Published in final edited form as:
*Methods Inf Med*. 2012 ; 51(2): 168–177. doi:10.3414/ME11-02-0021.

# Regularization for Generalized Additive Mixed Models by Likelihood-Based Boosting

**Andreas Groll**[*] and **Gerhard Tutz**[#]

[*]Department of Statistics, University of Munich, Akademiestrasse 1, D-80799, Munich, Germany, andreas.groll@stat.uni-muenchen.de, *phone*: +49 89 2180 2925, *fax* : +49 89 2180 5308

[#]Department of Statistics, University of Munich, Akademiestrasse 1, D-80799, Munich, Germany, gerhard.tutz@stat.uni-muenchen.de, *phone*: +49 89 2180 3044, *fax*: +49 89 2180 5308

## Abstract

**Objective—**With the emergence of semi- and nonparametric regression the generalized linear mixed model has been extended to account for additive predictors. However, available fitting methods fail in high dimensional settings where many explanatory variables are present. We extend the concept of boosting to generalized additive mixed models and present an appropriate algorithm that uses two different approaches for the fitting procedure of the variance components of the random effects.

**Methods—**The main tool developed is likelihood-based componentwise boosting that enforces variable selection in generalized additive mixed models. In contrast to common procedures they can be used in high-dimensional settings where many covariates are available and the form of the influence is unknown. The complexity of the resulting estimators is determined by information criteria. The performance of the methods is investigated in simulation studies for binary and Poisson responses with comparisons to alternative approaches and it is applied to clinical real world data.

**Results—**Simulations show that the proposed methods are considerably more stable and more accurate in estimating the regression function than the conventional approach, especially when a large number of predictors is available. The methods also produce reasonable results in applications to real data sets, which is illustrated by the Multicenter AIDS Cohort Study.

**Conclusions—**The boosting algorithm allows to extract relevant predictors in generalized additive mixed models. It works in high-dimensional settings and is very stable.

### Keywords

Generalized additive mixed model; Boosting; Smoothing; Variable selection; Penalized quasi-likelihood

## 1 Introduction

Generalized additive mixed models (GAMMs) are an extension of generalized additive models incorporating random effects. They are widely used to model correlated and clustered responses. For example, the dependence structure of longitudinal data and of designs with repeated measurements can be captured. Due to heavy computational problems in the estimation of parameters modeling usually is restricted to a moderate number of predictor variables. 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 re-weighted 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

[1] have presented their famous AdaBoost many extensions have been developed (e.g. gradient boosting by [2], generalized linear and additive regression based on the $L_2$-loss by [3]). In particular boosting as an optimization technique in function space, investigated for example by [4], is an attractive method for the modeling of high-dimensional data. An experimental evaluation of boosting methods for classification is found in [5], who compare the AdaBoost with gradient boosting ensembles of regression trees both in a simulation study and in a clinical application on breast tumor diagnosis. A detailed overview of componentwise boosting is given in [6].

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. An application to the Multicenter AIDS Cohort Study (MACS, see [7, 8]) is presented in Section 5, which is based on the CD4 cell data of male American citizens who are infected with HIV.

## 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 $\boldsymbol{y}_i^T=(y_{i1},\ldots,y_{iT_i})$. Let $\boldsymbol{x}_{it}^T=(1, x_{it1},\ldots, x_{itp})$ be the covariate vector associated with be the covariate vector associated with fixed effects and $\boldsymbol{z}_{it}^T=(z_{it1},\ldots, z_{itq})$ the covariate vector associated with cluster-specific random effects $\boldsymbol{b}_i \sim N(0,\boldsymbol{Q})$, where $\mathbf{Q}$ is a $q \times q$ dimensional known or unknown covariance matrix. It is assumed that the observations $y_{it}$ are conditionally independent with means $\mu_{it} = E(y_{it}|\boldsymbol{b}_i, \boldsymbol{x}_{it}, \boldsymbol{z}_{it})$ and variances $var(y_{it}|\boldsymbol{b}_i)= \phi v(\mu_{it})$, where $v(.)$ is a known variance function and $\phi$ is a scale parameter.

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

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

, where $g$ is a monotonic differentiable link function, $\eta_{it}^{\text{par}}=\boldsymbol{x}_{it}^T\beta$ is a linear parametric term with parameter vector $\beta^T = (\beta_0, \beta_1,\ldots, \beta_p)$, 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 $\eta_{it}^{\text{rand}}=\boldsymbol{z}_{it}^T\boldsymbol{b}_i$ contains the random effects part. An alternative form that we also use in the following is

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

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 [8–10]. While [9] 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, [11–14].

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 [15]. More detailed information about the B-spline basis can be found for example in [16]. In the following let $\alpha_j^T=(\alpha_1^{(j)},\ldots,\alpha_k^{(j)})$ denote the unknown parameter vector of the $j$-th smooth function and let $\boldsymbol{B}_j^T(u)=(B_1^{(j)}(u;d),\ldots,B_k^{(j)}(u;d))$ represent the vector-valued evaluations of the $k$ basis functions. Then the parameterized model for (1) has the form

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

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

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

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

$$g(\mu)=\boldsymbol{X}\beta+\boldsymbol{B}_1\boldsymbol{\alpha}_1+\ldots+\boldsymbol{B}_m\alpha_m+\mathbf{Zb}, \quad (2)$$

With $\boldsymbol{B}_j^T=[\boldsymbol{B}_{1j}^T,\ldots,\boldsymbol{B}_{nj}^T]$ representing the transposed B-spline design matrix of the $j$-th smooth function as in equation (C.6) in Web Appendix C. The model can be further reduced to

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

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

## 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 $\boldsymbol{b}_i$, is of exponential family type

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

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 $\phi$ the dispersion parameter (for example [17]).

A popular method to maximize generalized mixed models is penalized quasi-likelihood (PQL), which has been suggested by [18–20]. In the following we briefly sketch the PQL approach for the semipara-metric model. As common in mixed models, we assume that the covariance matrix $\mathbf{Q}(\boldsymbol{\varrho})$ of the random effects $\boldsymbol{b}_i$ 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 $\phi$, 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

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

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

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

where $\boldsymbol{K}_j$ penalizes the parameters $\boldsymbol{\alpha}_j$ and $\lambda_j$ are smoothing parameters which control the influence of the $j$-th penalty term. When using P-splines one penalizes the difference between adjacent coefficients in the form $\lambda_j \alpha_j^T \boldsymbol{K}_j \alpha_j = \lambda_j \alpha_j^T (\boldsymbol{\Delta}^d)^T \boldsymbol{\Delta}^d \alpha_j$, where $\boldsymbol{\Delta}^d$ denotes the difference operator matrix of degree $d$, for details see, for example, [16]. The log-likelihood (4) has also been considered by [9] but with $\boldsymbol{K}_j$ 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 [18] one obtains the double penalized log-likelihood:

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

where the first penalty term $\sum_{i=1}^{n} \boldsymbol{b}_i^T \boldsymbol{Q}(\boldsymbol{\varrho})^{-1} \boldsymbol{b}_i$ is due to the approximation based on the Laplace method and the second penalty term $\sum_{j=1}^{m} \lambda_j \alpha_j^T \boldsymbol{K}_j \alpha_j$ determines the smoothness of the functions $\alpha_{(j)}(.)$, depending on the chosen smoothing parameter $\lambda_j$. The boosting algorithm proposed in the following aims at maximizing the penalized log-likelihood (5).

