Stat Comput (2014) 24:137–154
DOI 10.1007/s11222-012-9359-z

Variable selection for generalized linear mixed models by L_1-penalized estimation

Andreas Groll · Gerhard Tutz

Received: 20 December 2011 / Accepted: 28 September 2012 / Published online: 18 October 2012

Abstract Generalized linear mixed models are a widely used tool for modeling longitudinal data. However, their use is typically restricted to few covariates, because the presence of many predictors yields unstable estimates. The presented approach to the fitting of generalized linear mixed models includes an L_1-penalty term that enforces variable selection and shrinkage simultaneously. A gradient ascent algorithm is proposed that allows to maximize the penalized log-likelihood yielding models with reduced complexity. In contrast to common procedures it can be used in high-dimensional settings where a large number of potentially influential explanatory variables is available. The method is investigated in simulation studies and illustrated by use of real data sets.

Keywords Generalized linear mixed model · Lasso · Gradient ascent · Penalty · Linear models · Variable selection

1 Introduction

Generalized linear mixed models (GLMMs) are widely used to model correlated and clustered responses. Various estimation methods have been proposed ranging from numerical integration techniques (for example Booth and Hobert 1999) over "joint maximization methods" (Breslow and Clayton 1993; Schall 1991), in which parameters and random effects are estimated simultaneously, to fully Bayesian approaches (Fahrmeir and Lang 2001). Overviews on current methods are found in McCulloch et al. (2008). Due to the heavy computational problems in GLMMs modeling usually is restricted to few predictor variables. When many predictors are available, estimates become very unstable. Therefore, procedures to select the relevant variables are important in modeling. Classical approaches to the selection of predictors are based on test statistics with the usual stability problems of forward-backward selection procedures, which are due to the inherent discreteness of the method (for example Breiman 1996).

A more timely approach to variable selection is based on boosting methods, which have originally been developed within the machine learning community as a method to improve classification. A first breakthrough was the AdaBoost algorithm proposed by Freund and Schapire (1996). Breiman (1998) considered the AdaBoost algorithm as a gradient descent optimization technique and Friedman (2001) extended boosting methods to include regression problems. Bühlmann and Yu (2003) showed how to fit smoothing splines by boosting base learners and introduced the concept of componentwise boosting, which may be exploited to select predictors. For a detailed overview of componentwise boosting, see Bühlmann and Yu (2003) and Bühlmann and Hothorn (2007). For linear mixed models the incorporation of random effects has been considered by Tutz and Reithinger (2007), first attempts to fit univariate GLMMs were proposed by Tutz and Groll (2010).

An alternative approach to variable selection that has received much attention is based on penalized regression techniques. The Lasso proposed by Tibshirani (1996) has be-

A. Groll (✉)
Department of Mathematics, Ludwig-Maximilians-University Munich, Theresienstr. 39, 80333 Munich, Germany
e-mail: groll@math.lmu.de

G. Tutz
Institute for Statistics, Seminar for Applied Stochastics, Ludwig-Maximilians-University Munich, Akademiestr. 1, 80799 Munich, Germany
e-mail: gerhard.tutz@stat.uni-muenchen.de

come a very popular approach to regression that uses an L_1-penalty on the regression coefficients. This has the effect that all coefficients are shrunken towards zero and some are set exactly to zero. The basic idea is to maximize the log-likelihood $l(\boldsymbol{\beta})$ of the model while constraining the L_1-norm of the parameter vector $\boldsymbol{\beta}$. Thus one obtains the Lasso estimate

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}} l(\boldsymbol{\beta}), \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_1 \leq s, \tag{1}$$

with $s \geq 0$ and with $\|\cdot\|_1$ denoting the L_1-norm. Equivalently, the Lasso estimate $\hat{\boldsymbol{\beta}}$ can be derived by solving the optimization problem

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}} \big[l(\boldsymbol{\beta}) - \lambda \|\boldsymbol{\beta}\|_1 \big], \tag{2}$$

with $\lambda \geq 0$. Both s and λ are tuning parameters that have to be determined, for example by information criteria or cross-validation. This can be very time-consuming, especially in high-dimensional data settings. Thus, for getting computation time under control, in general problems involving a complex log-likelihood, efficient algorithms are needed to derive the solutions of (1) or (2).

For *linear* models the optimization problem of the Lasso can be solved by quadratic programming (Tibshirani 1996), whereas Osborne et al. (2000) recommend an algorithm considering simultaneously the primal problem and its dual, which is highly efficient and is also applicable in high-dimensional cases. A substantial progress was achieved by the LARS algorithm (Efron et al. 2004), which simultaneously produces the set of Lasso fits for all values of the tuning parameters by following the exact, piecewise linear solution path of $\boldsymbol{\beta}$ as a function of s or λ, respectively, and which also inspired the regularization path algorithm for the support vector machine (Hastie et al. 2004). In the last decade several improvements have been designed for the Lasso, e.g. the adaptive Lasso (Zou and Hastie 2006), SCAD (Fan and Li 2001), the Elastic Net (Zou and Hastie 2005), the Dantzig selector (Candes and Tao 2007), the Double Dantzig (James and Radchenko 2009) and the VISA (Radchenko and James 2008).

The Lasso has been extended to more general models, for example Tibshirani (1997) proposed a new method to perform variable selection in the Cox model. He minimizes the partial log-likelihood subject to the L_1-norm of the parameters being bounded by a constant, which is done by an iterative two-step estimation scheme, using reweighted least squares and adaption to the constraint alternately through a quadratic programming procedure. This procedure was improved by Gui and Li (2005), who suggested an iteratively reweighted estimation approach based on the LARS algorithm, called the LARS-Cox procedure. But according to Segal (2006) and Goeman (2010) both algorithms are computational so demanding, that they cannot be used very well in high-dimensional scenarios.

For generalized linear models a flexible and efficient approach is the L_1-regularized path following algorithm by Park and Hastie (2007), who extended the concept of the LARS algorithm (Efron et al. 2004) to generalized linear models. The exact solution coefficients $\hat{\beta}_j$ are computed at particular values of the smoothing parameter λ and then the coefficients are connected in a piecewise linear manner. Another promising approach uses the componentwise gradients, initiating from a starting value $\boldsymbol{\beta}^{(0)}$ and then running through the single coordinates of $\boldsymbol{\beta}$, updating them accordant to the gradient of the penalized likelihood (see e.g. Shevade and Keerthi 2003, Kim and Kim 2004 or Genkin et al. 2007). Recently Goeman (2010) presented another approach based on a combination of gradient ascent optimization with the Newton-Raphson algorithm.

The use of penalization techniques for the selection of variables in mixed models is still in the beginning. For Gaussian mixed models Ni et al. (2010) proposed SCAD penalty techniques, while Wang et al. (2010a) proposed an adaptive mixed Lasso method, which can incorporate a large number of predictors and simultaneously accounts for the population structure. Schelldorfer et al. (2011) developed a L_1-penalized estimation procedure that works for high-dimensional linear mixed-effects models based on maximum likelihood. Bondell et al. (2010) and Wang et al. (2010b) considered the case of joint selection for fixed and random effects in linear models. A wide class of variable selection procedures for GLMMs with a focus on longitudinal data analysis is studied in Yang (2007).

In the following we develop L_1-penalty approaches for the generalized linear mixed model. The method works by combining gradient ascent optimization with the Fisher scoring algorithm and is based on the approach of Goeman (2010). The article is structured as follows. In Sect. 2 we introduce the GLMM. In Sect. 3 we present the gradient ascent algorithm with its computational details and give further information about starting values and computation of tuning parameters. Then the performance of the gradient ascent algorithm is investigated in two simulation studies. Applications are considered in Sect. 4. The presented algorithm is implemented in the `glmmLasso` function of the corresponding R-package (Groll 2011a; publicly available via CRAN, see http://www.r-project.org). Details concerning the determination of the tuning parameter and standard errors as well as the partition of the Fisher matrix are given in the Appendix.

