LMU
LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

INSTITUT FÜR STATISTIK

Gerhard Tutz, Sebastian Petry

Generalized Additive Models with Unknown Link Function Including Variable Selection

Technical Report Number 145, 2013
Department of Statistics
University of Munich

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

Generalized Additive Models with Unknown Link Function Including Variable Selection

Gerhard Tutz & Sebastian Petry
Ludwig-Maximilians-Universität München
Akademiestraße 1, 80799 München
{tutz, petry}@stat.uni-muenchen.de

May 21, 2013

Abstract

The generalized additive model is a well established and strong tool that allows to model smooth effects of predictors on the response. However, if the link function, which is typically chosen as the canonical link, is misspecified, substantial bias is to be expected. A procedure is proposed that simultaneously estimates the form of the link function and the unknown form of the predictor functions including selection of predictors. The procedure is based on boosting methodology, which obtains estimates by using a sequence of weak learners. It strongly dominates fitting procedures that are unable to modify a given link function if the true link function deviates from the fixed function. The performance of the procedure is shown in simulation studies and illustrated by a real world example.

Keywords: Variable Selection, Generalized Additive Models, Single Index Models, Link Function Estimation.

1 Introduction

Methods for the estimation of the unknown link function have been considered in particular within the framework of single index models (SIMs). Let data be given by $(y_i, \boldsymbol{x}_i)$, $i = 1, \dots, n$, where y_i denotes the response and $\boldsymbol{x}_i = (x_{i1}, ..., x_{ip})^T$ the vector of p covariates. In SIMs, as discussed, for example, by Weisberg and Welsh (1994), Ruckstuhl and Welsh (1999), Härdle et al. (1993), and Yu and Ruppert (2002), the conditional expectation of y_i given $\boldsymbol{x}_i$, is modeled by

$$E(y_i|\boldsymbol{x}_i) = \mu_i = h_T(\eta_i),$$

where $h_T(.)$ is the true unknown response function and $\eta_i = \boldsymbol{x}_i^T\boldsymbol{\gamma}$ is the linear predictor that contains no intercept. The unknown $h_T(.)$ has to be estimated nonparametrically. Härdle et al. (1993) used kernel functions whereas Yu and Ruppert (2002) and Muggeo and Ferrara (2008) used a P-Spline approach. For uniqueness, typically the Euclidean norm of the parameter vector $\boldsymbol{\gamma}$ is fixed at 1, that is, $\|\boldsymbol{\gamma}\| = 1$, and the intercept is absorbed into the response function $h_T(.)$. To guarantee uniqueness of the estimates of a SIM an additional constraint is needed. Yu and Ruppert (2002) and Cui et al. (2009) set one specific component of $\boldsymbol{\gamma}$ to be positive. Alternatively, a monotonicity restriction on the response function guarantees uniqueness. We will consider only monotonically increasing (isotone) response functions with $\partial h_T(t)/\partial t > 0$. An advantage is that monotonicity entails invertibility of the response function, which is fundamental in generalized linear models (GLMs). Within the framework of GLMs the inverse of the response function is usually called link function, $h_T^{-1}(.) = g_T(.)$. With the monotonicity constraint SIMs are equivalent to GLMs with unknown response function. Typically, when GLMs are used the response function is considered fixed and known. In most applications the canonical response function is chosen. However, among others, Czado and Santner (1992) showed that misspecified response functions can lead to a substantial bias in the estimate of $\boldsymbol{\gamma}$.

The same holds for the more general class of generalized additive models (GAMs). In contrast to GLMs, where the predictor is a linear combination of covariates, in GAMs the predictor is given as a sum of unspecified functions of covariates. The conditional expectation is modeled by a transformation $h_T(.)$ of a sum of covariate functions and an intercept β_0 in the form

$$\mu_i = E(y_i|\boldsymbol{x}_i) = h_T(\beta_0+\sum_{j=1}^{p} f_j(x_{ij})), \text{ s.t. } \int_{\min\{\boldsymbol{x}_j\}}^{\max\{\boldsymbol{x}_j\}} f_j(t)dt = 0, \; j = 1, ..., p. \quad (1)$$

The constraint $\int_{\min\{\boldsymbol{x}_j\}}^{\max\{\boldsymbol{x}_j\}} f_j(t)dt = 0$, $j = 1, ..., p$ is needed to guarantee uniqueness because a shift of the function $\widetilde{f}_j(.) = f_j(t) + c_j$ can be compensated by a shift of the intercept $\widetilde{\beta}_0 = \beta_0 - c_j$. An extensive discussion of GAMs was given by Hastie and Tibshirani (1990) and Wood (2006).

As in GLMs, in GAMs usually the canonical response function is chosen and substantial bias has to be expected if the true response function differs strongly from the canonical response function. The focus of the present paper is on GAMs for which the response function is unknown and has to be estimated. In addition we allow for the selection of predictors. The advantages of variable selection in GAMs are the same as in GLMs. If variables with low or no influence are excluded noise is eliminated and the predictive performance of the estimated model increases. In particular when many predictors are available selection of predictors is unavoidable.

Nowadays many procedures for variable selection are available in the case of known link functions. For GLMs, in particular $L1$-penalization is a very popular way to obtain variable selection, see for example Tibshirani (1996), Park and Hastie (2007), Goeman (2010). More general penalty terms that work for GAMs were considered by Avalos et al. (2007) and Marra and Wood (2011). An alternative strategy is componentwise boosting with early stopping. Tutz and Binder (2006) and Bühlmann and Hothorn (2007) presented such boosting techniques for GLMs and GAMs. For variable selection in the case of unknown link functions not much seems to be available. A procedure that works for GLMs was proposed by Tutz and Petry (2012). For single index models, which allow non-monotonic response functions, and will not be considered here, selection procedures were proposed, for example, by Naik and Tsai (2001) and Kong and Xia (2007).

In Section 2 the model with flexible link and estimation procedures are introduced. In Section 3 the performance is evaluated in simulation studies. An application to the modelling of deaths rates in Sao Paulo is given in Section 4.

2 Flexible Link Generalized Additive Models (FLGAM)

The model that is assumed to hold has the form

$$\mu_i = E(y_i|\boldsymbol{x}_i) = h_T(\sum_{j=1}^{p} f_j(x_{ij})). \tag{2}$$

But, in contrast to conventional GAMs, the functions $f_j(.)$ as well as the response function $h_T(.)$ are unknown. For unspecified link function the model is not identifiable because it is equivalent to the model $E(y_i|\boldsymbol{x}_i) = \tilde{h}_T(\sum_{j=1}^{p} a \cdot f_j(x_{ij}) + b)$ for constants a, b and appropriately chosen response function $\tilde{h}_T$. Therefore additional constraints are needed. Similar to the constraints in SIMs, where the norm of the parameter vector is held constant, we postulate that for a fixed value $c > 0$ the constraint

$$\sum_{j=1}^{p} \int_{\min\{\boldsymbol{x}_j\}}^{\max\{\boldsymbol{x}_j\}} f_j(t)^2 dt = c \tag{3}$$

holds. In addition, each predictor function is centered by postulating

$$\int_{\min\{\boldsymbol{x}_j\}}^{\max\{\boldsymbol{x}_j\}} f_j(t) dt = 0, \, j = 1, ..., \, p. \tag{4}$$

Moreover, we assume that the response function is monotonically increasing, that is,

$$\frac{\partial h_T(t)}{\partial t} \geq 0. \tag{5}$$

In summary, the data generating model is given by (2) together with the constraints (3), (4), and (5).