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` [13]. Further aspects were discussed by [21–23].

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, [17]).

## 3 Boosted GAMMs - bGAMM

According to [24], boosting is one of the most powerful learning ideas introduced in the last 20 years. Though it was originally designed for classification problems, it can be also applied to regression. In the form of componentwise boosting, where in each fitting step only one parameter or a group of parameters is refitted, the method allows to select relevant terms of the predictor. The methods vary with the criterion that is minimized and the learner that is used. A widely used criterion is $L_2$-loss and componentwise least-squares estimate as learner ([25]). It applies in particular in the linear model but can also be used in generalized linear models as an approximate learner that maximizes the log-likelihood ([6]). Likelihood-based methods that use Fisher scoring or variants thereof were considered by [2] for logit type models and for generalized additive models by [26]. An overview on available boosting methods is given in [6], see also [24].

In the following we will use likelihood-based boosting methods. The essential difference to the methods mentioned previously is that the data generating model is a mixed model. As a consequence, the likelihood has a quite different form and includes, in particular, parameters that specify the distribution of random effects. Therefore alternative algorithms have to be used. First steps to boosting in mixed models, but restricted to linear predictors, are found in [27]. 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, by componentwise ascent of the penalized log-likelihood from (5). We use a reparametrization technique explained in more detail in Appendix C. The B-spline design matrices $\boldsymbol{B}_j$ from equation (2), corresponding to the difference penalty matrices $\boldsymbol{K}_j$ 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 $\tilde{\boldsymbol{K}} := \tilde{\boldsymbol{K}}_j = diag(0,\ldots,0,1,\ldots,1)$, 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 D).

Moreover, we define $\boldsymbol{\Phi} := [\boldsymbol{\Phi}_1,\ldots,\boldsymbol{\Phi}_m]$ and introduce the new parameter vector $\boldsymbol{\gamma}^T := (\boldsymbol{\beta}^T, \tilde{\boldsymbol{\alpha}}^T, \boldsymbol{b}^T)$. The following boosting algorithm uses the EM-type algorithm given in [17]. We further want to introduce the vector $\gamma_r^T := (\beta^T, \tilde{\alpha}_r^T, \boldsymbol{b}^T)$, containing only the spline coefficients of the $r$-th smooth component. A detailed description of the single steps of the bGAMM algorithm can be found in Web Appendix A.

Algorithm bGAMM

---

**1** Initialization

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

**2** Iteration

For $l = 1, 2, \ldots$

**a.** Refitting of residuals

**i.** Computation of parameters

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

$$g(\mu) = \hat{\eta}^{(l-1)} + X\beta + \mathbf{\Phi}_r a_r + Zb$$

is fitted, where

$$\hat{\eta}^{(l-1)} = X\hat{\beta}^{(l-1)} + \mathbf{\Phi}\hat{a}^{(l-1)} + Z\hat{b}^{(l-1)}$$

is considered a known off-set.

Estimation refers to $\gamma_r^T = (\beta^T, a_r^T, b^T)$. 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

$$\hat{\gamma}_r^{(l)} = (F_r^{\text{pen}}(\hat{\gamma}^{(l-1)}))^{-1} s_r(\hat{\gamma}^{(l-1)}) \quad (6)$$

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

**ii.** Selection step

Select from $r \in \{1,\ldots, m\}$ the component $j$ that leads to the smallest $AIC_r^{(l)}$ or $BIC_r^{(l)}$ as given in Web Appendix A.3 and select the corresponding vector

$$(\hat{\gamma}_j^{(l)})^T = \left((\hat{\beta}^*)^T, (\hat{a}_j^*)^T, (\hat{b}^*)^T\right).$$

**iii.** Update

Set

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

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

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

$$(\hat{\gamma}^{(l)})^T = \left((\hat{\beta}^{(l)})^T, (\hat{a}_1^{(l)})^T, \ldots, (\hat{a}_m^{(l)})^T, (\hat{b}^{(l)})^T\right).$$

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

$$\hat{\eta}^{(l)} = A\hat{\gamma}^{(1)}$$

**b.** Computation of variance-covariance components

Estimates of $\hat{\mathbf{Q}}^{(l)}$ are obtained as approximate REML-type estimates or alternative methods (see Web Appendix A.2).

Note that the EM-type algorithm may be viewed as an approximate EM algorithm, where the posterior of $\boldsymbol{b}_i$ 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 $\boldsymbol{b}_i$ is normal, and so posterior mode and mean coincide, as do posterior covariance and curvature.

## 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 (see Web Appendix B). We also compare the algorithm to alternative approaches. The optimal smoothing parameter $\lambda$ chosen as the value $\lambda_{opt}$ which leads to the smallest *AIC* or *BIC* from (A.2) and (A.3), which are computed on a fine grid. Also general cross validation could be used, with the negative effect of expanding computational time.

### Bernoulli Data with Logit-Link

The underlying model is the random intercept additive Bernoulli model

$$\eta_{it}=\sum_{j=1}^{m} f_j(u_{itj})+b_i, \quad i=1,\ldots,40, \quad t=1,\ldots,10$$

$$E[y_{it}]=\frac{\exp(\eta_{it})}{1+\exp(\eta_{it})} := \pi_{it} \quad y_{it}\sim \mathrm{B}(1,\pi_{it})$$

with smooth effects given by

| | | | |
|---|---|---|---|
| $f_1(u) = 6\sin(u)$ | with | $u \in [-\pi, \pi]$ | |
| $f_2(u) = 6\cos(u)$ | with | $u \in [-\pi, 2\pi]$ | |
| $f_3(u) = u^2$ | with | $u \in [-\pi, \pi]$ | |
| $f_4(u) = 0.4u^3$ | with | $u \in [-\pi, \pi]$ | |
| $f_5(u) = -u^2$ | with | $u \in [-\pi, \pi]$ | |
| $f_j(u) = 0$ | with | $u \in [-\pi, \pi]$ | for $j = 6,\ldots,50$. |

