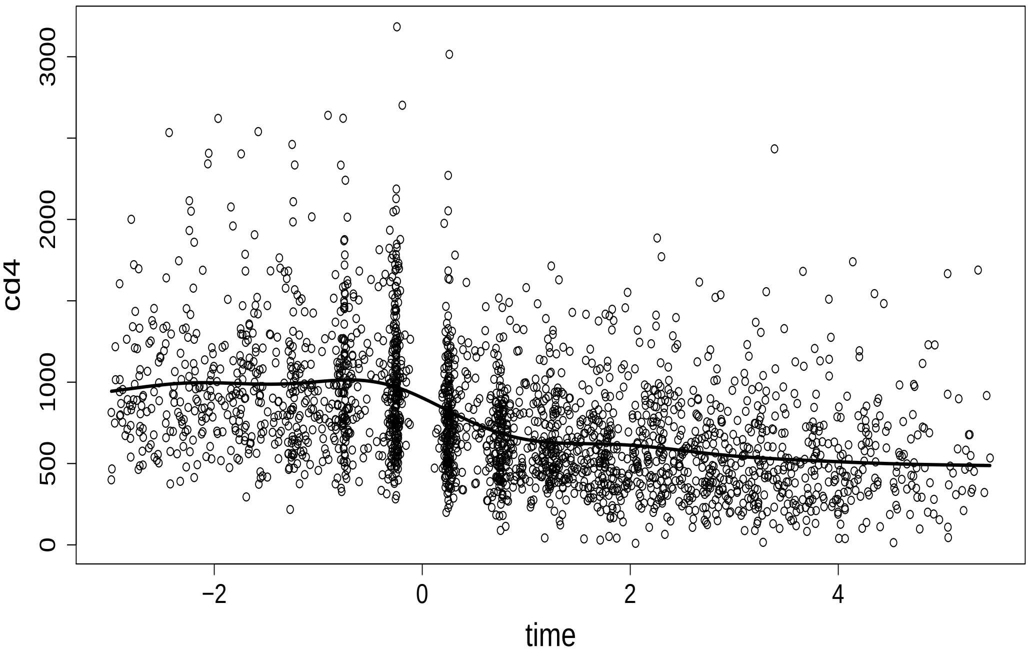

Figure 6: Data from Multicenter AIDS Cohort Study (MACS) and smoothed time effect

data can be found in Table 6.

Covariate	Description
ball possession	average percentage of ball possession per game
tackle	average percentage of tackles won per game
unfairness	average number of unfairness points per game (1 point for yellow card, 3 points for second yellow card, 5 points for red card)
transfer spendings	money spent for new players during a season (in Euro)
transfer receipts	money earned through player transfers during a season (in Euro)
attendance	average attendance during a season
sold out	number of ticket sold outs during a season

Table 6: Description of covariates for the German Bundesliga data

Except for the variables“ball possession" and “tackles", which were treated as parametric terms, for all other variables unspecified additive effects were considered. Due to the very different ranges of values covariates have been standardized. The corresponding semi-parametric mixed model has the form