At this point we want to mention the highly relevant paper by Schelldorfer and Bühlmann (2011), which is available on the first author's webpage. It was unknown to us when first versions of this paper were written and we are

grateful to an unknown reviewer who referred to it. The paper is devoted to the same problem and is interesting both from an algorithmic and theoretical perspective. The approach uses an algorithm called GLMMLasso, that is based on a Lasso-type regularization with a cyclic coordinate descent optimization and is implemented in the R-package `glmmlasso`. We will compare our method with their approach (which we denote by `GLMMLasso` (SB) for better distinctness) in simulation studies and refer to differences between the two approaches.

2 Generalized linear mixed models—GLMMs

Let y_{it} denote observation t in cluster i, $i = 1, \ldots, n$, $t = 1, \ldots, T_i$, collected in $\mathbf{y}_i^T = (y_{i1}, \ldots, y_{iT_i})$. Furthermore, let $\mathbf{x}_{it}^T = (1, x_{it1}, \ldots, x_{itp})$ be the covariate vector associated with fixed effects and $\mathbf{z}_{it}^T = (z_{it1}, \ldots, z_{itq})$ be 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 $var(y_{it}|\mathbf{b}_i) = \phi\upsilon(\mu_{it})$, where $\upsilon(.)$ is a known variance function and ϕ is a scale parameter. The GLMM that we consider in the following has the form

$$g(\mu_{it}) = \mathbf{x}_{it}^T\boldsymbol{\beta} + \mathbf{z}_{it}^T\mathbf{b}_i = \eta_{it}^{\text{par}} + \eta_{it}^{\text{rand}}, \tag{3}$$

where g is a monotonic and continuously 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 intercept and $\eta_{it}^{\text{rand}} = \mathbf{z}_{it}^T\mathbf{b}_i$ contains the cluster-specific random effects $\mathbf{b}_i \sim N(\mathbf{0}, \mathbf{Q})$, with $q \times q$ covariance matrix $\mathbf{Q}$. An alternative form that we also use is

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

where $h = g^{-1}$ is the inverse link function.

A closed representation of model (3) is obtained by using matrix notation. By collecting observations within one cluster, the model has the form

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

where $\mathbf{X}_i^T = (\mathbf{x}_{i1}, \ldots, \mathbf{x}_{iT_i})$ denotes the design matrix of the i-th cluster and $\mathbf{Z}_i^T = (\mathbf{z}_{i1}, \ldots, \mathbf{z}_{iT_i})$. For all observations one obtains

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

with $\mathbf{X}^T = [\mathbf{X}_1^T, \ldots, \mathbf{X}_n^T]$ and block-diagonal matrix $\mathbf{Z} = \text{diag}(\mathbf{Z}_1, \ldots, \mathbf{Z}_n)$. For the random effects vector $\mathbf{b}^T = (\mathbf{b}_1^T, \ldots, \mathbf{b}_n^T)$ one has a normal distribution with block-diagonal covariance matrix $\mathbf{Q}_\mathbf{b} = \text{diag}(\mathbf{Q}, \ldots, \mathbf{Q})$.

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

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

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 (compare Fahrmeir and Tutz 2001).

One popular method to maximize GLMMs is penalized quasi-likelihood (PQL), which has been suggested by Breslow and Clayton (1993), Lin and Breslow (1996) and Breslow and Lin (1995). Typically the covariance matrix $\mathbf{Q}(\boldsymbol{\varrho})$ of the random effects $\mathbf{b}_i$ depends on an unknown parameter vector $\boldsymbol{\varrho}$. In penalization-based concepts the joint likelihood-function is specified by the parameter vector of the covariance structure $\boldsymbol{\varrho}$ together with the dispersion parameter ϕ, which are collected in $\boldsymbol{\gamma}^T = (\phi, \boldsymbol{\varrho}^T)$, and parameter vector $\boldsymbol{\delta}^T = (\boldsymbol{\beta}^T, \mathbf{b}^T)$. The corresponding log-likelihood is

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

where $p(\mathbf{b}_i, \boldsymbol{\gamma})$ denotes the density of the random effects. Breslow and Clayton (1993) derived the approximation

$$l^{\text{app}}(\boldsymbol{\delta}, \boldsymbol{\gamma}) = \sum_{i=1}^{n} \log\big(f(\mathbf{y}_i|\boldsymbol{\delta}, \boldsymbol{\gamma})\big) - \frac{1}{2}\mathbf{b}^T\mathbf{Q}(\boldsymbol{\varrho})^{-1}\mathbf{b}, \tag{4}$$

where the penalty term $\mathbf{b}^T\mathbf{Q}(\boldsymbol{\varrho})^{-1}\mathbf{b}$ is due to the approximation based on the Laplace method.

PQL usually works within the profile likelihood concept. It is distinguished between the estimation of $\boldsymbol{\delta}$, given the plugged-in estimate $\hat{\boldsymbol{\gamma}}$, resulting in the profile-likelihood $l^{\text{app}}(\boldsymbol{\delta}, \hat{\boldsymbol{\gamma}})$, and the estimation of $\boldsymbol{\gamma}$. The PQL method is implemented in the macro `GLIMMIX` and proc `GLMMIX` in SAS (Wolfinger 1994), as well as in the `glmmPQL` and `gamm` functions of the R-packages `MASS` (Venables and Ripley 2002) and `mgcv` (Wood 2006). Further notes were given by Wolfinger and O'Connell (1993), Littell et al. (1996) and Vonesh (1996).

3 Regularization in GLMMs

In the following the log-likelihood (4) is expanded to include the penalty term $\lambda\sum_{i=1}^{p}|\beta_i|$. Approximation along the lines of Breslow and Clayton (1993) yields the penalized log-likelihood

$$l^{\text{pen}}(\boldsymbol{\beta}, \mathbf{b}, \boldsymbol{\gamma}) = l^{\text{pen}}(\boldsymbol{\delta}, \boldsymbol{\gamma}) = l^{\text{app}}(\boldsymbol{\delta}, \boldsymbol{\gamma}) - \lambda\sum_{i=1}^{p}|\beta_i|. \tag{5}$$

For given $\hat{\boldsymbol{\gamma}}$ the optimization problem reduces to

$$\hat{\boldsymbol{\delta}} = \underset{\boldsymbol{\delta}}{\operatorname{argmax}}\, l^{\text{pen}}(\boldsymbol{\delta}, \hat{\boldsymbol{\gamma}}) = \underset{\boldsymbol{\delta}}{\operatorname{argmax}} \left[l^{\text{app}}(\boldsymbol{\delta}, \hat{\boldsymbol{\gamma}}) - \lambda \sum_{i=1}^{p} |\beta_i| \right]. \tag{6}$$

We will use a full gradient algorithm that is based on the algorithm of Goeman (2010). As Goeman (2010) already pointed out, the algorithm can easily be amended to situations in which some parameters should not be penalized. In this case the penalty term from the optimization problem of equation (2) is replaced by $\sum_{i=1}^{p} \lambda_i |\beta_i|$, where $\lambda_i = 0$ is chosen for unpenalized parameters. The penalty used in (5) and (6) can be seen as a partially penalized approach if the whole parameter vector $\boldsymbol{\delta}^T = (\boldsymbol{\beta}^T, \mathbf{b}^T)$ is considered.

3.1 Gradient ascent algorithm—glmmLasso

In the following an algorithm is presented for maximizing the penalized log-likelihood $l^{\text{pen}}(\boldsymbol{\delta}, \boldsymbol{\gamma})$ from (5). In contrast to the approaches of Shevade and Keerthi (2003), Kim and Kim (2004) and Genkin et al. (2007), where only a single component is updated at a time, it follows the gradient of the likelihood from a given starting value of $\boldsymbol{\delta}$ and uses the full penalized gradient at each step.