We choose different settings $m = 5, 10, 15, 20, 50$. For $j = 1,\ldots,50$ the vectors $\boldsymbol{u}_{it}^T=(u_{it1},\ldots,u_{it50})$ have been drawn independently with components following a uniform distribution within the specified interval. The number of observations is fixed as $n = 40$, $T_i = T = 10$, $\forall i = 1,\ldots,n$. 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 [13], 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 $\sigma_b$ given by

$$\text{mse}_f := \sum_{t=1}^{N}\sum_{j=1}^{m}(f_j(v_{tj}) - \widehat{f}_j(v_{tj}))^2, \quad \text{mse}_{\sigma_b} := (\sigma_b - \widehat{\sigma}_b)^2,$$

where $v_{tj}$, $t = 1,\ldots,N$ denote fine and evenly spaced grids on the different predictor spaces for $j = 1,\ldots,m$. 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. More-over, *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 $f_j$, $j = 6,\ldots, m$, that were wrongly selected. As the `gamm` function is not able to perform variable selection it always estimates all functions $f_j$, $j = 1,\ldots, m$.

The results of all quantities for different scenarios of $\sigma_b$ 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 $\text{mse}_f$ and $\text{mse}_{\sigma_b}$ 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 Table 2.

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 $\text{mse}_f$ 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 $\text{mse}_f$. In terms of $\text{mse}_{\sigma_b}$ `gamm` dominates but the REML version of boosting comes close. For the EM version there is more variance $\hat{\sigma}_b$ in as well as more bias with the tendency of underestimating the true standard deviation for $\sigma_b \in \{0.4, 0.8\}$ and overestimating it for $\sigma = 1.6$, resulting in poorer estimates in terms of $\text{mse}_{\sigma_b}$. It is especially remarkable that the selection of relevant variables works that well that both $\text{mse}_f$ and $\text{mse}_{\sigma_b}$ hardly deteriorate with increasing number of noise variables.

Exemplarily, for the case $m = 5$ 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.

## 5 Application: The Multicenter AIDS Cohort Study

In this section we apply our boosting method on a real data set and compare the results of our method with the `gamm` approach. Standard errors for fixed effects and for $\hat{\sigma}_b$ as well as point-wise confidence bands have been derived by simulation-based parametric bootstrap evaluations (see Web Appendix E).

The CD4 cell 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 [7, 8]). 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). Note that all variables except of age are time-dependent.

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, compare [28], where a normal response model (the square root CD4 number) with additive effects has been regarded. We consider the semi-parametric mixed model with linear predictor $g(\mu_{it})=\eta_{it}=\eta_{it}^{par}+\eta_{it}^{add}+b_i$, where $\mu_{it}$ denotes the expected CD4 number of cells for subject $i$ on measurement $t$ (taken at irregular time intervals). The parametric and nonparametric terms are

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

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

$$\widehat{\phi}=\frac{1}{N-\text{df}}\sum_{i=1}^{n}\sum_{t=1}^{T_i}\widehat{r}_{it}^2, \quad N=\sum_{i=1}^{n}T_i,$$

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 ([13]) and for the `gamm` algorithm are given in Table 3.

The main interest is in the typical time course of CD4 cell decay and the variability across subjects (see also[8]). Figure 4 shows the data together with an estimated overall smooth effect of time on CD4 cell decay derived by the `gamm` function. In Figure 5 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, but the corresponding point-wise confidence interval indicates that the variable is not significant. For the `bGAMM` algorithm age is not selected and therefore has no effect at all.

For numerical comparison of `gamm` function and `bGAMM` algorithm in real data applications, we use the mean squared prediction error. We repeatedly split the data randomly into training and test data, fit the model on the training data and use the fitted parameters $\widehat{\beta}, \widehat{\alpha}$ and $\widehat{b}$ for the prediction of the response in the test data. With $n_{test}$ denoting the size of the test data we can derive the following prediction error on every random split:

$$mse_{pred}=\sum_{i=1}^{n_{test}}(y_i-\widehat{y}_i)^2.$$

As we include the fitted random effects $\hat{\mathbf{b}}_1,\ldots,\hat{\mathbf{b}}_n$ for prediction, we restrict the random splitting of the data by constraining the test data to contain only observations of clusters with at least $n_{min}$ replications, say for example $n_{min}=5$. In this way we want to ensure to use reasonable random effects estimates for prediction. Note, that instead of random splits also cross-validation could be used.

Figure 6 shows boxplots of the prediction error differences with the `gamm` function as the reference. The `bGAMM` algorithm clearly outperforms the `gamm` function in each random split.

## 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.

As clustered data often occur in clinical or biological contexts, where different individuals are observed over a period of time, the proposed methods are highly relevant in such applications. The application on the CD4 data has shown that variables that may be not significant (for example in terms of point-wise confidence intervals) can be excluded from the model and thus the accuracy of the regression model can be improved. This implicit selection of relevant variables is especially useful in clinical and biological trials where often many possibly relevant covariates are present.

For example in dealing with gene-expression data usually thousands of genes are available and one needs models that can handle and analyze such large systems. This creates keen challenges with respect to data analysis and data management and the existing software programs and analysis methods are still in the beginning (for overviews see for example [29] or [30]).

## References


1. Freund, Y.; Schapire, RE. Proceedings of the Thirteenth International Conference on Machine Learning. San Francisco, CA: Morgan Kaufmann; 1996. Experiments with a New Boosting Algorithm; p. 148-156.
2. Friedman JH, Hastie T, Tibshirani R. Additive logistic regression: A statistical view of boosting. Annals of Statistics. 2000; 28:337–407.
3. Bühlmann P, Yu B. Boosting with the L2 loss: Regression and classification. Journal of the American Statistical Association. 2003; 98:324–339.
4. Friedman JH. Greedy function approximation: a gradient boosting machine. Annals of Statistics. 2001; 29:337–407.
5. Stollhoff R, Sauerbrei W, Schumacher M. An Experimental Evaluation of Boosting Methods for Classification. Methods of Information in Medicine. 2010; 49:219–229. [PubMed: 20135078]
6. Bühlmann P, Hothorn T. Boosting algorithms: Regularization, prediction and model fitting. Statistical Science. 2007; 22:477–505.
7. Kaslow RA, Ostrow DG, Detels R, Phair JP, Polk BF, Rinaldo CR. The multicenter AIDS cohort study: rationale, organization and selected characteristic of the participants. American Journal of Epidemiology. 1987; 126:310–318. [PubMed: 3300281]
8. Zeger SL, Diggle PJ. Semi-parametric models for longitudinal data with application to CD4 cell numbers in HIV seroconverters. Biometrics. 1994; 50:689–699. [PubMed: 7981395]
9. Lin X, Zhang D. Inference in Generalized Additive Mixed Models by Using Smoothing Splines. Journal of the Royal Statistical Society. 1999; B61:381–400.
10. Zhang D, Lin X, Raz J, Sowers M. Semi-parametric stochastic mixed models for longitudinal data. Journal of the American Statistical Association. 1998; 93:710–719.
11. Marx DB, Eilers PHC. Direct Generalized Additive Modelling with Penalized Likelihood. Comp Stat & Data Analysis. 1998; 28:193–209.
12. Wood SN. Stable and efficient multiple smoothing parameter estimation for generalized additive models. Journal of the American Statistical Association. 2004; 99:673–686.
13. Wood, SN. Generalized Additive Models: An Introduction with R. London: Chapman & Hall; 2006.
14. Wand MP. A Comparison of Regression Spline Smoothing Procedures. Computational Statistics. 2000; 15:443–462.