In the estimating model the true response function is approximated by a composition of functions of the form $h_T(.) = h_0(h(.))$, where $h_0(.)$ is a fixed response function, typically the canonical response function. The inner (unknown) function $h(.)$ that has to be estimated is specified by a basis expansion

$$h(\eta_i) = \Phi^T(\eta_i)\boldsymbol{\alpha} = \sum_{k=1}^{m_h} \phi_k(\eta_i)\alpha_k,$$

where $\Phi(\eta_i) = \Phi_i$ is the vector of the m_h B-spline basis functions evaluated at η_i and $\boldsymbol{\alpha}$ is the corresponding basis coefficient vector of the inner function. B-spline basis expansions have been proposed by De Boor (1978) and have become very popular in many fields (see Eilers and Marx, 1996; Ramsey and Silverman, 2005).

For the estimation of the functions $f_j(.)$, which are also unknown in the classical GAM we also use a B-spline basis expansion given by

$$f_j(x) = \boldsymbol{\psi}_j^T(x)\boldsymbol{\beta}_j,\ j = 1,\ ...,\ p, \tag{6}$$

where $\boldsymbol{\psi}_j(x)$ is the vector of the m_j basis functions evaluated at x and $\boldsymbol{\beta}_j$ is the corresponding coefficient vector. Let $\boldsymbol{\psi}_{ij} := \boldsymbol{\psi}_{ij}(x_{ij})$ denote the vector of basis function evaluated at observation x_{ij}. The estimating model in the case of GAMs becomes

$$\mu_i = E(y_i|\boldsymbol{x}_i) = h_T\left(\beta_0 + \sum_{j=1}^{p} \boldsymbol{\psi}_{ij}^T\boldsymbol{\beta}_j\right).$$

In summary, the estimating model for the ith observation is

$$\mu_i = h_0\left(\Phi^T\left(\sum_{j=1}^{p} \boldsymbol{\psi}_{ij}^T\boldsymbol{\beta}_j\right)\boldsymbol{\alpha}\right) \tag{7}$$

subject to the constraints given by (3), (4), and (5).

2.1 Estimation Procedure

For a compact notation, let $\boldsymbol{\Psi}_j = (\boldsymbol{\psi}_{1j},\ ...,\ \boldsymbol{\psi}_{nj})^T$ denote the $n \times m_j$-dimensional matrix of basis function evaluated at all observation of the jth covariate and $\boldsymbol{\Psi} = (\boldsymbol{\Psi}_1,\ ..,\ \boldsymbol{\Psi}_p)$ denote the $n \times (\sum_{j=1}^{p} m_j)$ total design matrix without intercept. Let $\boldsymbol{\Phi} = (\Phi_1,\ ...,\ \Phi_n)^T$ denote the basis expansion evaluate at each observation. Note that the intercept and a multiplicative factor is absorbed into the inner function $h(\eta) = \Phi^T(\eta)\boldsymbol{\alpha}$.

Then the estimating model (7) for all observations can be given in vector form as

$$\boldsymbol{\mu}(\boldsymbol{\alpha},\ \boldsymbol{\beta}) = h_0\left(\boldsymbol{\Phi}(\boldsymbol{\Psi}\boldsymbol{\beta})\boldsymbol{\alpha}\right). \tag{8}$$

As in GAMs we assume that the response is from a simple exponential family

$$f(y_i|\theta_i,\, \phi) = \exp\left\{ \frac{y_i\theta_i - b(\theta_i)}{\phi} + c(y_i,\, \phi) \right\}, \tag{9}$$

where θ_i is the natural parameter of the family, ϕ is a dispersion parameter and $b(.)$, $c(.)$ are specific functions corresponding to the type of the family.

While the log-likelihoods of GAMs depends only on $\boldsymbol{\beta}$ the log-likelihood function for the more general model is

$$l(\boldsymbol{\alpha},\, \boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i\theta_i - b(\theta_i))/\phi.$$

It depends on $\boldsymbol{\alpha}$ by $h(\eta_i) = \boldsymbol{\Phi}_i^T \boldsymbol{\alpha}$ and on $\boldsymbol{\beta}$ by $\mu_i = h_0(h(\boldsymbol{x}_i^T \boldsymbol{\beta}))$. Estimates are obtained by minimizing the log-likelihood function $l(\boldsymbol{\alpha},\, \boldsymbol{\beta})$, subject to constraints. We present an algorithm which is based on boosting techniques. Each boosting iteration splits into two steps

1. Update of the response function with the predictor fixed.

2. Update of the predictor with the response function fixed.

Both updates are based on penalized and constrained Fisher scoring, respectively. In each boosting iteration only one of these steps is carried out. For the updating of the predictor we use componentwise boosting. Therefore, only one predictor function is updated within one iteration. Early stopping ensures that not all variables are updated and variable selection is obtained. First we give an unrestricted version of the algorithm and then we will add the necessary constraints.

2.1.1 Estimation of Response Function for Fixed Predictor

Let $\widehat{\boldsymbol{\eta}}^{(l-1)} = \boldsymbol{\Psi}\widehat{\boldsymbol{\beta}}^{(l-1)}$ be the fixed estimate of the predictor of the previous step. Then the estimation of the response function corresponds to fitting the model $\boldsymbol{\mu} = h_0(\widehat{\boldsymbol{\Phi}}^{(l-1)}\widehat{\boldsymbol{\alpha}}^{(l-1)} + \widehat{\boldsymbol{\Phi}}^{(l-1)}\widehat{\boldsymbol{a}}^{(l)})$ where $\widehat{\boldsymbol{\Phi}}^{(l-1)}\widehat{\boldsymbol{\alpha}}^{(l-1)}$ is the previously fitted value, which is included as an offset. Note that $\boldsymbol{\Phi}^{(l-1)}$ denotes the evaluation of basis functions at the current value $\widehat{\boldsymbol{\eta}}^{(l-1)}$. One step of penalized Fisher scoring has the form

$$\begin{aligned} \widehat{\boldsymbol{a}}^{(l)} \;=\; & \nu_h \left((\widehat{\boldsymbol{\Phi}}^{(l-1)})^T \widehat{\boldsymbol{D}}_h^{(l-1)} (\widehat{\boldsymbol{\Sigma}}^{(l-1)})^{-1} \widehat{\boldsymbol{D}}_h^{(l-1)} \widehat{\boldsymbol{\Phi}}^{(l-1)} + \lambda_h \boldsymbol{K}_h \right)^{-1} \cdot \\ & \times (\widehat{\boldsymbol{\Phi}}^{(l-1)})^T \widehat{\boldsymbol{D}}_h^{(l-1)} (\widehat{\boldsymbol{\Sigma}}^{(l-1)})^{-1} (\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^{(l-1)}), \end{aligned} \tag{10}$$

where

$$\widehat{\boldsymbol{D}}_h^{(l-1)} = \mathrm{diag}\left\{ \frac{\partial h_0(\widehat{h}^{(l-1)}(\widehat{\eta}_i^{(l-1)}))}{\partial \widehat{h}^{(l-1)}(\eta)} \right\}_{i=1}^{n} \tag{11}$$

is the estimated derivative matrix evaluated at the estimate of the previous step $h_0(\widehat{h}^{(l-1)}(\eta))$ and

$$\widehat{\boldsymbol{\Sigma}}^{(l-1)} = \text{diag}\left\{\sigma^2(h_0(\widehat{h}^{(l-1)}(\widehat{\eta}_i^{(l-1)})))\right\}_{i=1}^{n} \tag{12}$$

is the matrix of variances evaluated at $h_0(\widehat{h}^{(l-1)}(\eta))$ and $\boldsymbol{K}_h$ is the penalty matrix which penalizes the second derivative of the estimated (approximated) response function. The matrix $\boldsymbol{K}_h$ is symmetric and each entry has the form

$$\boldsymbol{K}_h = \{k_{ij}\}, \text{ with } k_{ij} = \int (\frac{d^2}{d\eta^2}\phi_i(t))(\frac{d^2}{d\eta^2}\phi_j(t))dt. \tag{13}$$

The main idea of boosting is to approximate the optimum in small steps. If the step size is too large the procedure suffers. Therefore, one uses the concept of weak learning proposed by Shapire (1990), see also Bühlmann and Yu (2003). In our procedure the weakness of learners is enforced by large λ_h and small ν_h. The latter is fixed by using $\nu_h = 0.1$. Since λ_h only penalizes the second derivative of the functions an additional shrinkage parameter $\nu_h = 0.1$ is helpful to make the learner weak (see also Tutz and Binder, 2006; Schmid and Hothorn, 2008; Hothorn et al., 2010).