Note, that due to the penalty term in (5) the penalized log-likelihood l^{pen} is not differentiable everywhere. However, for every point $\boldsymbol{\delta}$ and every direction $\mathbf{v} \in \mathbb{R}^{p+nq}$ the directional derivative can be defined as

$$l'_{\text{pen}}(\boldsymbol{\delta}; \mathbf{v}, \boldsymbol{\gamma}) = \lim_{t \downarrow 0} \frac{1}{t} \big(l^{\text{pen}}(\boldsymbol{\delta} + t\mathbf{v}, \boldsymbol{\gamma}) - l^{\text{pen}}(\boldsymbol{\delta}, \boldsymbol{\gamma}) \big).$$

The gradient ascent algorithm uses a series of Taylor approximations and approximates at each step the penalized log-likelihood l^{pen} from (5) locally from a current estimate $\hat{\boldsymbol{\delta}}$ in direction of the gradient by a directional second order Taylor approximation

$$\begin{aligned} l^{\text{pen}}\big(\hat{\boldsymbol{\delta}} + t\mathbf{s}^{\text{pen}}(\hat{\boldsymbol{\delta}}, \boldsymbol{\gamma}), \boldsymbol{\gamma}\big) \approx\ & l^{\text{pen}}(\hat{\boldsymbol{\delta}}, \boldsymbol{\gamma}) \\ & + t l'_{\text{pen}}\big(\hat{\boldsymbol{\delta}}; \mathbf{s}^{\text{pen}}(\hat{\boldsymbol{\delta}}, \boldsymbol{\gamma}), \boldsymbol{\gamma}\big) \\ & + 0.5 t^2 l''_{\text{pen}}\big(\hat{\boldsymbol{\delta}}; \mathbf{s}^{\text{pen}}(\hat{\boldsymbol{\delta}}, \boldsymbol{\gamma})\big), \end{aligned} \tag{7}$$

with $t > 0$ and where $\mathbf{s}^{\text{pen}}(\cdot, \cdot)$ and $l''_{\text{pen}}(\cdot; \cdot)$ are defined in step 2 (a) and (b) in the following algorithm (compare Goeman 2010).

A central issue is to find the correct step size t for the update. In the following, the optimal step size resulting from Taylor approximation is denoted by t_{opt}. But in order to guarantee differentiability of the log-likelihood, the step size of the update needs to be constrained, such that no discontinuity points of the gradient are crossed, while going into the direction of the gradient. This could happen, if single components of the gradient have opposite directions as their corresponding estimates. Hence, an adequate upper bound t_{edge} for the step size is derived in the update step 2. (d) of the algorithm.

Similar to Goeman (2010) the algorithm can automatically switch to a Fisher scoring procedure when it gets close to the optimum and therefore avoids the tendency to slow convergence which is typical for gradient ascent algorithms. An additional step is needed to estimate the variance-covariance components $\mathbf{Q}$ of the random effects. To keep the notation simple, we omit the argument $\boldsymbol{\gamma}$ in the following description of the algorithm and write $l^{\text{app}}(\boldsymbol{\delta})$ instead of $l^{\text{app}}(\boldsymbol{\delta}, \boldsymbol{\gamma})$.

Algorithm `glmmLasso`

1. *Initialization*
   Compute starting values $\hat{\boldsymbol{\beta}}^{(0)}, \hat{\mathbf{b}}^{(0)}, \hat{\boldsymbol{\gamma}}^{(0)}$ (see Sect. 3.2.1); $\hat{\boldsymbol{\eta}}^{(0)} = \mathbf{X}\hat{\boldsymbol{\beta}}^{(0)} + \mathbf{Z}\hat{\mathbf{b}}^{(0)}$.
2. *Iteration*
   For $l = 1, 2, \ldots$ until convergence:
   (a) *Calculation of the log-likelihood gradient for given* $\hat{\boldsymbol{\gamma}}^{(l-1)}$*:*
   With $\mathbf{s}(\boldsymbol{\delta}) = \partial l^{\text{app}}(\boldsymbol{\delta}) / \partial \boldsymbol{\delta}$ derive:

$$s_0^{\text{pen}}\big(\hat{\boldsymbol{\delta}}^{(l-1)}\big) = s_0\big(\hat{\boldsymbol{\delta}}^{(l-1)}\big),$$

$$s_i^{\text{pen}}\big(\hat{\boldsymbol{\delta}}^{(l-1)}\big) = s_i\big(\hat{\boldsymbol{\delta}}^{(l-1)}\big), \quad i = p+1, \ldots, p+ns.$$

   Furthermore, for $i = 1, \ldots, p$ derive:

$$s_i^{\text{pen}}(\hat{\boldsymbol{\delta}}^{(l-1)}) = \begin{cases} s_i(\hat{\boldsymbol{\delta}}^{(l-1)}) - \lambda \operatorname{sign}(\hat{\beta}_i^{(l-1)}), \\ \quad \text{if } \hat{\beta}_i^{(l-1)} \neq 0, \\ s_i(\hat{\boldsymbol{\delta}}^{(l-1)}) - \lambda \operatorname{sign}(s_i(\hat{\boldsymbol{\delta}}^{(l-1)})), \\ \quad \text{if } \hat{\beta}_i^{(l-1)} = 0 \text{ and } |s_i(\hat{\boldsymbol{\delta}}^{(l-1)})| > \lambda, \\ 0, \quad \text{otherwise,} \end{cases}$$

$$\text{where } \operatorname{sign}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0. \end{cases}$$

   (b) *Calculation of the directional second derivative:*
   Let $\mathbf{A} := [\mathbf{X}, \mathbf{Z}]$ and $\mathbf{K} = \operatorname{diag}(0, \ldots, 0, \mathbf{Q}^{-1}, \ldots, \mathbf{Q}^{-1})$ be a block-diagonal penalty matrix with a diagonal of $p+1$ zeros corresponding to the fixed effects and then n times the matrix $\mathbf{Q}^{-1}$. Then the Fisher matrix is given in closed form as $\mathbf{F}^{\text{pen}}(\boldsymbol{\delta}) = \mathbf{A}^T \mathbf{W}(\boldsymbol{\delta}) \mathbf{A} + \mathbf{K}$, with $\mathbf{W}(\boldsymbol{\delta}) = \mathbf{D}(\boldsymbol{\delta}) \boldsymbol{\Sigma}^{-1}(\boldsymbol{\delta}) \mathbf{D}(\boldsymbol{\delta})^T$ and $\mathbf{D}(\boldsymbol{\delta}) = \partial h(\boldsymbol{\eta}) / \partial \boldsymbol{\eta}$, $\boldsymbol{\Sigma}(\boldsymbol{\delta}) = \operatorname{cov}(\mathbf{y}|\boldsymbol{\delta})$. The directional second derivative is given for every $\boldsymbol{\delta}$ and every direction vector $\mathbf{v} \in \mathbb{R}^{p+1+ns}$ by

$$l''_{\text{pen}}(\boldsymbol{\delta}; \mathbf{v}) = -\mathbf{v}^T \mathbf{F}^{\text{pen}}(\boldsymbol{\delta}) \mathbf{v}$$

(c) *Optimum of Taylor approximation:*
Maximization of the Taylor approximation (7) with respect to t, using $l'_{\text{pen}}(\boldsymbol{\delta}; \mathbf{s}^{\text{pen}}(\boldsymbol{\delta}, \boldsymbol{\gamma}), \boldsymbol{\gamma}) = \|\mathbf{s}^{\text{pen}}(\boldsymbol{\delta})\|_2$, yields

$$t_{\text{opt}}^{(l-1)} = -\frac{\|\mathbf{s}^{\text{pen}}(\hat{\boldsymbol{\delta}}^{(l-1)})\|_2}{l''_{\text{pen}}(\hat{\boldsymbol{\delta}}^{(l-1)}, \mathbf{s}^{\text{pen}}(\hat{\boldsymbol{\delta}}^{(l-1)}))},$$

and

$$t_{\text{edge}}^{(l-1)} = \min_i \left\{ -\frac{\hat{\delta}_i^{(l-1)}}{s_i^{\text{pen}}(\hat{\boldsymbol{\delta}}^{(l-1)})} : \right.$$
$$\left. \text{sign}\big(\hat{\delta}_i^{(l-1)}\big) = -\text{sign}\big[s_i^{\text{pen}}\big(\hat{\boldsymbol{\delta}}^{(l-1)}\big)\big] \neq 0 \right\}$$

with $\|\cdot\|_2$ denoting the L_2 norm.