15. Ruppert, D.; Wand, MP.; Carroll, RJ. Semiparametric Regression. Cambridge: Cambridge University Press; 2003.
16. Eilers PHC, Marx BD. Flexible Smoothing with B-Splines and Penalties. Statistical Science. 1996; 11:89–121.
17. Fahrmeir, L.; Tutz, G. Multivariate Statistical Modelling Based on Generalized Linear Models. 2nd ed. New York: Springer-Verlag; 2001.
18. Breslow NE, Clayton DG. Approximate Inference in Generalized Linear Mixed Model. Journal of the American Statistical Association. 1993; 88:9–25.
19. Lin X, Breslow NE. Bias Correction in Generalized Linear Mixed Models with Multiple Components of Dispersion. Journal of the American Statistical Association. 1996; 91:1007–1016.
20. Breslow NE, Lin X. Bias Correction in Generalized Linear Mixed Models with a Single Component of Dispersion. Biometrika. 1995; 82:81–91.
21. Wolfinger R, O'Connell M. Generalized Linear Mixed Models; A Pseudolikelihood Approach. Journal of Statistical Computation and Simulation. 1993; 48:233–243.
22. Littell, R.; Milliken, G.; Stroup, W.; Wolfinger, R. SAS System for Mixed Models. Cary, NC: SAS Institute Inc; 1996.
23. Vonesh EF. A note on the use of Laplace's approximation for nonlinear mixed-effects models. Biometrika. 1996; 83:447–452.
24. Hastie, T.; Tibshirani, R.; Friedman, JH. The Elements of Statistical Learning. 2nd ed. New York: Springer; 2009.
25. Bühlmann P. Boosting for high-dimensional linear models. Annals of Statistics. 2006; 34:559–583.
26. Tutz G, Binder H. Generalized additive modelling with implicit variable selection by likelihood based boosting. Biometrics. 2006; 62:961–971. [PubMed: 17156269]
27. Tutz G, Groll A. Binary and Ordinal Random Effects Models Including Variable Selection. Journal of Computational and Graphical Statistics. 2011 Submitted.
28. Tutz G, Reithinger F. A boosting approach to flexible semiparametric mixed models. Statistics in Medicine. 2007; 26:2872–2900. [PubMed: 17133647]
29. Parmigiani, EG.; Garrett, ES.; Irizarry, RA.; Zeger, SL. The Analysis of Gene Expression Data: Methods and Software. New-York: Springer-Verlag; 2003.
30. Dudoit S, Gentleman RC, Quackenbush J. Open source software for the analysis of microarray data. Biotechniques. 2003; 34:45–51. [PubMed: 12664684]