2.1.2 Componentwise Boosting for Fixed Response Function

Let $\widehat{h}^{(l-1)}(.)$ be the fixed estimate of the response function of the previous step. The design matrix of the predictor is $\boldsymbol{\Psi} = (\boldsymbol{\Psi}_1, ..., \boldsymbol{\Psi}_p)$ and $\boldsymbol{\beta}^T = (\boldsymbol{\beta}_1^T, ..., \boldsymbol{\beta}_p^T)$ is the corresponding parameter vector. Componentwise boosting for additive predictors means that within one boosting step only one subvector $\boldsymbol{\beta}_j$ of $\boldsymbol{\beta}$ is updated. So we fit the model $\boldsymbol{\mu} = h_0(\widehat{h}^{(l-1)}(\boldsymbol{\Psi}\widehat{\boldsymbol{\beta}}^{(l-1)} + \boldsymbol{\Psi}_j\boldsymbol{b}_j))$, where $\boldsymbol{\Psi}\widehat{\boldsymbol{\beta}}^{(l-1)}$ is a fixed offset representing the previous update. Therefore only the covariate $\boldsymbol{x}_j$ is included in the model. The penalized Fisher scoring for parameter $\boldsymbol{\beta}_j$ has the form

$$\begin{aligned}\widehat{\boldsymbol{b}}_j^{(l)} =& \nu_f \left(\boldsymbol{\Psi}_j^T\widehat{\boldsymbol{D}}_{\boldsymbol{\eta}}^{(l-1)}(\widehat{\boldsymbol{\Sigma}}^{(l-1)})^{-1}\widehat{\boldsymbol{D}}_{\boldsymbol{\eta}}^{(l-1)}\boldsymbol{\Psi}_j + \lambda_f\boldsymbol{K}_j\right)^{-1} \\ & \times \boldsymbol{\Psi}_j^T\widehat{\boldsymbol{D}}_{\boldsymbol{\eta}}^{(l-1)}(\widehat{\boldsymbol{\Sigma}}^{(l-1)})^{-1}(\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^{(l-1)}),\end{aligned} \tag{14}$$

where $\nu_f = 0.1$ is a fixed shrinkage parameter and

$$\begin{aligned}\widehat{\boldsymbol{D}}_{\boldsymbol{\eta}}^{(l-1)} &= \text{diag}\left\{\frac{\partial h_0(\widehat{h}^{(l-1)}(\widehat{\eta}_i^{(l-1)}))}{\partial\eta}\right\}_{i=1}^{n} \\ &= \text{diag}\left\{\frac{\partial h_0(\widehat{h}^{(l-1)}(\widehat{\eta}_i^{(l-1)}))}{\partial h^{(l-1)}(\eta)} \cdot \frac{\partial\widehat{h}^{(l-1)}(\widehat{\eta}_i^{(l-1)})}{\partial\eta}\right\}_{i=1}^{n}\end{aligned} \tag{15}$$

is the matrix of derivatives evaluated at the values of the previous iteration, and

$$\widehat{\boldsymbol{\Sigma}}^{(l-1)} = \text{diag}\left\{\sigma^2(h_0(\widehat{h}(\widehat{\eta}_i^{(l-1)})))\right\}_{i=1}^{n} \tag{16}$$

is the variance from the previous step and $\boldsymbol{K}_j$ is a penalty matrix which penalizes the second derivatives of the predictor functions. $\boldsymbol{K}$ is a symmetric matrix and is similar to (13). Let $\psi_{jk}(.)$ be the kth basis function of the jth basis function then

$$\boldsymbol{K}_j = \{k_{j|kl}\} = \int (\frac{d^2}{d\eta^2}\psi_{jk}(t))(\frac{d^2}{d\eta^2}\psi_{jl}(t))dt.$$

As for the update of the response function we fix $\nu_f = 0.1$ to make the procedure a weak learner.

2.1.3 Constraints

As already mentioned, for uniqueness three constraints must be fulfilled. First we consider the constraints of the predictor. The predictor is constrained in two ways. The first set of constraints is

$$\int_{\min\{\boldsymbol{x}_j\}}^{\max\{\boldsymbol{x}_j\}} \boldsymbol{\psi}_j^T(t)\widehat{\boldsymbol{\beta}}_j dt = \int_{\min\{\boldsymbol{x}_j\}}^{\max\{\boldsymbol{x}_j\}} \sum_{k=1}^{m_j} \psi_{jk}(t)\hat{\beta}_{jk}0,\ j = 1,\ ...,\ p.$$

It is fulfilled if in each update step

$$\sum_{k=1}^{m_j} w_{jk}\beta_{jk} = 0,\ j = 1,\ ...,\ p \text{ with } w_{jk} = \int \psi_{jk}(t)dt \tag{17}$$

holds for all predictor functions. The restricted quadratic optimization problem that corresponds to one penalized Fisher scoring step (14) with the constraints (17) is

$$\begin{aligned}\widehat{\boldsymbol{b}}_j = \nu_f \operatorname{argmin}_{\boldsymbol{b}\in\mathbb{R}^{m_j}} \Big\{ & \boldsymbol{b}_j^T \left(\boldsymbol{\Psi}_j^T \widehat{\boldsymbol{D}}_{\boldsymbol{\eta}}^{(l-1)} (\widehat{\boldsymbol{\Sigma}}^{(l-1)})^{-1} \widehat{\boldsymbol{D}}_{\boldsymbol{\eta}}^{(l-1)} \boldsymbol{\Psi}_j + \lambda_f \boldsymbol{K}_j\right) \boldsymbol{b}_j \\ & -2\boldsymbol{b}_j^T \boldsymbol{\Psi}_j^T \widehat{\boldsymbol{D}}_{\boldsymbol{\eta}}^{(l-1)} (\widehat{\boldsymbol{\Sigma}}^{(l-1)})^{-1} (\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^{(l-1)}),\ \text{s.t. } (17) \Big\}.\end{aligned} \tag{18}$$

The linear restricted quadratic optimization problem is solved by use of the R-package `quadprog` from Turlach (2009).

The second constraint of the predictor is

$$\sum_{j=1}^{p} \int_{\min\{\boldsymbol{x}_j\}}^{\max\{\boldsymbol{x}_j\}} f_j(t)^2 dt = c,$$

which for the expansion in basis functions has the form

$$\sum_{j=1}^{p} \int_{\min\{\boldsymbol{x}_j\}}^{\max\{\boldsymbol{x}_j\}} \left(\sum_{k=1}^{m_j} (\beta_{jk}\psi_{jk}(t))\right)^2 dt = c, \tag{19}$$

where the choice of c is arbitrary. After updating the jth subvector of $\boldsymbol{\beta}$ by (18) we scale $\boldsymbol{\beta}$ to Euclidean norm 1, $\|\boldsymbol{\beta}\| = 1$, which determines the value c. We use natural cubic B-splines (compare Dierckx, 1993). This basis expansion is provided by the `fda` package in `R` (Ramsay et al., 2010). For illustration the B-spline basis for 8 equidistant inner knots within $[-1, 1]$ are shown in Figure 1.