(d) *Update*

$$\hat{\boldsymbol{\delta}}^{(l)} = \begin{cases} \hat{\boldsymbol{\delta}}^{(l-1)} + t_{\text{edge}}^{(l-1)} \mathbf{s}^{\text{pen}}(\hat{\boldsymbol{\delta}}^{(l-1)}), \\ \quad \text{if } t_{\text{opt}}^{(l-1)} \geq t_{\text{edge}}^{(l-1)}, \\ \hat{\boldsymbol{\delta}}_{\text{FS}}^{(l-1)}, \quad \text{if } t_{\text{opt}}^{(l-1)} < t_{\text{edge}}^{(l-1)} \\ \qquad \text{and } \text{sign}(\hat{\boldsymbol{\delta}}_{\text{FS}}^{(l)}) = \text{sign}(\hat{\boldsymbol{\delta}}^{(l-1)}), \\ \hat{\boldsymbol{\delta}}^{(l-1)} + t_{\text{opt}}^{(l-1)} \mathbf{s}^{\text{pen}}(\hat{\boldsymbol{\delta}}^{(l-1)}), \quad \text{otherwise}, \end{cases}$$

where $\hat{\boldsymbol{\delta}}_{\text{FS}}^{(l)}$ denotes the Fisher scoring estimate as given in Sect. 3.2.2.

a) *Computation of variance-covariance components*
Estimates $\hat{\mathbf{Q}}^{(l)}$ are obtained as approximate EM-type estimates or by alternative methods (see Sect. 3.2.3) yielding the update $\boldsymbol{\varrho}^{(l)}$. If necessary, the whole vector $\hat{\boldsymbol{\gamma}}^{(l)}$ is completed by an estimate of the dispersion parameter.

3. *Re-Estimation*
In a final step a model that includes only the variables corresponding to non-zero parameters of $\hat{\boldsymbol{\beta}}$ is fitted. A simple Fisher scoring, resulting in the final estimates $\hat{\boldsymbol{\delta}}, \hat{\mathbf{Q}}$ is used.

3.2 Computational details of `glmmLasso`

In the following we give a more detailed description of the single steps of the `glmmLasso` algorithm. First details of the computation of starting values are given, then the Fisher scoring step is further explained and finally two estimation techniques for the variance-covariance components are described.

3.2.1 Starting values for `glmmLasso`

We compute the starting values $\hat{\boldsymbol{\beta}}^{(0)}, \hat{\mathbf{b}}^{(0)}, \hat{\mathbf{Q}}^{(0)}$ from step 1 of the `glmmLasso` algorithm by fitting the simple global intercept model with random effects, given by $g(\mu_{it}) = \beta_0 + \mathbf{z}_{it}^T \mathbf{b}_i$. This can be done very easily, for example by using the R-function `glmmPQL` (Wood 2006) from the `MASS` library (Venables and Ripley 2002).

3.2.2 Fisher scoring

Similar to Goeman (2010) we combine gradient ascent optimization with the Fisher scoring algorithm in the update step 2 (d) of the `glmmLasso` algorithm. Although gradient ascent optimization is computationally simple, because no matrix inversion or other computationally expensive calculations are involved, often a large number of steps is required for convergence. By allowing the algorithm to switch to the Fisher scoring algorithm it becomes much faster.

For an arbitrary iteration we define $J = \{j : \beta_j \neq 0, j = 0, 1, \ldots, p\}$, the index set of the "active" covariates, corresponding to the $m = \#J \leq p + 1$ non-zero coefficients. Furthermore, let $\tilde{\boldsymbol{\delta}}^T = (\beta_{J_1}, \ldots, \beta_{J_m}, \mathbf{b}^T)$, and let $\tilde{\mathbf{s}}^{\text{pen}}(\boldsymbol{\delta}) = \{s_{J_1}^{\text{pen}}(\boldsymbol{\delta}), \ldots, s_{J_m}^{\text{pen}}(\boldsymbol{\delta}), s_{p+1}^{\text{pen}}(\boldsymbol{\delta}), \ldots, s_{p+ns}^{\text{pen}}(\boldsymbol{\delta})\}^T$ be the gradient in the constrained domain and $\tilde{\mathbf{F}}^{\text{pen}}$ the $(m + ns) \times (m + ns)$ Fisher matrix of the constrained optimization, given by $\tilde{\mathbf{F}}^{\text{pen}}(\boldsymbol{\delta}) = \mathbf{A}_J^T \mathbf{W}(\boldsymbol{\delta}) \mathbf{A}_J + \mathbf{K}_J$, with $\mathbf{A}_J := [\mathbf{X}_J, \mathbf{Z}]$, whereas $\mathbf{X}_J$ contains only those columns of $\mathbf{X}$ corresponding to J, $\mathbf{K}_J = \text{diag}(0, \ldots, 0, \mathbf{Q}^{-1}, \ldots, \mathbf{Q}^{-1})$ is a block-diagonal penalty matrix with a diagonal of m zeros corresponding to the non-zero fixed effects and then n times the matrix $\mathbf{Q}^{-1}$.

One step of Fisher scoring in the current subdomain takes the form

$$\hat{\tilde{\boldsymbol{\delta}}}^{(l)} = \hat{\tilde{\boldsymbol{\delta}}}^{(l-1)} + \big(\tilde{\mathbf{F}}^{\text{pen}}\big(\hat{\boldsymbol{\delta}}^{(l-1)}\big)\big)^{-1} \tilde{\mathbf{s}}^{\text{pen}}\big(\hat{\boldsymbol{\delta}}^{(l-1)}\big).$$

This estimator can be mapped back to a $(p + 1 + ns)$-vector $\hat{\boldsymbol{\delta}}_{FS}^{(l)}$ by augmenting $\hat{\tilde{\boldsymbol{\delta}}}^{(l)}$ with zeros for all non-active covariates. In order that the Taylor approximation which is underlying such a step of Fisher scoring holds within the current subdomain, $\hat{\boldsymbol{\delta}}_{FS}^{(l)}$ is accepted only when $\text{sign}(\hat{\boldsymbol{\delta}}_{\text{FS}}^{(l)}) = \text{sign}(\hat{\boldsymbol{\delta}}^{(l-1)})$.

As Goeman (2010) pointed out, it is often better to avoid the attempt of trying a Fisher scoring step whenever it is likely to fail, because it can be computational expensive. Practical experience with our `glmmLasso` algorithm has shown the same tendencies. We do not try a Fisher scoring step at $l = 0$ and after a Fisher scoring step has failed we try another step of Fisher scoring not until the active set has changed. Nevertheless the incorporation of Fisher scoring into the procedure can greatly speed up convergence once the algorithm gets close to the optimum.

3.2.3 Variance-covariance components

Variance estimates for the random effects can be derived as an approximate EM algorithm, using the posterior mode estimates and posterior curvatures. One derives $(\mathbf{F}^{\text{pen}}(\hat{\boldsymbol{\delta}}^{(l)}))^{-1}$,

the inverse of the penalized pseudo Fisher matrix, using the posterior mode estimates $\hat{\boldsymbol{\delta}}^{(l)}$ to obtain the posterior curvatures $\hat{\mathbf{V}}_{ii}^{(l)}$. Now compute $\hat{\mathbf{Q}}^{(l)}$ by

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

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

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

where $\mathbf{F}_{\boldsymbol{\beta}\boldsymbol{\beta}}, \mathbf{F}_{i\boldsymbol{\beta}}, \mathbf{F}_{ii}$ are elements of the partitioned Fisher matrix, see Appendix B.