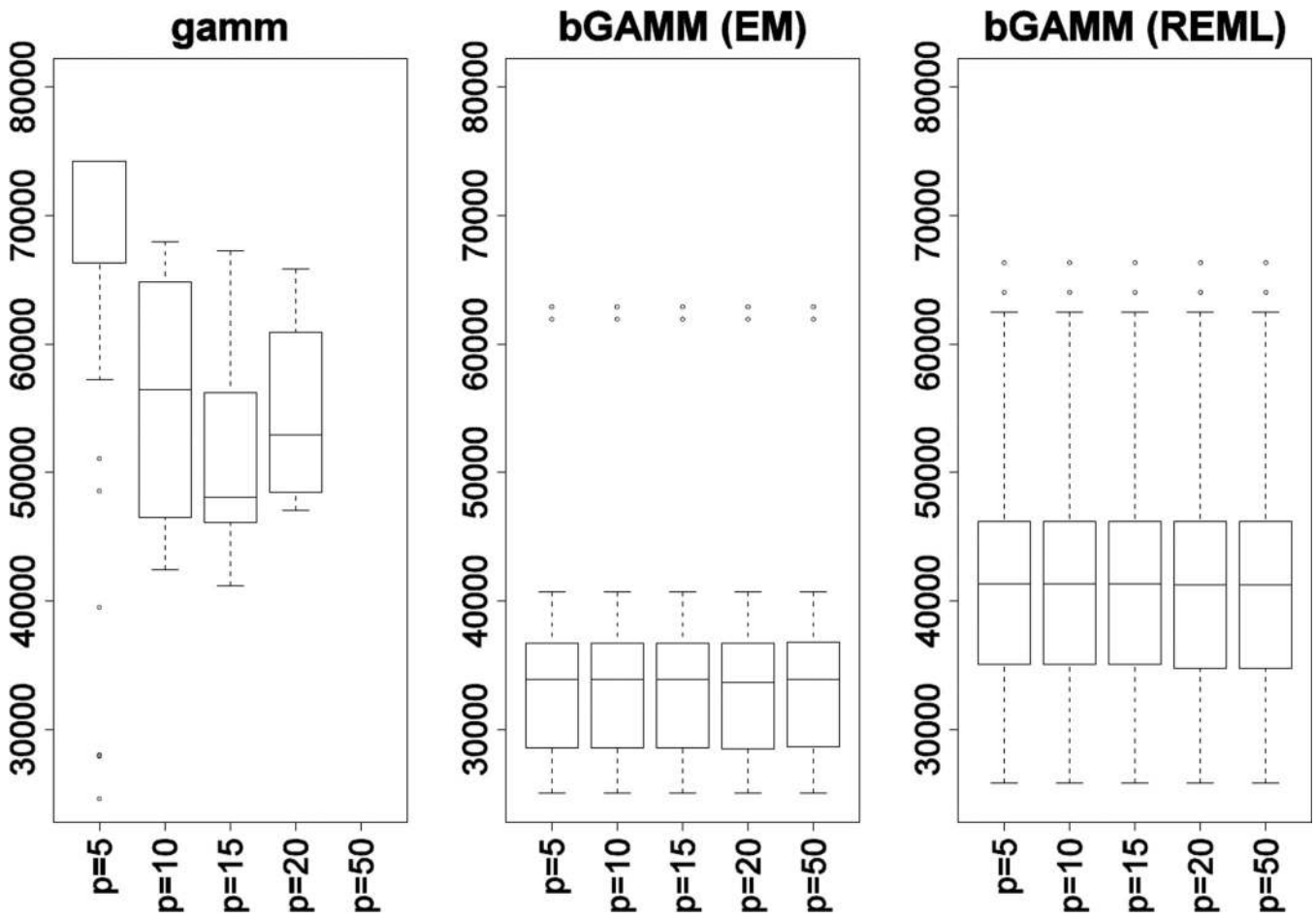



**Figure 1.**
Boxplots of $mse_f$ for `gamm`* (left), `bGAMM` EM (middle) and `bGAMM` REML (right) for $m = 5$, 10, 15, 20, 50 and $\sigma_b = 0.4$ (* only those cases, where `gamm` did converge).

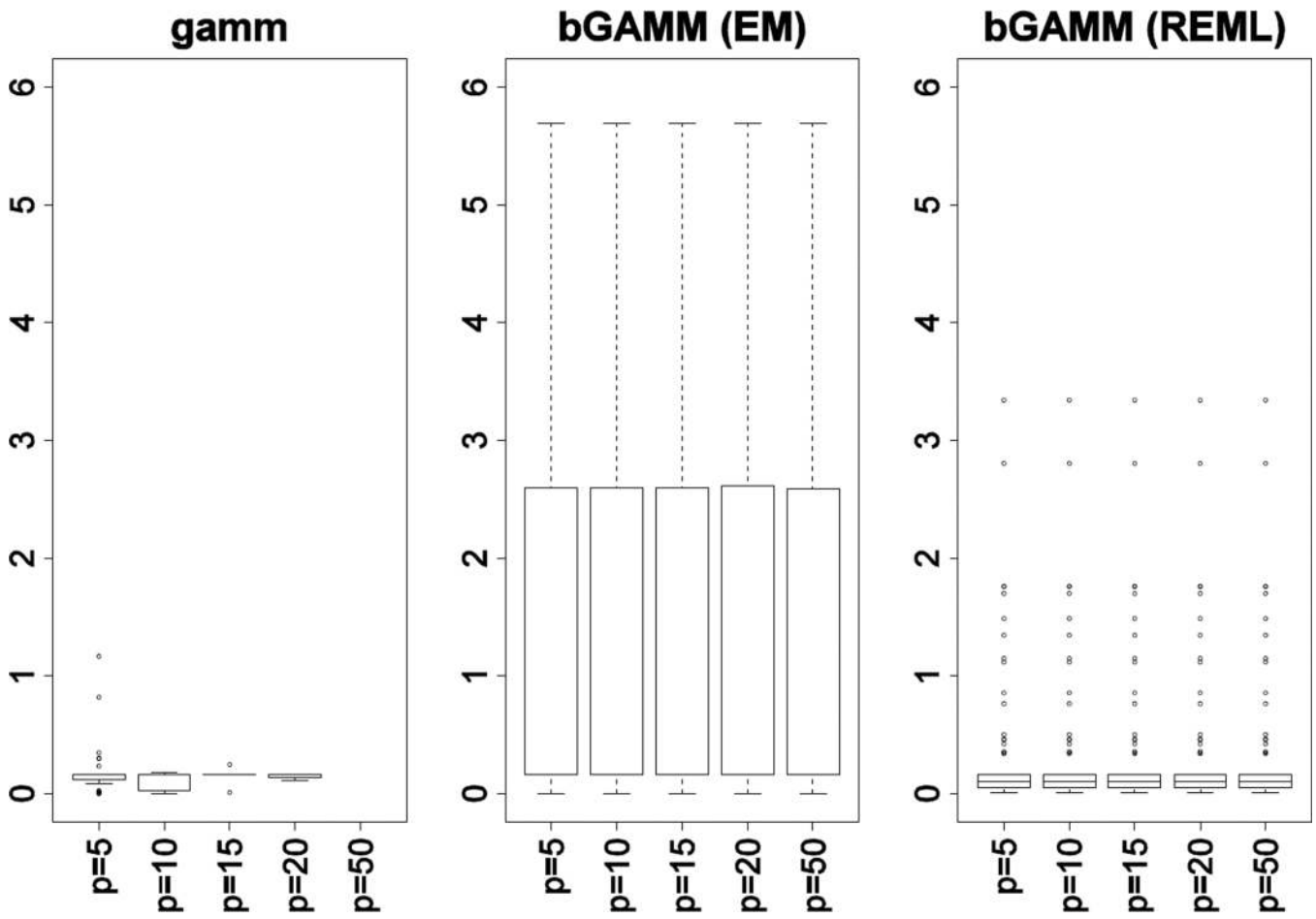
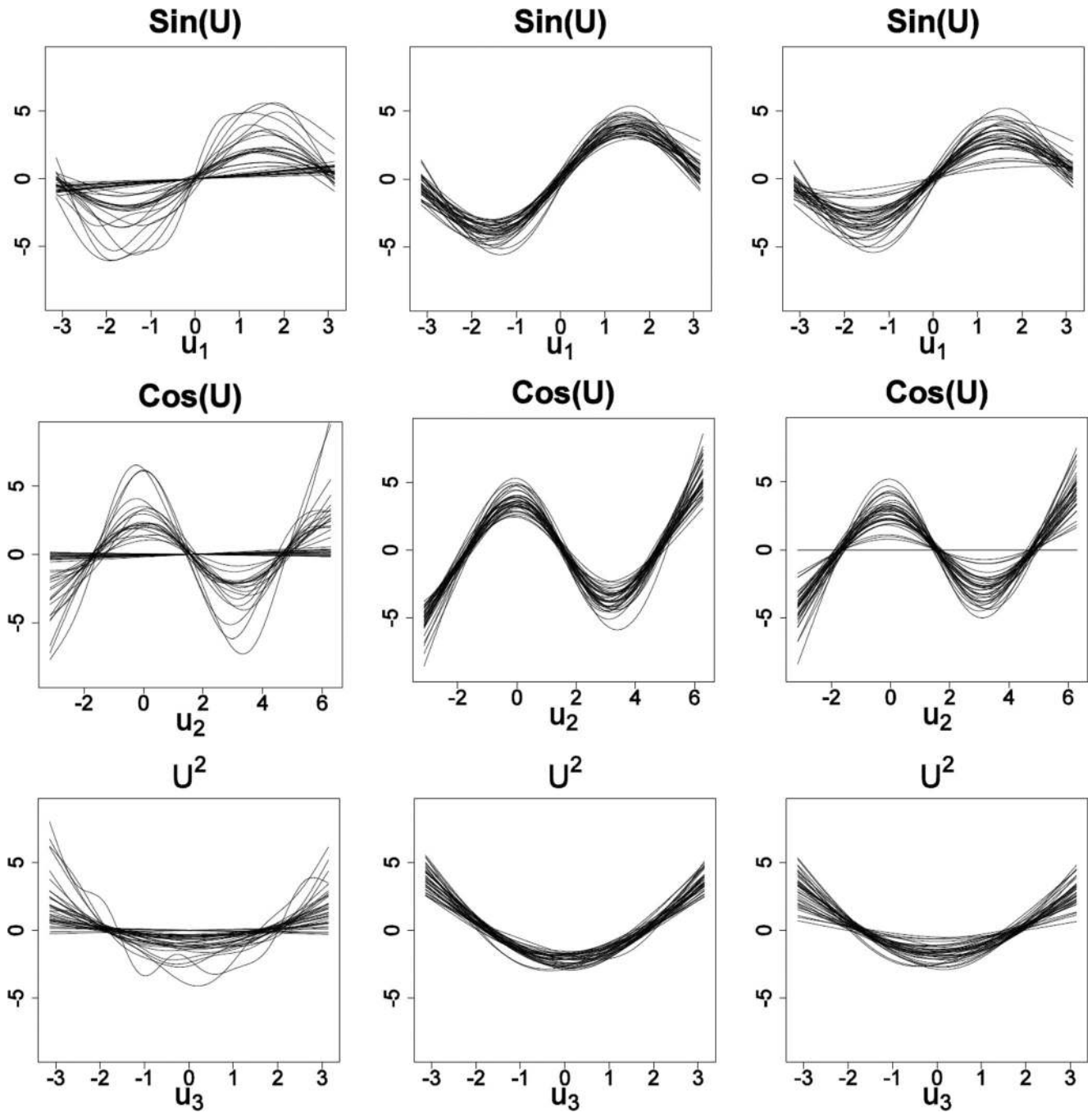