Since the basis functions $\psi_{jk}(t)$ for fixed j sum up to 1 it follows fom the Cauchy-Schwarz inequality and $\|\boldsymbol{\beta}\| = 1$ that the range of the linear predictor $\eta = \sum_{j=1}^{p} \boldsymbol{\psi}_j(x_j)^T \boldsymbol{\beta}_j$ is in $[-\sqrt{p}, \sqrt{p}]$. So the domain of $h(.)$ is known and the knots of its basis expansion are fixed on this range.

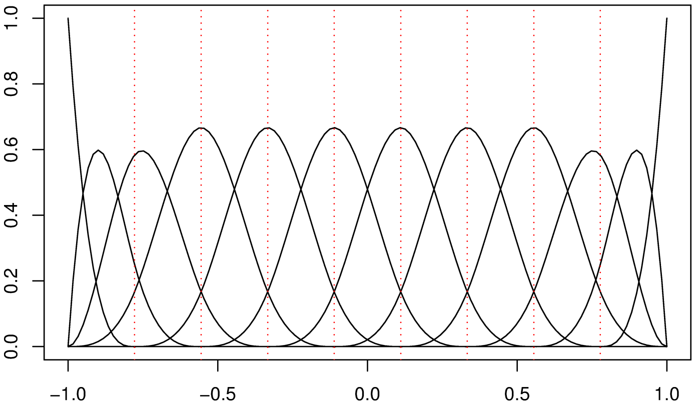

Figure 1: *The cubic natural B-spline basis for* 8 *equidistant inner knots on the interval* $[-1, 1]$.

2.1.4 Constraints for the Response Function

We assume that the response function $h_T(.) = h_0(h(.))$ is monotonically non-decreasing. Since the canonical link function is non-decreasing we have to estimate a monotonically non-decreasing inner function $h(.)$. The inner function is approximated by a basis expansion $h(\eta_i) = \Phi^T(\eta_i)\boldsymbol{\alpha}$, where $\Phi^T(\eta_i)$ is a vector of basis functions evaluated at η_i and $\boldsymbol{\alpha}$ is the corresponding coefficient vector. $h(\eta) = \Phi^T(\eta)\boldsymbol{\alpha}$ is monotonically non-decreasing if the components of the coefficient vector $\boldsymbol{\alpha}$ are monotonically non-decreasing, i.e. $\alpha_i \leq \alpha_{i+1}$ for all $i = 1, ..., m_h - 1$. A boosting update has the form $\widehat{\boldsymbol{\alpha}}^{(l)} = \widehat{\boldsymbol{\alpha}}^{(l-1)} + \widehat{\boldsymbol{a}}^{(l)}$. So after each update step the system of inequations $\widehat{\alpha}_i^{(l-1)} + \widehat{a}_i^{(l)} \leq \widehat{\alpha}_{i+1}^{(l-1)} + \widehat{a}_{i+1}^{(l)}$, $i = 1, ..., m_h - 1$, must be fulfilled . Each update step is restricted on the following space

$$\mathcal{A} = \left\{ \boldsymbol{a}^{(l)} : a_2^{(l)} - a_1^{(l)} \geq \widehat{\alpha}_1^{(l-1)} - \widehat{\alpha}_2^{(l-1)}, ..., a_k^{(l)} - a_{k-1}^{(l)} \geq \widehat{\alpha}_{k-1}^{(l-1)} - \widehat{\alpha}_k^{(l-1)} \right\}. \quad (20)$$

$\mathcal{A}$ can be rewritten as a system of inequations. In the same way as for restricted updates of the predictor functions we use the corresponding quadratic optimization problem with linear constraints $\mathcal{A}$

$$\begin{aligned}\widehat{\boldsymbol{a}} = \nu_h \operatorname{argmin}\Big\{&\boldsymbol{a}^T\Big(\boldsymbol{\Phi}^T\widehat{\boldsymbol{D}}^{(l-1)}(\widehat{\boldsymbol{\Sigma}}^{(l-1)})^{-1}\widehat{\boldsymbol{D}}^{(l-1)}\boldsymbol{\Phi}+\lambda_h\boldsymbol{K}_h\Big)\boldsymbol{a}\\ &-2\boldsymbol{a}^T\boldsymbol{\Phi}^T\widehat{\boldsymbol{D}}^{(l-1)}(\widehat{\boldsymbol{\Sigma}}^{(l-1)})^{-1}(\boldsymbol{y}-\widehat{\boldsymbol{\mu}}^{(l-1)}),\ \text{ s.t. } \boldsymbol{a}\in\mathcal{A}\Big\}.\end{aligned} \tag{21}$$

by using the R-package `quadprog` (see Turlach, 2009).

2.2 Algorithm

The basic algorithm is given below and shows the interplay of the two steps. In each iteration step an update of the predictor function as well as an update of the response functions are computed. In the last step of each iteration it is evaluated which update is to be preferred and is actually performed. In the final step only the maximizer of the log-likelihood function is used. Thus in each iteration step either one predictor function or the response function is updated.

Algorithm: FLGAM

Step 1 (Initialization)

Set $\widehat{\boldsymbol{\beta}}^{(0)} = \mathbf{0}$ and $\widehat{\boldsymbol{\eta}}^{(0)} = \mathbf{0}$. Determine $\boldsymbol{\alpha}^{(0)}$ so that $\boldsymbol{\phi}(t)^T\boldsymbol{\alpha}^{(0)} = g(\bar{\boldsymbol{y}}) + 0.0001t$ is a line with small gradient and intercept $g(\bar{\boldsymbol{y}})$, where $\bar{\boldsymbol{y}} = \sum_{i=1}^{n} y_i$. Compute $\widehat{\boldsymbol{D}}^{(0)}$, $\widehat{\boldsymbol{D}}_{\boldsymbol{\eta}}^{(0)}$ and $\widehat{\boldsymbol{\Sigma}}^{(0)}$.

Step 2 (Iteration)

For $l = 1, 2, \ldots, M$

1. *Predictor update*
    - Compute for each $j \in \{1, \ldots, p\}$ the update $\widehat{\boldsymbol{b}}_j^{(l)}$ as described in (18) and set $\boldsymbol{b}_j^{(l)} = (\mathbf{0}^T, \ldots, (\widehat{\boldsymbol{b}}_j^{(l)})^T, \ldots, \mathbf{0}^T)^T$ and determine the update candidate
    $$\boldsymbol{\beta}_j^{(l)} = \widehat{\boldsymbol{\beta}}^{(l-1)} + \boldsymbol{b}_j^{(l)}.$$
    - Compute $\widehat{\boldsymbol{\beta}}_j^{(l)} = \boldsymbol{\beta}_j^{(l)}/||\boldsymbol{\beta}_j^{(l)}||$ and the corresponding log-likelihood function $l(\boldsymbol{\alpha}^{(l-1)}, \widehat{\boldsymbol{\beta}}_j^{(l)})$.
    - Choose the parameter vector $\widehat{\boldsymbol{\beta}}_{opt}^{(l)} = \operatorname{argmax}_{\widehat{\boldsymbol{\beta}}_j^{(l)}, j=1,\ldots,p}\, l(\boldsymbol{\alpha}^{(l-1)}, \widehat{\boldsymbol{\beta}}_j^{(l)})$ which minimizes the log-likelihood function and set $\widehat{\boldsymbol{\beta}}^{(l)} = \widehat{\boldsymbol{\beta}}_{opt}^{(l)}$