For an alternative estimation of variances (Breslow and Clayton 1993) maximize the profile likelihood that is associated with the normal theory model. By replacing $\boldsymbol{\beta}$ with $\hat{\boldsymbol{\beta}}$ one maximizes

$$l(\mathbf{Q}_\mathbf{b}) = -\frac{1}{2}\log\big(|\mathbf{V}(\hat{\boldsymbol{\delta}})|\big) - \frac{1}{2}\log\big(|\mathbf{X}^T\mathbf{V}^{-1}(\hat{\boldsymbol{\delta}})\mathbf{X}|\big) - \frac{1}{2}\big(\tilde{\boldsymbol{\eta}}(\hat{\boldsymbol{\delta}}) - \mathbf{X}\hat{\boldsymbol{\beta}}\big)^T\mathbf{V}^{-1}(\hat{\boldsymbol{\delta}})\big(\tilde{\boldsymbol{\eta}}(\hat{\boldsymbol{\delta}}) - \mathbf{X}\hat{\boldsymbol{\beta}}\big)$$

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

In our simulation studies and applications the approximate EM-method is used, as in our experience the REML-type approach did not improve on the results but increased the computational costs.

3.3 Incorporation of categorical predictors

A frequently found type of structured regressors are categorical predictors (factors), which are usually dummy-coded and hence result in groups of dummy variables. That means a one-dimensional variable is transformed into a group of variables. By construction, the standard Lasso solution is only able to select distinct dummy variables but not whole factors. Since one wants variable selection, the algorithm has to be modified in the spirit of the group Lasso, which was proposed by Yuan and Lin (2006). It was explicitly designed for the selection of grouped variables in the form of dummy-coded factors in the usual linear regression set-up and represents an elegant combination of penalization within groups of variables and groupwise selection by using a Lasso penalty at the factor level, and a Ridge-type penalization within coefficient groups.

Meier et al. (2008) have extended the group Lasso to logistic regression and present an efficient algorithm to solve the corresponding convex optimization problem. Their resulting logistic group Lasso estimator is obtained by replacing the Lasso penalty term from (2) by the penalty $\sum_{g=1}^{G}\lambda_g\|\boldsymbol{\beta}_{I_g}\|_2$, where I_g denotes the index set of to the g-th group of variables, $g = 1, \ldots, G$ and $\lambda_g = \lambda\sqrt{\mathrm{df}_g}$, with df_g representing the number of parameters of group g, which is equal to the number of factor levels minus one for categorical predictors and df_g=1 for continuous predictors.

Suppose that the $p+1$ columns of our design matrix $\mathbf{X}$ are now resulting from G predictors, which may be categorical or continuous, plus intercept. Using the same notations as above, we incorporate the penalization adjustment of Meier et al. (2008) into the `glmmLasso` algorithm by simply modifying step 2 (a) in the following way:

(a2) *Calculation of the log-likelihood gradient*
With $\mathbf{s}(\boldsymbol{\delta}) = \partial l^{\mathrm{app}}(\boldsymbol{\delta})/\partial\boldsymbol{\delta}$ derive:

$$s_0^{\mathrm{pen}}(\hat{\boldsymbol{\delta}}^{(l-1)}) = s_0(\hat{\boldsymbol{\delta}}^{(l-1)}),$$
$$s_i^{\mathrm{pen}}(\hat{\boldsymbol{\delta}}^{(l-1)}) = s_i(\hat{\boldsymbol{\delta}}^{(l-1)}), \quad i = p+1, \ldots, p+ns.$$

Furthermore, for $g = 1, \ldots, G$ derive:

$$\mathbf{s}_{I_g}^{\mathrm{pen}}(\hat{\boldsymbol{\delta}}^{(l-1)}) = \begin{cases} \mathbf{s}_{I_g}(\hat{\boldsymbol{\delta}}^{(l-1)}) - \lambda_g\dfrac{\hat{\boldsymbol{\beta}}_{I_g}^{(l-1)}}{\|\hat{\boldsymbol{\beta}}_{I_g}^{(l-1)}\|_2}, \\ \quad \text{if } \|\hat{\boldsymbol{\beta}}_{I_g}^{(l-1)}\|_2 \neq 0, \\ \mathbf{s}_{I_g}(\hat{\boldsymbol{\delta}}^{(l-1)}) - \lambda_g\dfrac{\mathbf{s}_{I_g}(\hat{\boldsymbol{\delta}}^{(l-1)})}{\|\mathbf{s}_{I_g}(\hat{\boldsymbol{\delta}}^{(l-1)})\|_2}, \\ \quad \text{if } \|\hat{\boldsymbol{\beta}}_{I_g}^{(l-1)}\|_2 = 0 \\ \quad \text{and } \|\mathbf{s}_{I_g}(\hat{\boldsymbol{\delta}}^{(l-1)})\|_2 > \lambda_g, \\ \mathbf{0}, \quad \text{otherwise.} \end{cases}$$

3.4 Simulation study

In the following small simulation study the performance of the `glmmLasso` algorithm is compared to alternative approaches.

Poisson Link The underlying model is the random intercept Poisson model

$$\eta_{it} = \sum_{j=1}^{p} x_{itj}\beta_j + b_i, \quad i = 1, \ldots, 40,\ t = 1, \ldots, 10,$$
$$E[y_{it}] = \exp(\eta_{it}) := \lambda_{it}, \quad y_{it} \sim \mathrm{Pois}(\lambda_{it}),$$

with linear effects given by $\beta_1 = -4, \beta_2 = -6, \beta_3 = 10$ and $\beta_j = 0, j \geq 4$. We chose the different settings $p = 3, 10, 100, 200, 500$. For $j = 1, \ldots, p$ the vectors $\mathbf{x}_{it}^T = (x_{it1}, \ldots, x_{itp})$ follow a uniform distribution within the interval $[-0.14, 0.14]$, where the parameter vector $\boldsymbol{\beta}$ and the

interval boundaries of the uniform distributions of the covariates have been chosen such that reasonable and manageable response values are obtained. The number of observations was determined by $n = 40$, $T_i := T = 10$, $i = 1, \ldots, n$. The random effect and the noise variable have been specified by $b_i \sim N(0, \sigma_b^2)$ with $\sigma_b = 0.4, 0.8, 1.6$.

In general, the R-functions `glmmPQL` (Venables and Ripley 2002), `glmmML` (Broström 2009) and `glmer` (Bates and Maechler 2010) are able to fit the underlying model. The `glmmPQL` routine is supplied by the `MASS` library. It operates by iteratively calling the R-function `lme` from the `nlme` library and returns the fitted `lme` model object for the working model at convergence. For more details about the `lme` function, see Pinheiro and Bates (2000). The `glmmML` function is supplied with the `glmmML` package (Broström 2009) and features two different methods of approximating the integrals in the log-likelihood function, Laplace and Gauss-Hermite, whereas for the first method the results coincide with the results of the `glmmPQL` routine. Unfortunately, for both functions no model testing methods are available, thus no subset selection procedures can be performed.

However, the `glmer` function from the `lme4` package (Bates and Maechler 2010) provides model testing based on an analysis of deviance. Hence, we use forward subset selection in order to perform variable selection and compare the results with our `glmmLasso` algorithm. We restrict consideration to forward procedures because forward/backward procedures imply huge computational costs. It should be mentioned that the `glmer` function also features two different methods of approximating the integrals in the log-likelihood function, Laplace and adaptive Gauss-Hermite. We focused on the former and call the corresponding forward selection procedure `glmer-select`.

In addition, we compare the results of our `glmmLasso` algorithm with the results obtained by the `GLMMLasso` (SB) algorithm proposed in Schelldorfer and Bühlmann (2011), which was already mentioned at the end of Sect. 1. For both L_1-regularized approaches the tuning parameter λ has been determined using the Bayesian Information Criterion (BIC), see Appendix A.

Finally, we also compare our results with two boosting functions, `bGLMM` (EM) and `bGLMM` (REML), introduced in Tutz and Groll (2010) and implemented in the R-package `GMMBoost` (see Groll 2011b), which perform variable selection by boosting techniques. They differ in the computation of the covariance matrix components $\mathbf{Q}$ of the random effects. The first one can be derived as an approximate EM algorithm, the second one by maximizing the profile likelihood that is associated with the normal theory model and therefore could be seen as an approximate REML-type estimate.