**Figure 2.**
Boxplots of $\text{mse}_\sigma$ for `gamm`* (left), `bGAMM` EM (middle) and `bGAMM` REML (right) for and $m$ = 5, 10, 15, 20, 50 and $\sigma_b = 0.4$(* only those cases, where `gamm` did converge).



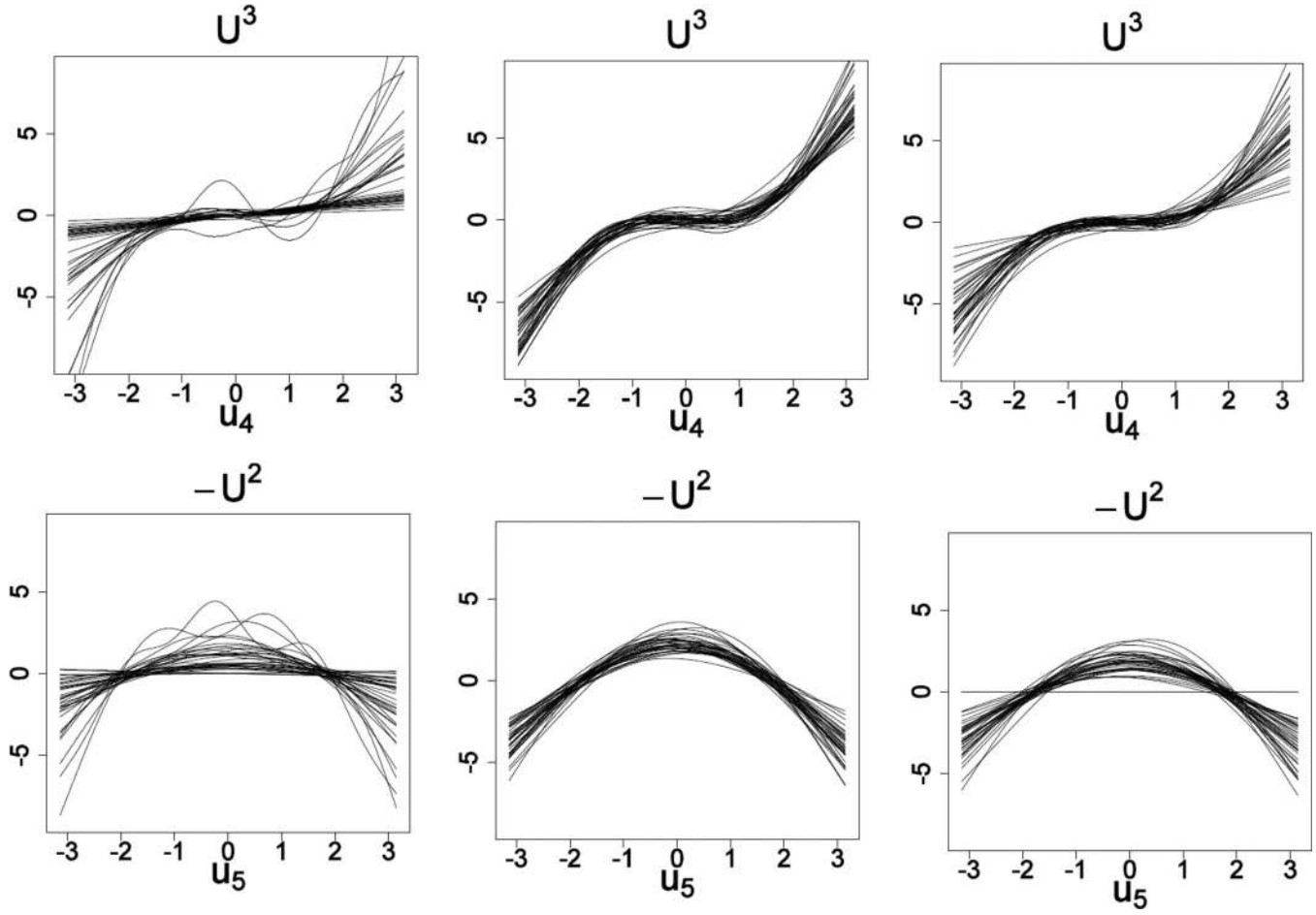



**Figure 3.**
Smooth functions computed with the `gamm` model (left), the `bGAMM` EM model (middle) and `bGAMM` the REML model (right) for $m = 5,\ \sigma_b = 0.4$.