2. *Response function update*
   - Compute $\widehat{\boldsymbol{a}}^{(l)}$ as described in (21) and set $\widehat{\boldsymbol{\alpha}}^{(l)} = \widehat{\boldsymbol{\alpha}}^{(l-1)} + \widehat{\boldsymbol{a}}^{(l)}$
   - Compute $\hat{h}^{(l)}(\boldsymbol{\eta}^{(l-1)}) = \boldsymbol{\Phi}^{(l-1)}\widehat{\boldsymbol{\alpha}}^{(l)}$ and the corresponding log-likelihood function $l(\widehat{\boldsymbol{\alpha}}^{(l)}, \widehat{\boldsymbol{\beta}}^{(l-1)})$.
3. *Update choice*
   - If $l(\widehat{\boldsymbol{\alpha}}^{(l)}, \widehat{\boldsymbol{\beta}}^{(l-1)}) > l(\widehat{\boldsymbol{\alpha}}^{(l-1)}, \widehat{\boldsymbol{\beta}}^{(l)})$ then $\boldsymbol{\alpha}^{(l)}$ is updated and $\widehat{\boldsymbol{\beta}}$ remains unchanged, $\widehat{\boldsymbol{\beta}}^{(l)} = \widehat{\boldsymbol{\beta}}^{(l-1)}$.
   - If $l(\widehat{\boldsymbol{\alpha}}^{(l)}, \widehat{\boldsymbol{\beta}}^{(l-1)}) \leq l(\widehat{\boldsymbol{\alpha}}^{(l-1)}, \widehat{\boldsymbol{\beta}}^{(l)})$ then $\widehat{\boldsymbol{\beta}}^{(l)}$ is updated and $\widehat{\boldsymbol{\alpha}}$ remains unchanged, $\hat{\boldsymbol{\alpha}}^{(l)} = \boldsymbol{\alpha}^{(l-1)}$.

Note that we transform the domain of each function by $\widetilde{\boldsymbol{x}}_j = \boldsymbol{x}_j/(\max\{\boldsymbol{x}_j\} - \min\{\boldsymbol{x}_j\})$, and the range of each domain is normed to 1. By this transformation the update of each function becomes more similar.

2.2.1 Choice of Tuning Parameter

The FLGAM procedure uses three tuning parameter: λ_f for the smoothing of the predictor function, λ_h for the smoothing of the response function and m_{stop} for the number of boosting iterations. We use 5-fold cross-validation for determining these parameters. While λ_f and λ_h serve only to obtain a weak learner the number of iterations is the crucial tuning parameter. Therefore the former two are chosen from a coarse grid of parameter values with $\lambda_h \in \{0.5, 1, 0.5\}$ and $\lambda_f \in \{0.5, 1, 0.5\}$ as candidates. The maximal number of boosting iteration was set to $M = 5000$. All in all, we have to cross-validate the model for nine tuning parameter constellations over 5000 boosting iterations.

2.2.2 Cut Version

An unsatisfying property of the presented boosting procedure is that some predictors are updated only once or twice. To enforce variable selection we also present a cut version of the algorithm in which estimated functions that are close to zero are excluded. If in the (l)th iteration the Euclidean length of the coefficient vector of the jth predictor function is smaller than $1/p$, $\|\boldsymbol{\beta}_j^{(l)}\| < 1/p$, we set the corresponding subvector to $\mathbf{0}$, that is, $\boldsymbol{\beta}_j^{(l)} = \mathbf{0}$. The new cut parameter vector is restandardized to Euclidean norm 1. The optimal tuning parameter for the cut version $\widetilde{\lambda}_h$, $\widetilde{\lambda}_f$ and $\widetilde{m}_{stop}$ are also determined by cross-validation. In the simulation study the threshold $1/p$ worked quite well. Of course the threshold limits could be optimized.

3 Simulation Study

To evaluate the performance of the FLGAM procedure we compare it with three established procedures:

GAMBoost , which is a likelihood based boosting procedure which performs variable selection by early stopping (Tutz and Binder, 2006).

mboost , which is a boosting procedure proposed by Hothorn et al. (2010) that also enforces variable selection by early stopping. The corresponding R-package is `mboost` (Hothorn et al., 2009).

mgcv , which fits a GAM with variable selection based on penalization. For details see Wood (2006) and Wood (2011).

We use two model assessment measurements for the comparison of models. After determining the optimal model by 5-fold crossvalidation we predict $\widehat{\boldsymbol{\mu}}_{test} = h_0(\Phi(\boldsymbol{\Psi}_{test}\widehat{\boldsymbol{\beta}})\widehat{\boldsymbol{\alpha}})$ based on an independently chosen data set $(\boldsymbol{y}_{test}, \boldsymbol{X}_{test})$ and evaluate the predictive deviance

$$\text{Dev(test)} = -2(l(\boldsymbol{y}_{test}, \widehat{\boldsymbol{\mu}}_{test}) - l(\boldsymbol{y}_{test}, \boldsymbol{y}_{test})).$$

The accuracy of the estimated predictor functions is evaluated by

$$\text{MSE}_f = \sum_{j=1}^{p} \int (\tilde{f}_j(t) - \hat{\tilde{f}}_j(t))^2 dt,$$

which compares two scaled versions of the functions. $\tilde{f}_j(t) = f_j(t) \cdot F$ is the true function $f_j(t)$ scaled by $F = (\sum_{j=1}^{p} \int f_j(t)^2 dt)^{-1}$ and $\widehat{\tilde{f}}_j(t) = \widehat{f}_j(t) \cdot \widehat{F}$ is the corresponding estimate, where $\widehat{F} = (\sum_{j=1}^{p} \int \widehat{f}_j(t)^2 dt)^{-1}$. The transformation makes the results comparable. Note that MSE_f measures only the similarity of shape between the estimate and the true function.

We investigate three cases of distribution: normal, Poisson, and binomial with non-canonical response function. For the *normal case* we use a sigmoid response function

$$h_{\text{Norm}}(\eta) = \frac{20}{1 + \exp(-5 \cdot \eta)}$$

and so the response is generated by $y_i = N(h_{\text{Norm}}(\eta_i), 1)$. In the *Poisson case* we use a sigmoid response function similar to the normal case

$$h_{\text{Pois}}(\eta) = \frac{10}{1 + \exp(-5 \cdot \eta)}$$

but the response is generated by $y_i = Pois(h_{\text{Pois}}(\eta_i))$. For the *binomial case* we choose an increasing smooth step function

$$h_{\text{Bin}}(\eta) = \frac{0.25}{1 + \exp(-10 \cdot \eta - 15)} + \frac{0.75}{1 + \exp(-10 \cdot \eta + 15)}$$

with three levels 0, 0.25, and 1. The transitions between these levels are quite smooth. The response is generated by $y_i = Bin(h(\eta_i))$.