$$g(\mu_{it}) = \eta_{it}^{\text{par}} + \eta_{it}^{\text{add}} + b_i,$$

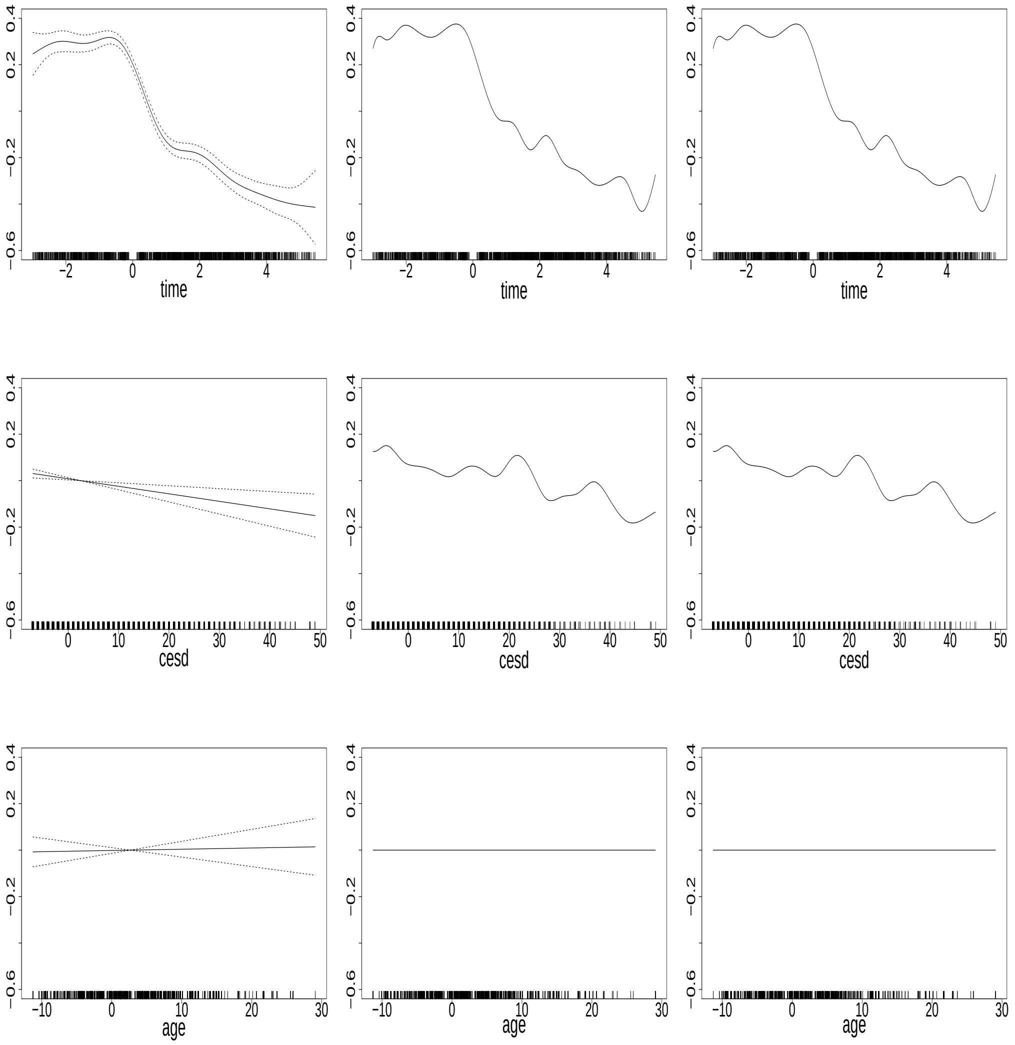

Figure 7: Estimated smooth effect of time, age and cesd computed with the `gamm` model (left), the `bGAMM` EM model (middle) and the `bGAMM` REML model (right) for CD4 data

where μ_{it} denotes the expected number of points for soccer team i in season t. The parametric and nonparametric terms are

$$\begin{aligned}\eta_{it}^{\text{par}} &= \beta_0 + \text{ball possession}_{it}\beta_1 + \text{tackles}_{it}\beta_2\\ \eta_{it}^{\text{add}} &= \alpha_1(\text{transfer spending}_{it}) + \alpha_2(\text{transfer receipts}_{it}) + \alpha_3(\text{unfairness}_{it})\\ &\quad +\alpha_4(\text{attendance}_{it}) + \alpha_5(\text{sold out}_{it}).\end{aligned}$$

Again we fit an overdispersed Poisson model with natural link while the overdispersion parameter Φ is estimated using (11).