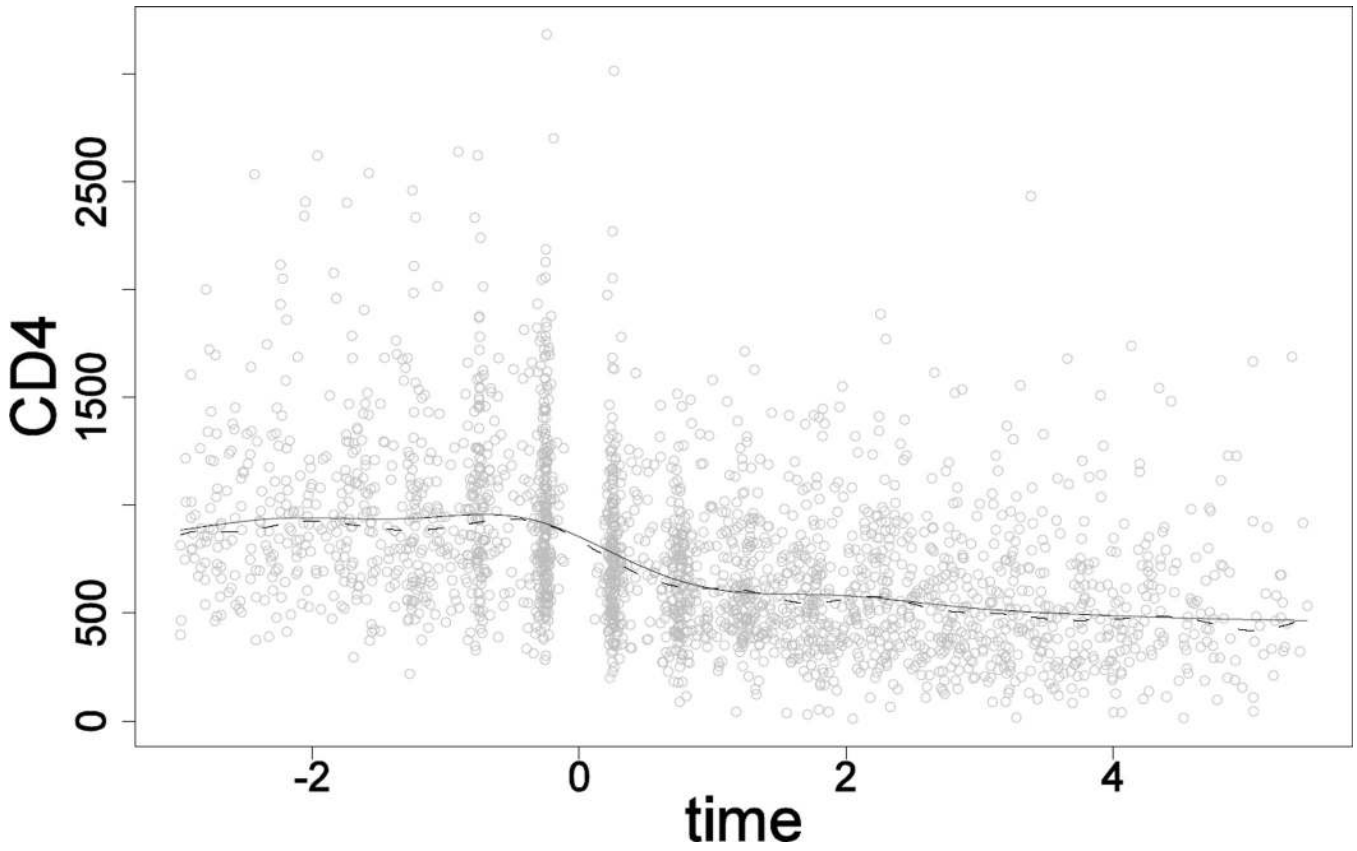



**Figure 4.**
Smoothed time effect (CD4 number of cells *versus* time) from MACS for `gamm` (solid line) and `bGAMM` (dashed line, EM version).