In each setting the predictor has the same form. The predictor η is generated by p covariate characteristic functions $f_j(.)$. Beyond the distribution and the response function each setting is given by the number of covariates, $p = 5$, 10, 25. Only the first five covariates have influence on the response. The predictor function are $f_1(x_1) = \sin(4 \cdot x_1)$, $f_2(x_2) = \cos(4 \cdot x_2)$, $f_3(x_3) = 0.5 \cdot x_3^2$, $f_4(x_4) = -0.5 \cdot x_4^2$, $f_5(x_5) = x_5^3/9$, $f_j(x_j) = 0$, $j = 6 \ldots, p$. Predictors are drawn from a truncated Normal distribution to avoid problems with outliers. We use the R-package `tmvtnorm` (see Genz et al., 2011) with the range for each covariate being restricted to $[-\pi, \pi]$. The mean of the generating distribution is fixed to $\boldsymbol{\mu} = \mathbf{0}_p$ and the covariance matrix is $\boldsymbol{\Sigma}^2 = \{\sigma_{jk}^2\}_{j,\, k=1,\ldots,p}$ where $\sigma_{jk}^2 = 1$ for $j = k$ and $\sigma_{jk}^2 = 0.5$ otherwise. For the normal and the Poisson case the number of observations of the training dataset is $n_{train} = 250$ and the test datasets have $n_{test} = 1000$ observations. In the binomial case the 0-1 information of the response is quite weak. Thus in contrast to the both other cases we increase the number of observations to $n_{train} = 1000$ and $n_{test} = 4000$. In all cases the maximal number of boosting iterations is set to $M = 5000$ which is never exhausted over all settings. Each predictor function $f_1(.)$, $f_2(.)$, ..., $f_5(.)$ is expanded by cubic B-splines basis with 20 (inner) knots. The response function is expanded in the same way with 50 (inner) knots.

The results are summarized in Table 1. For illustration we show the boxplots of MSE_f and Dev(test) for the Poisson case in Figure 2. It is seen that the fitting of a flexible link function provided by FLGAM and FLGAM(cut) strongly outperforms the procedures with fixed canonical response function in terms of prediction. The only exception is the binomial case with 25 predictors in which `mboost` performs slightly better than the FLGAM procedures. In particular in the Poisson case the FLGAM and the FLGAM(cut) procedures show superior performance. In terms of MSE_f, which measures the accuracy of the function fits, `mboost` is a strong competitor, which in some cases even slightly outperforms the FLGAM procedures. In general the `mboost` performs quite well for MSE_f. The procedure uses a carefully designed update step, which is controlled by the degrees of freedom (cf. Hothorn et al., 2010; Hofner et al., 2009, 2011). This strategy seems to be very successful even if the response function is misspecified.

The cut version of the FLGAM procedure performs quite similar to the simple FLGAM procedure in terms of MSE_f and predictive deviance (Dev(test)). But the false positive rates of FLAP (cut) are the best across all settings. Surprisingly `mgcv` includes all covariates in each setting although the option variable selection was chosen. It should be noted that in some cases the procedure did not converge. In the binomial case we leave out the `GAMBoost` because the high number of observations increases the computational costs immensely.

		FLAP	FLAP (cut)	mgcv*	GAMBoost**	mboost
Normal distribution						
$p = 5$	MSE$_f$	0.0132	0.0132	0.0153	0.0238	0.0106
	Dev(test)	10884.34	10954.11	26810.77	25874.14	42457.43
	hits	1.000	0.988	1.000	1.000	1.000
	false pos.	—	—	—	—	—
$p = 10$	MSE$_f$	0.0172	0.0166	0.0174	0.0235	0.0130
	Dev(test)	15589.11	14254.97	28275.76	27325.15	40147.39
	hits	1.000	1.000	1.000	1.000	1.000
	false pos.	1.000	0.608	1.000	0.972	0.952
$p = 25$	MSE$_f$	0.0209	0.0208	0.0299	0.0235	0.0188
	Dev(test)	25375.87	25112.45	38757.15	27150.12	45861.43
	hits	1.000	1.000	1.000	1.000	1.000
	false pos.	0.906	0.539	1.000	0.753	0.834
Poisson distribution						
$p = 5$	MSE$_f$	0.0103	0.0103	0.0295	0.0408	0.0176
	Dev(test)	1610.23	1610.23	3921.14	4593.06	4226.54
	hits	1.000	1.000	1.000	1.000	1.000
	false pos.	—	—	—	—	—
$p = 10$	MSE$_f$	0.0143	0.0138	0.0322	0.0530	0.0205
	Dev(test)	2017.60	2033.99	5432.79	8127.55	4570.11
	hits	1.000	1.000	1.000	1.000	1.000
	false pos.	0.996	0.304	1.000	0.912	0.884
$p = 25$	MSE$_f$	0.0253	0.0258	0.0489	0.0482	0.0262
	Dev(test)	2877.95	2872.24	1025052	5382.05	4637.24
	hits	1.000	1.000	1.000	1.000	1.000
	false pos.	0.789	0.413	1.000	0.803	0.643
Binomial distribution						
$p = 5$	MSE$_f$	0.0132	0.0139	0.0183	—	0.0135
	Dev(test)	4226.13	4264.90	4280.19	—	4235.23
	hits	1.000	0.952	1.000	—	1.000
	false pos.	—	—	—	—	—
$p = 10$	MSE$_f$	0.0182	0.0184	0.0232	—	0.0171
	Dev(test)	4335.16	4324.11	4356.42	—	4325.13
	hits	1.000	0.996	1.000	—	1.000
	false pos.	0.912	0.232	1.000	—	0.984
$p = 25$	MSE$_f$	0.0216	0.0221	0.0295	—	0.0228
	Dev(test)	4455.67	4461.87	4627.87	—	4439.88
	hits	0.992	0.988	1.000	—	1.000
	false pos.	0.529	0.237	1.000	—	0.745

Table 1: *Medians of the* Dev(*test*) *and* MSE$_f$ *for each setting of the simulation study and the hits false positive rates across the replications.* * *Convergence only in part of the replications.* ** *No results in the binomial case because of high computational costs.*