The results for the estimation of fixed effects, overdispersion parameter $\hat{\Phi}$ and $\hat{\sigma}_b$ for the gamm function and for the bGAMM algorithm are given in Table 7. Both boosting functions estimate dispersion parameters not far away from one, so that the Poisson model seems adequate. The gamm function provides a very low standard deviation ($\hat{\sigma}_b$=0.000014) of the random intercepts, while the bGAMM models lead to results that support the application of a random effects model, indicating that each soccer team has an individual bases level of points.

	gamm		bGAMM (EM)	bGAMM (REML)
intercept	3.816	(0.025)	4.023	4.027
ball possession	0.018	(0.041)	-0.148	-0.157
tackles	0.005	(0.039)	-0.053	-0.056
$\hat{\sigma}_b$	0.000		0.349	0.247
$\hat{\Phi}$	1.4114		1.039	1.065

Table 7: Estimates for the German Bundesliga data with gamm function (standard deviations in brackets) and bGAMM algorithm

In Figure 8 the five smooth effects are presented. It becomes obvious, that all three approaches estimate similar functions, but the two boosting approaches exclude the variable “transfer receipts” from the model. Furthermore the smooth effect of the variable “transfer spendings” as well as the strongly positive effect of the variable “attendance” on the number of points are remarkable.

6 Concluding Remarks

Variable selection methods have been proposed that allow to extract the relevant predictors in generalized additive mixed models. The methods are shown to work in high-dimensional settings and turn out to be very stable. Performance suffers hardly when the number of noise variables grows.

Acknowledgement

The authors are grateful to Jasmin Abedieh for providing the German Bundesliga data, which were part of her bachelor thesis.