The performance of estimators was evaluated separately for the structural components and the variance. By averaging across 100 training data sets we consider mean squared errors for $\boldsymbol{\beta}$ and σ_b given by $\text{mse}_{\boldsymbol{\beta}} := \|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\|^2$, $\text{mse}_{\sigma_b} := (\sigma_b - \hat{\sigma}_b)^2$. The means of both quantities are presented in Table 1 and 2, together with the corresponding standard errors in brackets. The results of $\text{mse}_{\boldsymbol{\beta}}$ are illustrated in Fig. 1, which shows boxplots of the ratios $\log(\text{mse}_{\boldsymbol{\beta}}(\cdot)/\text{mse}_{\boldsymbol{\beta}}(\texttt{glmer-select}))$ for the different methods, for different numbers of noise variables and the scenario $\sigma_b = 0.4$. Additionally, we present boxplots of the ratios $\log(\text{mse}_{\sigma_b}(\cdot)/\text{mse}_{\sigma_b}(\texttt{glmer-select}))$ corresponding to $\sigma_b = 0.4$ in Fig. 2.

Additional information on the performance of the algorithm was collected in *falseneg* (f.n.), the mean over all 100 simulations of the number of variables β_j, $j = 1, 2, 3$, that were not selected and in *falsepos* (f.p.), the mean over all 100 simulations of the number of variables β_j, $j = 4, \ldots, p$, that were selected. In order to make the different approaches comparable with respect to computational efficiency, we present the computational times (in minutes; the total sum over all 100 simulation runs) for each approach, including the determination of the tuning parameter for the L_1-regularized approaches and including the forward selection procedure for `glmer`. The corresponding results are presented in Table 3.

Results for varying number p of covariates $x_{it1}, \ldots, x_{itp}$ are summarized in Tables 1 to 3. It is seen that simple forward selection with `glmer` performs rather well and dominates all other procedures with the exception of `glmmLasso` in terms of $\text{mse}_{\boldsymbol{\beta}}$. `glmmLasso` performs best, and turns out to be very stable also in high dimensional settings, whereas, in particular the boosting approaches, deteriorate in high dimensional settings. Although the performance of the latter could be improved by adapting tuning parameters, especially by allowing more iterations, computational costs would increase tremendously. In general, the computational expense is drastically growing with the number of covariates for all approaches, but `glmmLasso` turns out to be most efficient in terms of computational time, especially in high-dimensional settings. `glmmLasso` also outperforms the other approaches in terms of false positives. False negatives are for all approaches extremely low. In terms of mse_{σ_b}, `glmer-select`, `GLMMLasso` (SB) and `glmmLasso` show almost identical results and outperform boosting for most scenarios.

An advantage of the L_1-penalization approaches over boosting techniques is that they also perform well when all variables in the predictor are influential as well as in really high dimensional settings. Also with respect to the variance component σ_b, both `glmmLasso` and `GLMMLasso` (SB) slightly outperform both boosting approaches.

Figures 1 and 2 compare the performance of the procedures with `glmer-select` as the reference. They show the varying proportions $\log(\text{mse}_{\boldsymbol{\beta}}(\cdot)/\text{mse}_{\boldsymbol{\beta}}(\texttt{glmer-}$

Table 1 Results for $\text{mse}_{\boldsymbol{\beta}}$ for `glmmLasso` and alternative approaches on Poisson data (standard errors in brackets)

σ_b	p	`glmer-select` $\text{mse}_{\boldsymbol{\beta}}$	`GLMMLasso` (SB) $\text{mse}_{\boldsymbol{\beta}}$	`glmmLasso` $\text{mse}_{\boldsymbol{\beta}}$	`bGLMM` (EM) $\text{mse}_{\boldsymbol{\beta}}$	`bGLMM` (REML) $\text{mse}_{\boldsymbol{\beta}}$
0.4	3	0.92 (0.73)	0.95 (0.79)	0.92 (0.73)	1.56 (1.52)	1.03 (0.85)
0.4	10	1.26 (1.08)	3.63 (2.47)	1.26 (1.08)	2.05 (1.37)	1.84 (1.36)
0.4	100	2.92 (2.22)	8.70 (3.78)	1.80 (1.85)	8.54 (3.79)	8.40 (3.79)
0.4	200	4.98 (3.63)	10.16 (4.46)	1.43 (1.48)	15.98 (6.97)	15.20 (6.78)
0.4	500	11.95 (6.47)	12.29 (4.78)	1.71 (2.42)	34.65 (12.20)	33.31 (11.82)
0.8	3	0.82 (0.63)	0.89 (0.71)	0.83 (0.64)	2.28 (2.43)	2.29 (2.46)
0.8	10	1.08 (0.97)	3.39 (2.42)	1.10 (1.04)	1.61 (1.40)	1.58 (1.31)
0.8	100	2.80 (2.59)	8.20 (3.66)	1.36 (2.20)	7.25 (4.32)	7.21 (4.32)
0.8	200	4.26 (3.24)	9.83 (4.25)	1.67 (3.42)	12.53 (5.11)	12.72 (5.50)
0.8	500	9.87 (6.48)	11.73 (5.38)	3.30 (9.34)	26.54 (10.97)	26.63 (11.44)
1.6	3	0.47 (0.47)	0.53 (0.54)	0.47 (0.45)	0.84 (1.41)	0.74 (1.37)
1.6	10	0.54 (0.56)	1.76 (1.38)	0.63 (0.47)	7.84 (4.60)	7.86 (4.60)
1.6	100	1.31 (1.25)	4.34 (2.54)	2.81 (5.43)	3.55 (2.32)	2.21 (1.84)
1.6	200	1.82 (1.59)	5.03 (3.04)	4.24 (4.12)	5.24 (2.93)	5.16 (2.86)
1.6	500	7.83 (2.76)	7.54 (4.71)	7.36 (5.76)	11.16 (5.74)	10.74 (5.77)

Table 2 Results for mse_{σ_b} for `glmmLasso` and alternative approaches on Poisson data (standard errors in brackets)

σ_b	p	`glmer-select` mse_{σ_b}	`GLMMLasso` (SB) mse_{σ_b}	`glmmLasso` mse_{σ_b}	`bGLMM` (EM) mse_{σ_b}	`bGLMM` (REML) mse_{σ_b}
0.4	3	0.003 (0.005)	0.004 (0.005)	0.003 (0.005)	0.035 (0.093)	0.003 (0.004)
0.4	10	0.004 (0.006)	0.004 (0.006)	0.004 (0.005)	0.032 (0.053)	0.003 (0.005)
0.4	100	0.004 (0.006)	0.004 (0.006)	0.004 (0.005)	0.065 (0.169)	0.004 (0.005)
0.4	200	0.005 (0.007)	0.004 (0.007)	0.004 (0.005)	0.045 (0.072)	0.004 (0.006)
0.4	500	0.007 (0.010)	0.004 (0.006)	0.004 (0.005)	0.057 (0.080)	0.006 (0.010)
0.8	3	0.009 (0.010)	0.009 (0.010)	0.009 (0.009)	0.117 (0.208)	0.009 (0.009)
0.8	10	0.011 (0.014)	0.011 (0.014)	0.011 (0.012)	0.153 (0.254)	0.011 (0.012)
0.8	100	0.009 (0.012)	0.009 (0.011)	0.009 (0.011)	0.155 (0.277)	0.008 (0.010)
0.8	200	0.011 (0.014)	0.011 (0.13)	0.010 (0.013)	0.186 (0.361)	0.010 (0.012)
0.8	500	0.012 (0.014)	0.012 (0.014)	0.011 (0.012)	0.218 (0.351)	0.010 (0.014)
1.6	3	0.034 (0.046)	0.035 (0.046)	0.034 (0.040)	0.957 (1.209)	0.404 (0.545)
1.6	10	0.031 (0.040)	0.031 (0.040)	0.032 (0.040)	0.893 (0.999)	0.708 (0.582)
1.6	100	0.031 (0.045)	0.031 (0.045)	0.030 (0.041)	1.349 (1.358)	0.035 (0.058)
1.6	200	0.036 (0.049)	0.036 (0.049)	0.034 (0.043)	1.315 (1.423)	0.039 (0.059)
1.6	500	0.035 (0.047)	0.035 (0.047)	0.042 (0.054)	1.200 (1.321)	0.452 (0.572)

`select`)) as well as $\log(\text{mse}_{\sigma_b}(\cdot)/\text{mse}_{\sigma_b}$(`glmer-select`)) over the simulations for $\sigma_b = 0.4$.