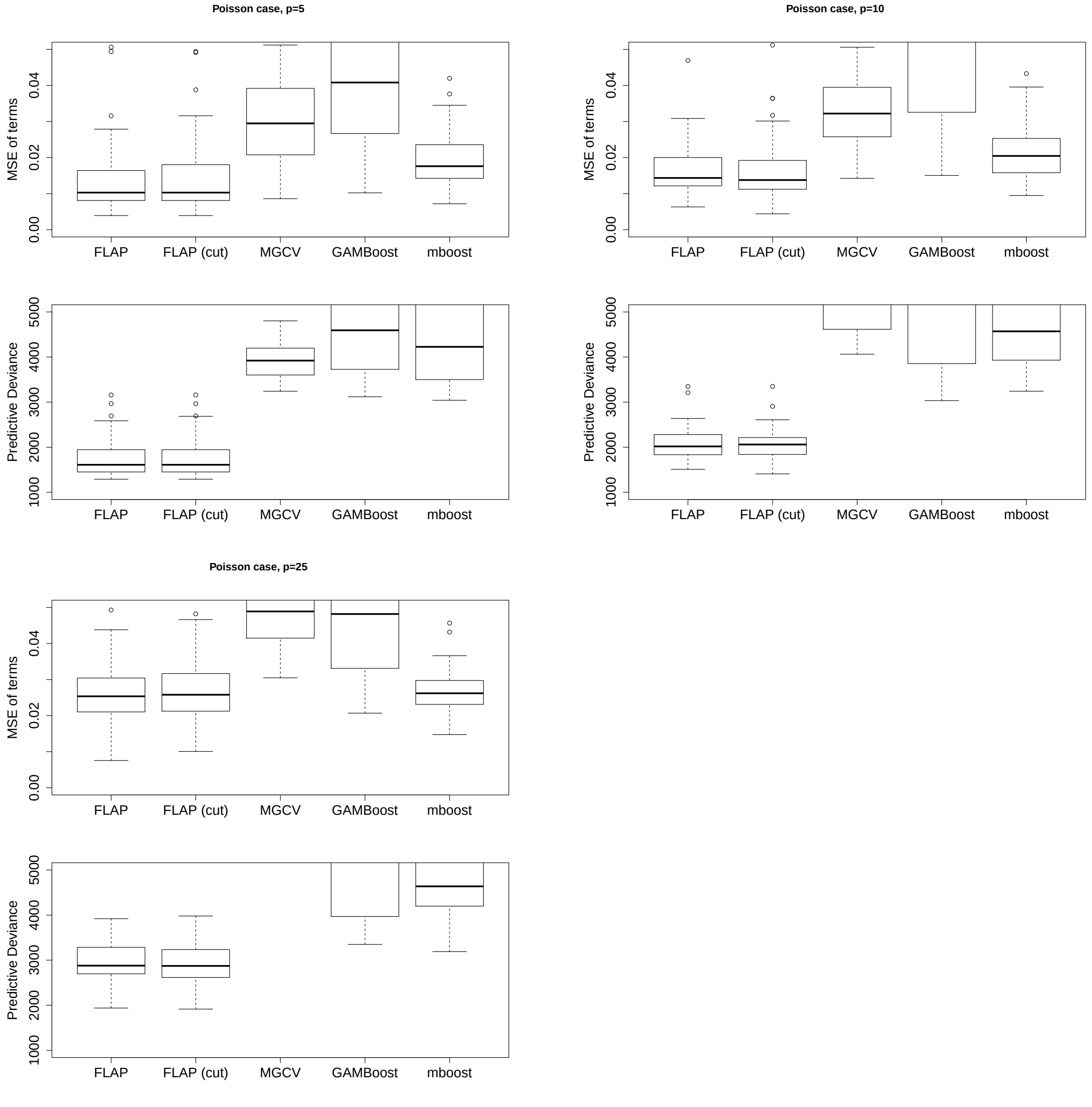

TABLE 2: *Boxplots of* Dev(*test*) *and* MSE_f *for the three Poisson setting.*

4 Data Example

The method is illustrated by modeling the death rate in the metropolitan area of Sao Paulo. The data were recorded from January 1994 to December 1997 $n = 1351$ days and are available at `http://www.ime.usp.br/~jmsinger/Polatm9497.zip`. We use a sub data set which was also used by Leitenstorfer and Tutz (2007) for the modelling of monotone functions. The response is the number of daily deaths caused by respiratory reasons of people which are 65 years or older `RES65`. The covariates are given in Table 3.

Label	Explanation
`TEMPO`	Time in days
`SO2ME.2`	The 24-hours mean of SO_2 concentration (in μ/m^3) over all monitoring measurement stations.
`TMIN.2`	The daily minimum temperature.
`UMID`	The daily relative humidity.
`DIASEM`	Day of the week. (1 =Tuesday, 2 =Wednesday, ..., 7 =Monday)
`CAR65`	Cardialogical caused deaths per day.
`OTH65`	Other (non respiratory or cardiological) caused deaths per day.

TABLE 3: *Table of covariates and their labels of the Sao Paulo air pollution data set.*

For `SO2ME.2` and `TMIN.2` we consider the measurements taken 2 days before as influential. This lag was proposed by Conceicao et al. (2001). All predictors are modelled nonparametrically. We used 20 knots for all covariates in FLGAM. For `mgcv`, the default values were used, but for the covariate `DIASEM` we had to reduce the number of knots to 7. We determined the optimal tuning parameter by a 5-fold cross-validation, where $\lambda_h, \lambda_f \in \{100, 10, 1, 0.1, 0.01\}$. For both versions of the FLGAM we got $\lambda_h = 1$ and $\lambda_f = 0.01$. For all boosting procedures the maximal number of boosting iterations was fixed to 1000. In Figures 2 we show the results for FLGAM and in 3 the results for mgcv. We do not show results for `GAMBoost` because the procedure did not work well on this dataset. As in mgcv for mboost (not shown) no predictor was selectd and estimates were similar to the results for mgcv.

For `TEMPO` the periodic character is identified by all procedures. The SO_2 concentration (`SO2ME.2`) has an clearly increasing trend. If we neglect the high valued outliers this covariate seems to have only a very weak influence. Increasing temperature `TMIN.2` has a decreasing influence on the response `RES65`. This characteristic was detected by all procedures. With the FLGAM this trend seems to be stronger. Beyond the outliers, the covariates `UMID` and `DIASEM` have only a small influence on the response. The cut version of FLGAM (FLGAM (cut)) does not include these covariates. Increasing number of non respiratory caused deaths (`CAR65` and `OTH65`) tends to increase the number respiratory caused deaths.

In contrast to the established methods with canonical link the models with estimated link function have only two main influential covariates, `TEMPO` and `TMIN.2`. The models with canonical link functions are more complex. In them also the covariates `SO2ME.2`, `CAR65`, and `OTH65` seem to be influential.

Figure 2 also shows the estimated response functions of both FLGAM procedures, which differ from the canonical response functions shown in Figure 3.

In addition we evaluated the prediction across 50 random splits. The training data set contains 1000 observations and the remaining observations are used as test data. For reducing the computational costs we determined the tuning parameter λ_F and λ_h on the complete data set ($n = 1351$) by 5fold cross-validation, and fixed the resulting $\lambda_h = 1$ and $\lambda_f = 0.01$ for the following investigation of prediction. Since we only had to determine the number of optimal boosting iterations by a 5fold cross-validation on the training data set the computational costs were strongly reduced. We used the training data for fitting the model for given tuning parameters and measured the prediction on the test data. We give the medians of the predictive deviances across the random splits and the deviance for complete data set in Table 4. The predictive deviance across the random splits underlines the results of the simulations study, prediction tends to be better when allowing for flexible link functions.

	FLGAM	FLGAM (cut)	`mgcv`	`mboost`
complete data set	1383.06	1412.30	1547.58	1407.42
random splits	411.26	414.13	437.63	430.52

Table 4: *Prediction measurements of the Sao Paulo data set for the different procedures. First row: The deviance on the complete data set. Second row: Median across 50 random splits.*

5 Conclusion and Perspectives

A competitive method for estimating GAMs with an unspecified response function is presented. The method is based on componentwise boosting, by early stopping variable selection is obtained. Especially in terms of the predictive performance the method dominates in nearly all simulation settings. In all cases the variable selection of the FLGAM and FLGAM(cut) procedures works quite well which is seen from the hits and false positive rates. Especially the variable selection of the cut version works very well.

By small modifications the FLGAM procedures can be generalized to semiparametric models, where smooth, linear, and categorial predictors are included. Since then predictors have quite different complexity, in particular the use of the