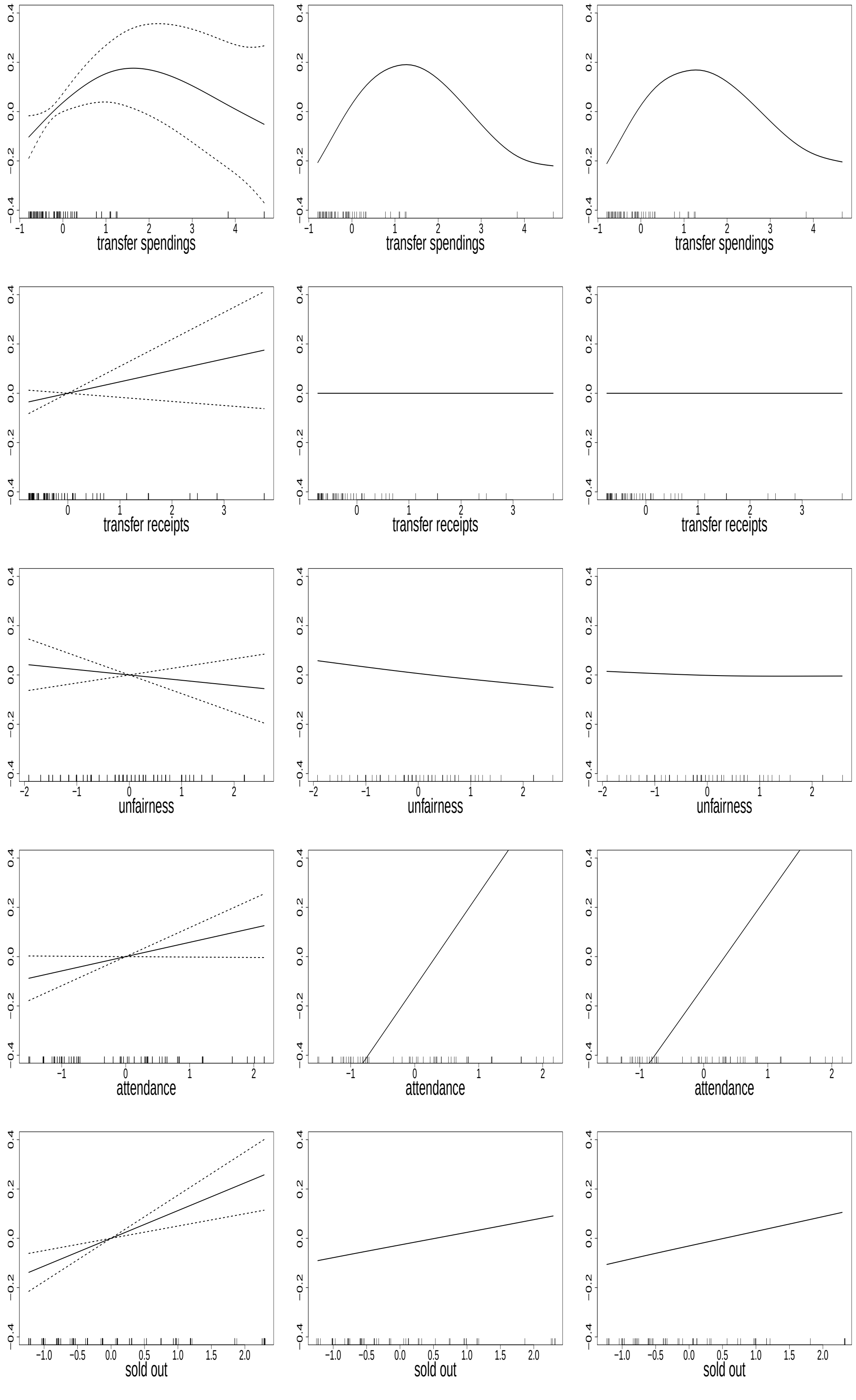

Figure 8: Estimated smooth effects computed with the `gamm` model (left), the `bGAMM` EM model (middle) and the `bGAMM` REML model (right) for the German Bundesliga data

Appendix

A Reparametrization of Penalized B-Splines

Suppose a function f can be represented by a k B-spline basis with functions $B_i(x; d)$ of degree d,

$$f(x) = \sum_{i=1}^{k} \alpha_i B_i(x; d), \tag{12}$$

where α_i are unknown weight parameters. Let

$$f(\mathbf{x}_i) = \mathbf{B}\boldsymbol{\alpha}, \qquad \text{where} \quad \mathbf{B} = \begin{bmatrix} B_1(x_{i1}; d) & \dots & B_k(x_{i1}; d) \\ \vdots & & \vdots \\ B_1(x_{in}; d) & \dots & B_k(x_{in}; d) \end{bmatrix} \tag{13}$$

be the matrix of evaluated basis functions called *B-spline design matrix.* To control the roughness or "wiggliness" of the estimated function in (12) a penalty term is added to the log-likelihood, e.g. the common penalty $J(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha}$. We choose $\mathbf{K} = (\boldsymbol{\Delta}^d)^T \boldsymbol{\Delta}^d$, where $\boldsymbol{\Delta}^d$ denotes the difference operator matrix of degree d, penalizing the differences between neighboring coefficients α_i in order to avoid sudden "jumps" in the estimated function; see for example Whittaker (1923), Eilers (1995) or Eilers and Marx (1996) for the difference penalty. Fahrmeir et al. (2004) suggested a decomposition of the P-spline coefficients into an unpenalized part and a penalized part:

$$\boldsymbol{\alpha} = \mathbf{T}\boldsymbol{\alpha}_0 + \mathbf{P}\boldsymbol{\alpha}_p,$$

where $\boldsymbol{\alpha}_0$ represents the unpenalized part and $\boldsymbol{\alpha}_p$ the penalized part of the spline coefficient vector. For the construction of the matrices $\mathbf{T}$ and $\mathbf{P}$ one uses that the penalty matrix $\mathbf{K}$ can be decomposed into $\mathbf{K} = (\boldsymbol{\Delta}^d)^T \boldsymbol{\Delta}^d$, where $\boldsymbol{\Delta}^d$ has full row rank $(k - d)$. Then the matrix $\mathbf{P}$ is given by

$$\mathbf{P} = \left(\boldsymbol{\Delta}^d (\boldsymbol{\Delta}^d)^T\right)^{-1} (\boldsymbol{\Delta}^d)^T.$$

According to Green (1987) the requirements $\boldsymbol{\Delta}^d \mathbf{T} = 0$ and $\mathbf{T}\boldsymbol{\Delta}^d = 0$ have to hold and the matrix $[\boldsymbol{\Delta}^d, \mathbf{T}]$ has to be nonsingular. As a consequence, $\mathbf{T}$ is a $(k \times d)$ matrix representing a basis of the nullspace of $\mathbf{K}$. For the difference penalty of degree d the basis is straightforward, consisting of all monomials up to degree $d - 1$ defined by the knots of the B-spline. With the B-spline design matrix from equation (13) one obtains.