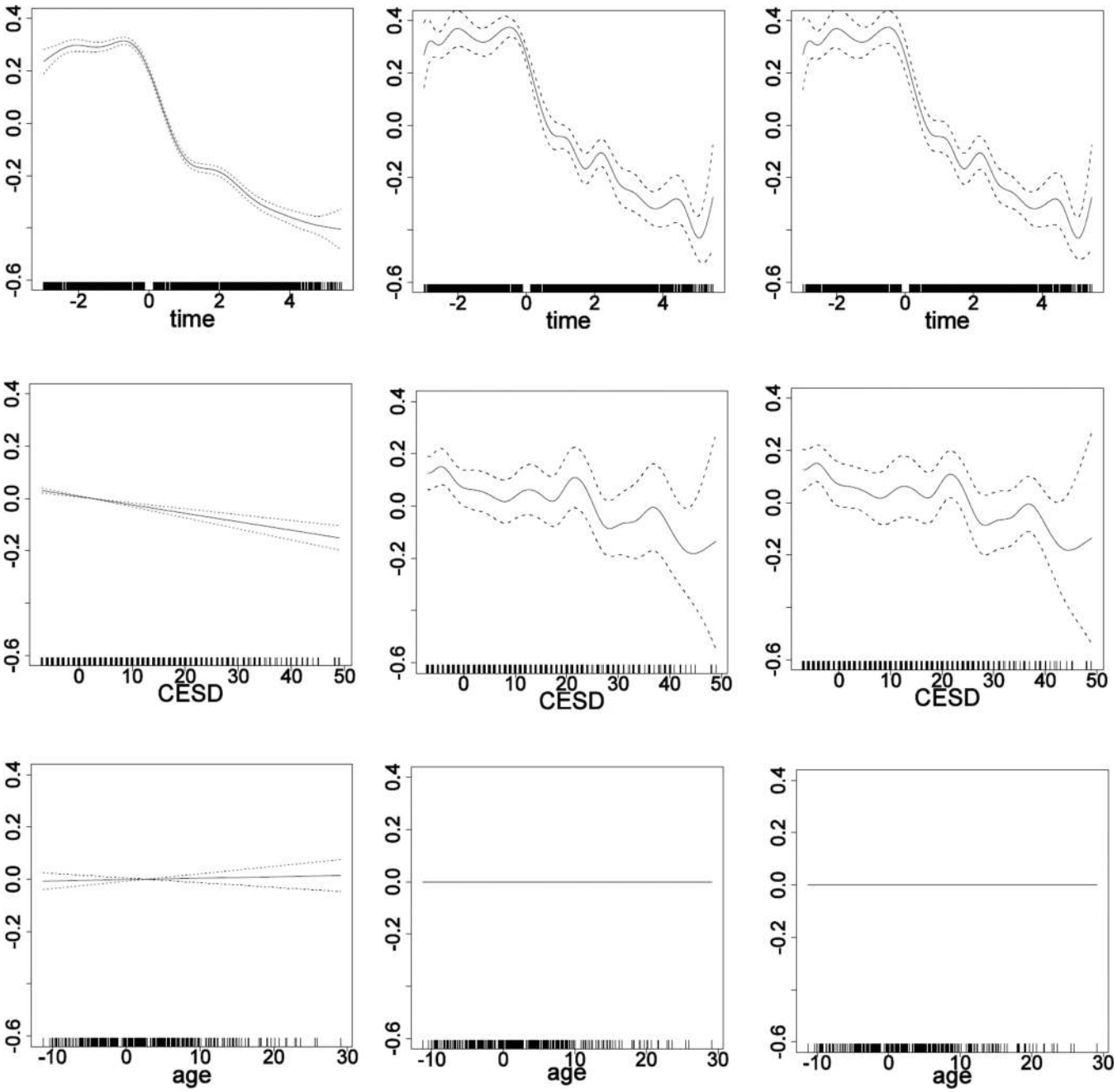



**Figure 4.**
Estimated smooth effect of time, CESD and age computed with the `gamm` model (left), the `bGAMM` EM model (middle) and the `bGAMM` REML model (right) for CD4 data.

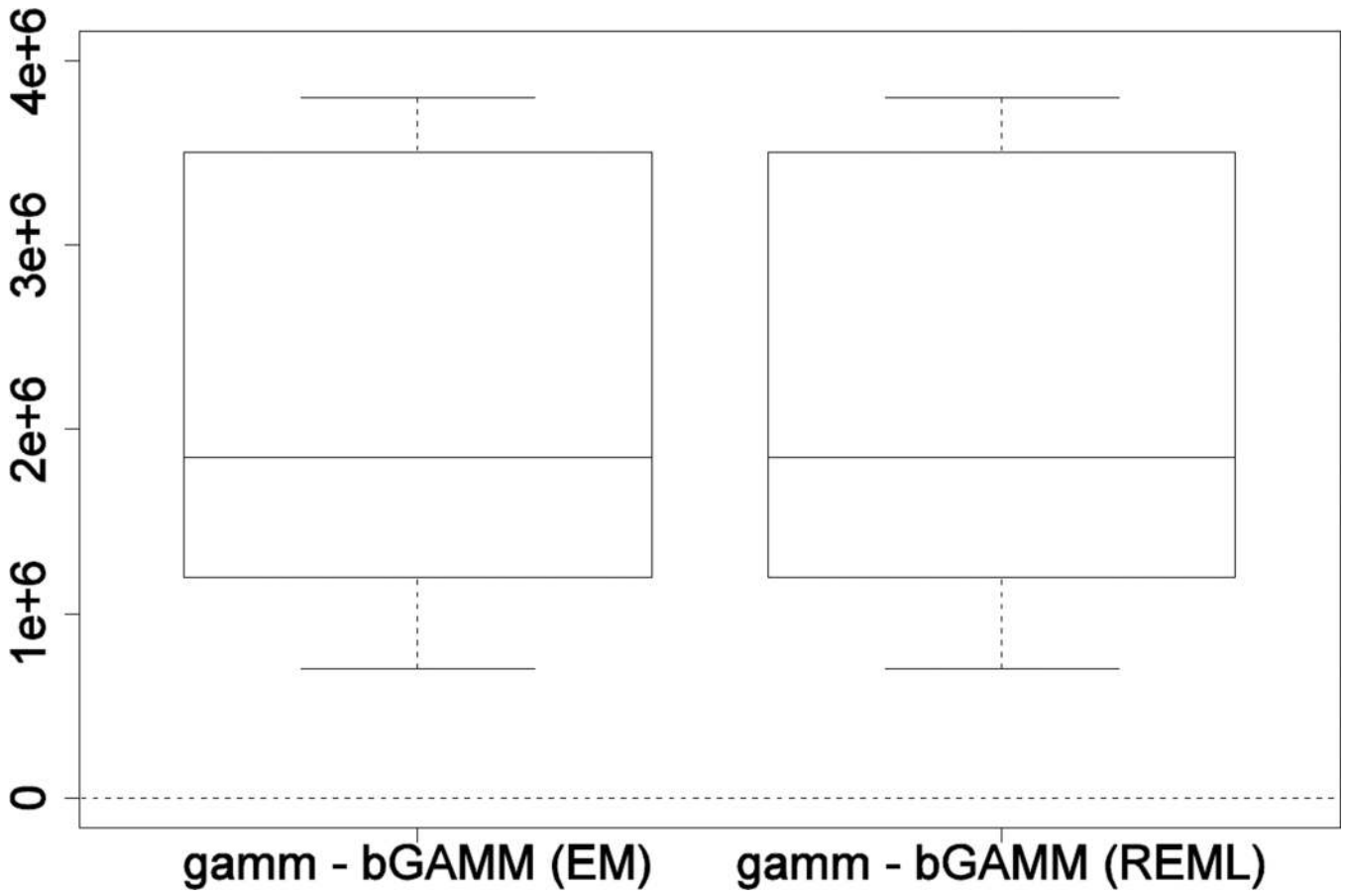



**Figure 5.**
Boxplots of $mse_{pred}$ (gamm)-$mse_{pred}(\cdot)$; 50 random splits with $n_{test} = 50$, $n_{min} = 5$.

**Table 1**

GAMM with `gamm` and boosting (`bGAMM`) on Bernoulli data

| | | gamm | | | bGAMM (EM) | | | | bGAMM (REML) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\sigma_b$ | $m$ | $\text{mse}_f$ | $\text{mse}_{\sigma_b}$ | n.c. | $\text{mse}_f$ | $\text{mse}_{\sigma_b}$ | f.p. | f.n. | $\text{mse}_f$ | $\text{mse}_{\sigma_b}$ | 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 2**

GAMM with boosting (bGAMM) on Bernoulli data averaged over all 100

| | | bGAMM (EM) | | bGAMM (REML) | |
|---|---|---|---|---|---|
| $\sigma_b$ | $m$ | $\text{mse}_f$ | $\text{mse}_{\sigma_b}$ | $\text{mse}_f$ | $\text{mse}_{\sigma_b}$ |
| 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 3**

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

| | `gamm` | `bGAMM` (EM) | `bGAMM` (REML) |
|---|---|---|---|
| intercept | 6.485 (0.026) | 6.470 | 6.470 |
| drugs | 0.034 (0.023) | 0.010 | 0.010 |
| 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 | 0.344 |
| $\hat{\phi}$ | 69.929 | 69.378 | 69.378 |