Bernoulli Link The underlying model is the random intercept Bernoulli model

$$\eta_{it} = \sum_{j=1}^{p} x_{itj}\beta_j + b_i, \quad i = 1, \ldots, 40,\ t = 1, \ldots, 10,$$

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

with linear effects given by $\beta_1 = -5, \beta_2 = -10, \beta_3 = 15$ and $\beta_j = 0$, $j = 4, \ldots, 500$. Again we choose the different settings $p = 3, 10, 100, 200, 500$. For $j = 1, \ldots, p$ the vectors $\mathbf{x}_{it}^T = (x_{it1}, \ldots, x_{itp})$ have been drawn independently with components following a uniform distribution within the interval $[-0.1, 0.1]$, where parameters and interval boundaries have been chosen such that the frequencies of the two

Table 3 Computational times (in minutes) together with false positives and false negatives for `glmmLasso` and alternative approaches on Poisson data

		`glmer-select`			`GLMMLasso` (SB)			`glmmLasso`			`bGLMM` (EM)			`bGLMM` (REML)		
σ_b	p	time	f.p.	f.n.	time	f.p.	f.n.	time	f.p.	f.n.	time	f.p.	f.n.	time	f.p.	f.n.
0.4	3	5	0	0	528	0	0	47	0	0	22	0	0	159	0	0
0.4	10	25	0.15	0	312	1.00	0	114	0.18	0	100	1.02	0	320	1.09	0
0.4	100	1415	0.82	0	1677	1.25	0	369	0.36	0	1728	5.17	0	2214	5.42	0
0.4	200	1570	1.66	0	2904	1.27	0	872	0.17	0	1888	9.38	0	1764	9.26	0
0.4	500	4633	4.01	0	6618	1.77	0	3625	0.24	0.01	5687	16.96	0	6141	17.01	0
0.8	3	5	0	0	548	0	0	29	0	0	23	0	0	196	0	0
0.8	10	38	0.09	0	582	1.08	0	302	0.12	0	73	1.25	0	198	1.25	0
0.8	100	520	0.90	0	1138	1.50	0	545	0.37	0.01	568	5.37	0	1754	5.43	0
0.8	200	1691	1.54	0	4197	1.88	0	1786	1.15	0.01	1662	8.47	0	1838	8.62	0
0.8	500	5163	3.64	0	8020	2.41	0	4913	4.49	0.07	6815	15.16	0	7007	15.36	0
1.6	3	7	0	0	990	0	0	398	0	0	118	0	0	423	0	0
1.6	10	53	0.10	0	2280	1.11	0	2415	1.59	0	174	0.82	0	584	0.78	0
1.6	100	856	0.77	0	2105	3.22	0	1404	7.28	0.06	1596	5.50	0	3172	4.70	0
1.6	200	1670	1.21	0	6377	3.73	0	3121	6.13	0.03	3757	7.56	0	4392	7.42	0
1.6	500	4850	2.62	0	12743	6.02	0	4294	8.17	0.08	9440	12.66	0	10027	12.40	0

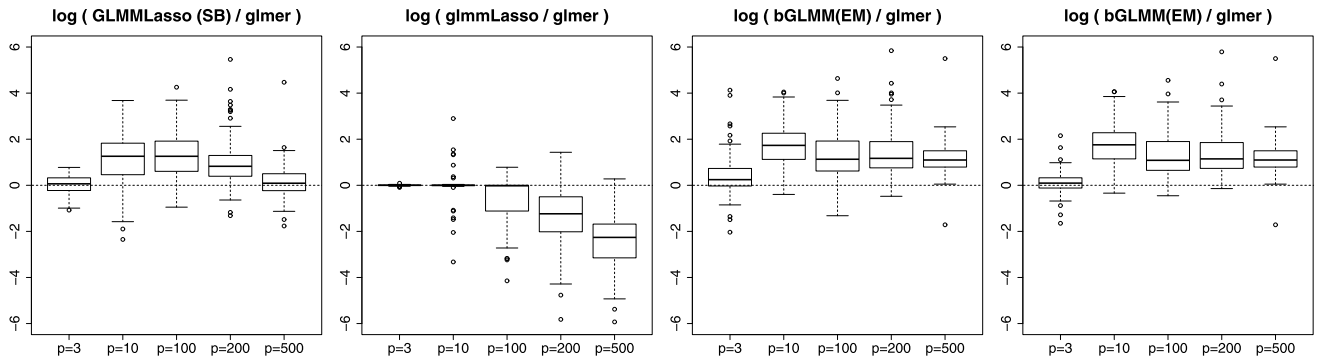

Fig. 1 Boxplots of $\log(\text{mse}_{\boldsymbol{\beta}}(\cdot)/\text{mse}_{\boldsymbol{\beta}}(\texttt{glmer-select}))$ for `glmmLasso` and alternative approaches on Poisson data for $\sigma_b = 0.4$

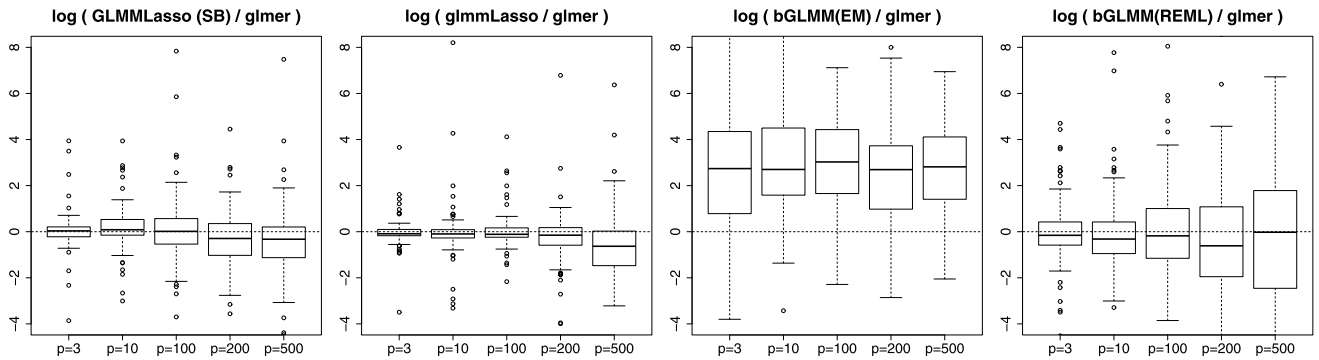

Fig. 2 Boxplots of $\log(\text{mse}_{\sigma_b}(\cdot)/\text{mse}_{\sigma_b}(\texttt{glmer-select}))$ for `glmmLasso` and alternative approaches on Poisson data for $\sigma_b = 0.4$

response categories are comparatively balanced. The number of observations remains $n = 40, T_i := T = 10, \forall i = 1, \ldots, n$. The random effects and the noise variable have been specified by $b_i \sim N(0, \sigma_b^2)$ with $\sigma_b = 0.4, 0.8, 1.6$.