$$\mathbf{B}\boldsymbol{\alpha} = \mathbf{B}(\mathbf{T}\boldsymbol{\alpha}_0 + \mathbf{P}\boldsymbol{\alpha}_p) = \mathbf{X}_u \boldsymbol{\alpha}_0 + \mathbf{Z}_p \boldsymbol{\alpha}_p,$$

and the penalty term simplifies to

$$\begin{aligned} J(\boldsymbol{\alpha}) &= \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} = \boldsymbol{\alpha}^T (\boldsymbol{\Delta}^d)^T \boldsymbol{\Delta}^d \boldsymbol{\alpha} \\ &= (\mathbf{T}\boldsymbol{\alpha}_0 + \mathbf{P}\boldsymbol{\alpha}_p)^T (\boldsymbol{\Delta}^d)^T \boldsymbol{\Delta}^d (\mathbf{T}\boldsymbol{\alpha}_0 + \mathbf{P}\boldsymbol{\alpha}_p) \\ &= \boldsymbol{\alpha}_p^T \mathbf{P}^T (\boldsymbol{\Delta}^d)^T \boldsymbol{\Delta}^d \mathbf{P} \boldsymbol{\alpha}_p = \boldsymbol{\alpha}_p^T \boldsymbol{\alpha}_p. \end{aligned}$$

Thus, all in all, with $\tilde{\boldsymbol{\alpha}}^T := (\boldsymbol{\alpha}_0^T, \boldsymbol{\alpha}_p^T)$, $\boldsymbol{\Phi} := [\mathbf{X}_u, \mathbf{Z}_p]$ and $\tilde{\mathbf{K}} := \text{Diag}(0, \ldots, 0, 1, \ldots, 1)$ being a diagonal matrix with zeros corresponding to $\boldsymbol{\alpha}_0$ and ones corresponding to $\boldsymbol{\alpha}_p$, one obtains $J(\boldsymbol{\alpha}) = \tilde{\boldsymbol{\alpha}}^T \tilde{\mathbf{K}} \tilde{\boldsymbol{\alpha}}$, and $\mathbf{B}\boldsymbol{\alpha} = \boldsymbol{\Phi}\tilde{\boldsymbol{\alpha}}$.

B Reparametrization in semiparametric models

In this section a small additional step to the reparametrization from Appendix A is explained, that becomes necessary if the model is semiparametric, with the parametric term containing the intercept. Notice that the $(k \times d)$-matrix $\mathbf{X}_u$ from Appendix A has the general form

$$\mathbf{X}_u = \begin{bmatrix} 1 & \xi_{1,1} & \ldots & \xi_{1,d-1} \\ 1 & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ 1 & \xi_{k,1} & \ldots & \xi_{k,d-1} \end{bmatrix},$$

where the first column with ones refers to the level of the function in (12). If the parametric term of the model already contains the intercept, the estimated function $f(x)$ must be centered around zero in order to avoid identification problems. This can be achieved by dropping the first column of the matrix $\mathbf{X}_u$. Then the dimensions of $\mathbf{X}_u$ and $\boldsymbol{\Phi}$ decrease to $(n \times (d-1))$ and to $(n \times (k-1))$, respectively. As a consequence the first value of $\boldsymbol{\alpha}_0$, representing the level of the estimated function, doesn't have to be estimated anymore and also $\tilde{\boldsymbol{\alpha}}$ decreases by one dimension.

References

Breslow, N. E. and D. G. Clayton (1993). Approximate inference in generalized linear mixed model. *Journal of the American Statistical Association 88*, 9–25.

Breslow, N. E. and X. Lin (1995). Bias correction in generalized linear mixed models with a single component of dispersion. *Biometrika 82*, 81–91.

Bühlmann, P. and B. Yu (2003). Boosting with the L2 loss: Regression and classification. *Journal of the American Statistical Association 98*, 324–339.