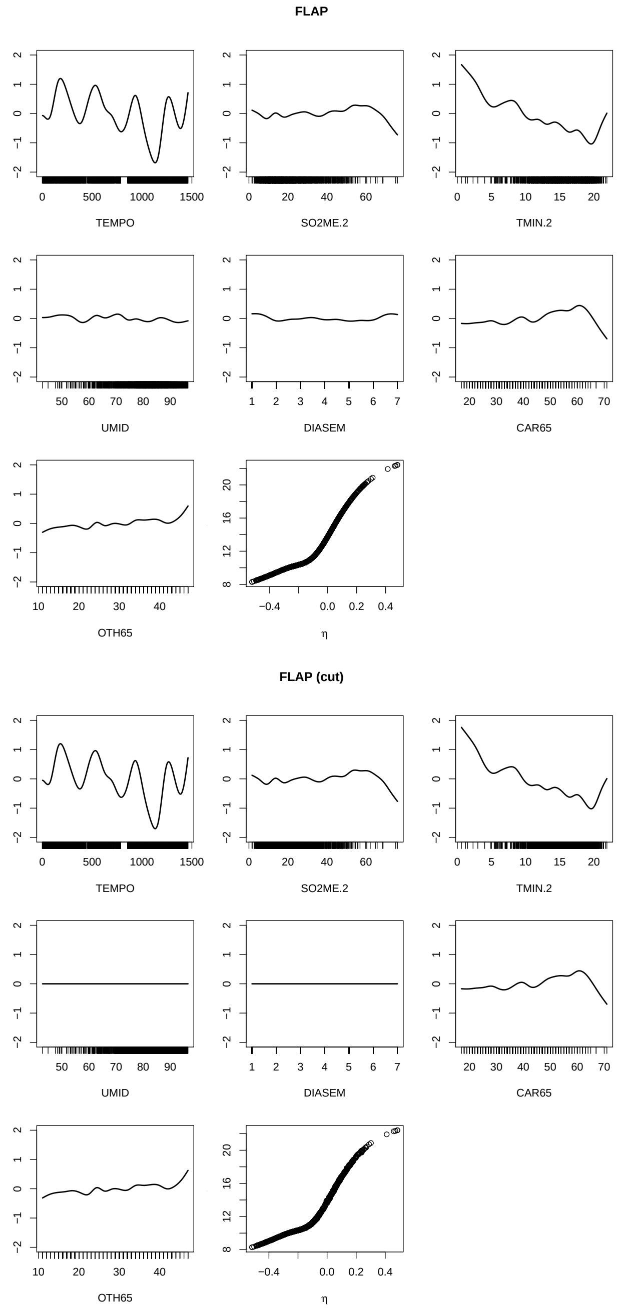

Figure 2: *Top: Estimated predictor functions and response function for Sao Paulo data set estimated by FLGAM. Bottom: Estimated predictor functions and response function for Sao Paulo data estimated by the cut version of the FLGAM algorithm.*

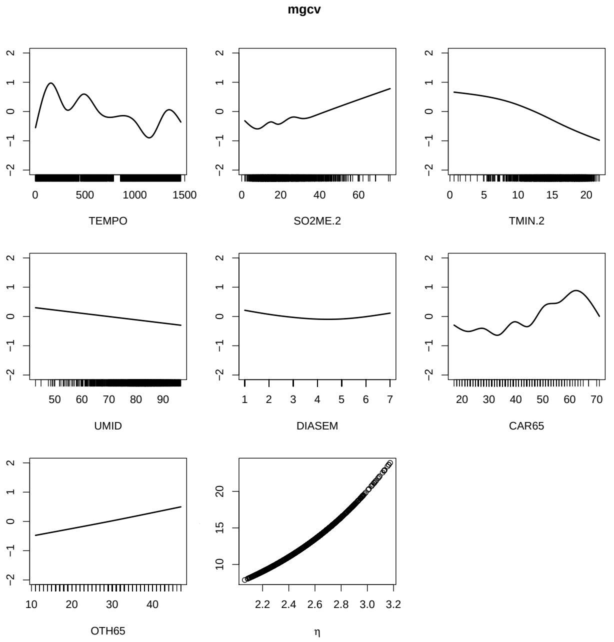

Figure 3: *Estimated predictor functions and (fixed) response function for Sao Paulo data set estimated by* `mgcv`.

degree of freedom based update criterion proposed by Hofner et al. (2009, 2011) should be useful.

References

Avalos, M., Y. Grandvalet, and C. Ambroise (2007). Parsimonious additive models. *Computational Statistics and Data Analysis 51*(6), 2851–2870.

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

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

Conceicao, G. M. S., S. G. E. K. Miraglia, H. S. Kishi, P. H. N. Saldiva, and J. M. Singer (2001). Air pollution and children mortlity: a time series study in são paulo, brazil. *Environmental Health Perspectives 109*, 347–350.

Cui, X., W. K. Härdle, and L. Zhu (2009). Generalized single index models: The efm approach. Discussion Paper 50, SFB 649, Humboldt University Berlin, Economic Risk.

Czado, Y. and T. Santner (1992). The effect of link misspecification on binary regression inference. *Journal of statistical planning and inference 33*, 213–231.

De Boor, C. (1978). *A Practical Guide to Splines.* New York: Springer-Verlag.

Dierckx, P. (1993). *Curve and Surface Fitting with Splines.* Oxford Science Publications.

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

Genz, A., F. Bretz, T. Miwa, X. Mi, F. Leisch, F. Scheipl, and T. Hothorn (2011). *mvtnorm: Multivariate Normal and t Distributions.* R package version 0.9-999.

Goeman, J. (2010). L1 penalized estimation in the cox proportional hazards model. *Biometrical Journal 52*, 70–84.

Härdle, W., P. Hall, and H. Ichimura (1993). Optimal smoothing in single-index models. *The Annals of Statistics 21*, 157–178.

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

Hofner, B., T. Hothorn, T. Kneib, and M. Schmid (2009). A framework for unbiased model selection based on boosting. Technical Report 72, Department of Statistics LMU Munich.

Hofner, B., T. Hothorn, T. Kneib, and M. Schmid (2011). A framework for unbiased model selection based on boosting. *Journal of Computational and Graphical Statistics (to appear) 39*(5), 1–13.

Hothorn, T., P. Buehlmann, T. Kneib, M. Schmid, and B. Hofner (2010)). Model-based boosting 2.0. *Journal of Machine Learning Research 11*, 2109–2113.

Hothorn, T., P. Bühlmann, T. Kneib, M. Schmid, and B. Hofner (2009). *mboost: Model-Based Boosting.* R package version 2.0-0.

Kong, E. and Y. Xia (2007). Variable selection for the single-index model. *Biometrika 94*(1), 217–229.

Leitenstorfer, F. and G. Tutz (2007). Generalized monotonic regression based on b-splines with an application to air pollution data. *Biostatistics 8*, 654–673.

Marra, G. and S. Wood (2011). Practical variable selection for generalized additive models. *Computational Statistics and Data Analysis 55*, 2372–2387.

Muggeo, V. M. R. and G. Ferrara (2008). Fitting generalized linear models with unspecified link function: A p-spline approach. *Computational Statistics & DataAnalysis 52*(5), 2529–2537.

Naik, P. and C. Tsai (2001). Single-index model selections. *Biometrika 88*(3), 821–832.

Park, M. Y. and T. Hastie (2007). L1 regularization-path algorithm for generalized linear models. *Journal of the Royal Statistical Society B 69*, 659–677.

Ramsay, J. O., H. Wickham, S. Graves, and G. Hooker (2010). *fda: Functional Data Analysis*. R package version 2.2.5.

Ramsey, J. O. and B. W. Silverman (2005). *Functional Data Analysis*. New York: Springer.

Ruckstuhl, A. and A. Welsh (1999). Reference bands for nonparametrically estimatedlink functions. *Journal of Computational and Graphical Statistics 8*(4), 699–714.

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

Shapire, R. E. (1990). The strength of weak learnability. *Machine Learning 5*, 197–227.

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

Turlach, B. A. (2009). *quadprog: Functions to solve Quadratic Programming-Problems*. R package version 1.4-11, S original by Berwin A. Turlach, R portby Andreas Weingessel.

Tutz, G. and H. Binder (2006). Generalized additive modeling with implicit variable selection by likelihood-based boosting. *Biometrics 62*, 961–971.

Tutz, G. and S. Petry (2012). Nonparametric estimation of the link function including variable selection. *Statistics and Computing 21*, 545–561.

Weisberg, S. and A. H. Welsh (1994). Adapting for the missing link. *Annals of Statistics 22*, 1674–1700.

Wood, S. (2011). *mgcv: GAMs with GCV/AIC/REML smoothness estimation and GAMMs by PQL*. R package version 1.7-2.

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

Yu, Y. and D. Ruppert (2002). Penalized spline estimation for partially linear single-index models. *Journal of the American Statistical Association 97*, 1042–1054.