Eilers, P. H. C. (1995). Indirect observations, composite link models and penalized likelihood. In G. U. H. Seeber, B. J. Francis, R. Hatzinger, and G. Steckel-Berger (Eds.), *Proceedings of the 10th International Workshop on Statistical Modelling*. New York: Springer.

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

Fahrmeir, L., T. Kneib, and S. Lang (2004). Penalized structured additive regression for space-time data: a Bayesian perspective. *Statistica Sinica 14*, 731–761.

Fahrmeir, L. and G. Tutz (2001). *Multivariate Statistical Modelling Based on Generalized Linear Models* (2nd ed.). New York: Springer-Verlag.

Freund, Y. and R. E. Schapire (1996). Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pp. 148–156. San Francisco, CA: Morgan Kaufmann.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics 29*, 337–407.

Friedman, J. H., T. Hastie, and R. Tibshirani (2000). Additive logistic regression: A statistical view of boosting. *Annals of Statistics 28*, 337–407.

Green, P. J. (1987). Penalized likelihood for general semi-parametric regression models. *International Statistical Review 55*, 245–259.

Hastie, T. and R. Tibshirani (1990). *Generalized Additive Models*. London: Chapman & Hall.

Kaslow, R. A., D. G. Ostrow, R. Detels, J. P. Phair, B. Polk, and C. R. Rinaldo (1987). The multicenter aids cohort study: rationale, organization and selected characteristic of the participants. *American Journal of Epidemiology 126*, 310–318.

Lin, X. and N. E. Breslow (1996). Bias correction in generalized linear mixed models with multiple components of dispersion. *Journal of the American Statistical Association 91*, 1007–1016.

Lin, X. and D. Zhang (1999). Inference in generalized additive mixed models by using smoothing splines. *Journal of the Royal Statistical Society B61*, 381–400.

Littell, R., G. Milliken, W. Stroup, and R. Wolfinger (1996). *SAS System for Mixed Models.* Cary, NC: SAS Institue Inc.

Marx, D. B. and P. H. C. Eilers (1998). Direct generalized additive modelling with penalized likelihood. *Comp. Stat. & Data Analysis 28*, 193–209.

Ruppert, D., M. P. Wand, and R. J. Carroll (2003). *Semiparametric Regression.* Cambridge: Cambridge University Press.

Tutz, G. and A. Groll (2011). Binary and Ordinal Random Effects Models Including Variable Selection. *Journal of Computational and Graphical Statistics*. Submitted.

Tutz, G. and W. Hennevogl (1996). Random effects in ordinal regression models. *Comput. Stat. & Data Analysis 22*, 537–557.

Tutz, G. and F. Reithinger (2007). Flexible semiparametric mixed models. *Statistics in Medicine 26*, 2872–2900.

Venables, W. N. and B. D. Ripley (2002). *Modern Applied Statistics with S* (Fourth ed.). New York: Springer.

Vonesh, E. F. (1996). A note on the use of laplace's approximatio for nonlinear mixed-effects models. *Biometrika 83*, 447–452.

Wand, M. P. (2000). A comparison of regression spline smoothing procedures. *Computational Statistics 15*, 443–462.

Whittaker, E. T. (1923). On a new method of graduation. *Proc. Edinborough Math. Assoc. 78*, 81–89.

Wolfinger, R. and M. O'Connell (1993). Generalized linear mixed models; a pseudolikelihood approach. *Journal Statist. Comput. Simulation 48*, 233–243.

Wood, S. N. (2004). Stable and efficient multiple smoothing parameter estimation for generalized additive models. *Journal of the American Statistical Association 99.*

Wood, S. N. (2006). *Generalized Additive Models: An Introduction with R*. London: Chapman & Hall.

Zeger, S. L. and P. J. Diggle (1994). Semi-parametric models for longitudinal data with application to CD4 cell numbers in HIV seroconverters. *Biometrics 50*, 689–699.

Zhang, D., X. Lin, J. Raz, and M. Sowers (1998). Semi-parametric stochastic mixed models for longitudinal data. *Journal of the American Statistical Association 93*, 710–719.