Again, we evaluate the performance of estimators separately for the structural components and the variance and compare the alternative approaches. Results for varying number p of covariates $x_{it1}, \ldots, x_{itp}$ and different ran-

Table 4 Results for $\text{mse}_{\boldsymbol{\beta}}$ for `glmmLasso` and alternative approaches on Bernoulli data (standard errors in brackets)

σ_b	p	`glmer-select` $\text{mse}_{\boldsymbol{\beta}}$	`GLMMLasso` (SB) $\text{mse}_{\boldsymbol{\beta}}$	`glmmLasso` $\text{mse}_{\boldsymbol{\beta}}$	`bGLMM` (EM) $\text{mse}_{\boldsymbol{\beta}}$	`bGLMM` (REML) $\text{mse}_{\boldsymbol{\beta}}$
0.4	3	24.38 (14.07)	16.38 (15.90)	21.79 (14.34)	23.36 (15.94)	41.51 (35.40)
0.4	10	24.95 (17.61)	47.99 (32.48)	25.33 (17.73)	21.84 (16.79)	40.41 (35.42)
0.4	100	62.63 (45.16)	94.42 (42.16)	52.06 (43.82)	65.77 (34.80)	69.46 (66.06)
0.4	200	135.72 (112.56)	113.35 (255.05)	82.20 (77.55)	130.01 (255.05)	438.31 (769.35)
0.4	500	3873.83 (34363.55)	128.38 (41.83)	113.16 (84.81)	2879.71 (1369.55)	2540.35 (1237.83)
0.8	3	26.82 (16.56)	17.56 (16.58)	24.60 (18.24)	24.73 (20.42)	68.38 (44.29)
0.8	10	27.94 (20.00)	52.45 (29.80)	26.88 (22.07)	26.75 (19.95)	61.17 (39.13)
0.8	100	73.07 (54.42)	107.74 (44.44)	58.59 (45.37)	80.37 (38.22)	74.60 (98.11)
0.8	200	138.25 (92.77)	121.23 (42.73)	86.71 (95.67)	678.95 (1050.71)	1233.89 (993.02)
0.8	500	400.78 (504.87)	124.95 (43.20)	71.78 (61.58)	2519.97 (1889.11)	2408.70 (1792.54)
1.6	3	39.54 (28.81)	22.72 (27.52)	34.25 (28.75)	45.20 (37.86)	52.73 (43.12)
1.6	10	34.66 (21.66)	67.21 (46.18)	33.16 (22.07)	52.03 (33.07)	61.44 (36.39)
1.6	100	91.91 (68.77)	129.46 (56.49)	75.45 (55.50)	654.34 (565.97)	638.86 (493.11)
1.6	200	182.83 (178.20)	144.04 (54.66)	91.44 (88.97)	2427.09 (943.12)	2369.36 (866.69)
1.6	500	5016.08 (25797.97)	164.06 (60.74)	104.27 (95.97)	3954.57 (2598.53)	2879.37 (1943.81)

Table 5 Results for mse_{σ_b} for `glmmLasso` and alternative approaches on Bernoulli data (standard errors in brackets)

σ_b	p	`glmer-select` mse_{σ_b}	`GLMMLasso` (SB) mse_{σ_b}	`glmmLasso` mse_{σ_b}	`bGLMM` (EM) mse_{σ_b}	`bGLMM` (REML) mse_{σ_b}
0.4	3	0.056 (0.062)	0.058 (0.062)	0.044 (0.049)	0.080 (0.008)	0.053 (0.066)
0.4	10	0.051 (0.058)	0.068 (0.069)	0.037 (0.045)	0.078 (0.008)	0.043 (0.058)
0.4	100	0.047 (0.052)	0.071 (0.069)	0.036 (0.043)	0.064 (0.015)	0.036 (0.055)
0.4	200	0.053 (0.061)	0.075 (0.072)	0.051 (0.057)	0.119 (0.676)	0.272 (0.633)
0.4	500	2.590 (25.173)	0.074 (0.070)	0.035 (0.044)	3.854 (2.437)	1.379 (1.184)
0.8	3	0.043 (0.063)	0.046 (0.067)	0.036 (0.054)	0.343 (0.130)	0.046 (0.065)
0.8	10	0.044 (0.061)	0.063 (0.082)	0.036 (0.050)	0.347 (0.124)	0.048 (0.060)
0.8	100	0.039 (0.058)	0.070 (0.103)	0.032 (0.051)	0.299 (0.093)	0.102 (0.039)
0.8	200	0.050 (0.086)	0.079 (0.104)	0.037 (0.075)	1.819 (3.018)	0.721 (0.757)
0.8	500	0.102 (0.174)	0.086 (0.132)	0.038 (0.056)	4.373 (3.833)	1.413 (1.326)
1.6	3	0.060 (0.074)	0.058 (0.075)	0.065 (0.082)	4.681 (5.589)	0.220 (0.175)
1.6	10	0.062 (0.099)	0.086 (0.104)	0.070 (0.084)	4.136 (5.627)	0.206 (0.170)
1.6	100	0.070 (0.107)	0.116 (0.140)	0.065 (0.102)	8.077 (6.464)	0.302 (0.480)
1.6	200	0.105 (0.214)	0.135 (0.136)	0.077 (0.092)	14.010 (8.809)	1.021 (0.906)
1.6	500	15.032 (84.298)	0.129 (0.135)	0.084 (0.108)	19.661 (14.804)	1.663 (1.765)

dom effects variances σ_b are summarized in Tables 4, 5, and 6 and visualized in Figs. 3 and 4. The difference between approaches is less distinct, but `glmmLasso` again dominates the other approaches in terms of $\text{mse}_{\boldsymbol{\beta}}$ when noise variables are in the model. Besides, `glmmLasso` again yields the best results with respect to computational time as well as in terms of mse_{σ_b}. The performance of the boosting approaches is much worse than for the other approaches, especially for the case $p = 500$. For this really high-dimensional case tuning parameters have to be adapted but we abstain from adapting tuning parameters to specific high-dimensional settings. Boosting approaches perform well in terms of false negatives, but not in terms of false positives. The `GLMMLasso` (SB) dominates `glmmLasso` in terms of false positives but not in terms of false negatives. `glmer-select` has a tendency to include too many irrelevant variables.

Table 6 Computational times (in minutes) together with false positives and false negatives for `glmmLasso` and alternative approaches on Bernoulli data

		`glmer-select`			`GLMMLasso` (SB)			`glmmLasso`			`bGLMM` (EM)			`bGLMM` (REML)		
σ_b	p	time	f.p.	f.n.	time	f.p.	f.n.	time	f.p.	f.n.	time	f.p.	f.n.	time	f.p.	f.n.
0.4	3	3	0	0.52	143	0	0.11	1825	0	0.45	65	0	0.39	1590	0	0.81
0.4	10	16	0.05	0.49	146	0.61	0.35	742	0.11	0.51	228	0.11	0.34	742	0.06	0.78
0.4	100	260	0.87	0.53	590	0.68	0.61	1025	0.74	0.65	3140	1.54	0.47	4593	1.04	0.86
0.4	200	478	2.36	0.55	1573	0.64	0.82	1032	1.55	0.64	2822	2.92	0.56	3055	6.16	0.78
0.4	500	5615	6.50	0.57	2112	0.70	0.93	2104	2.33	0.74	7084	30.62	0.33	7570	30.65	0.43
0.8	3	3	0	0.57	271	0	0.13	187	0	0.51	161	0	0.38	657	0	1.23
0.8	10	17	0.06	0.55	273	0.32	0.75	182	0.16	0.48	420	0.32	0.28	1004	0	1.17
0.8	100	300	1.01	0.58	704	0.57	0.78	415	0.84	0.72	4283	2.37	0.38	5385	0.28	1.12
0.8	200	499	2.35	0.58	1647	0.64	0.78	732	1.79	0.67	2949	8.33	0.43	3375	14.35	0.58
0.8	500	3201	4.96	0.64	2029	0.60	0.88	423	0.88	0.91	6996	23.10	0.48	7758	22.38	0.71
1.6	3	4	0	0.80	345	0	0.15	85	0	0.67	211	0	0.43	1010	0	0.53
1.6	10	21	0.05	0.69	342	0.53	0.48	102	0.10	0.64	666	0.40	0.46	1537	0.39	0.50
1.6	100	338	0.95	0.74	931	0.62	0.99	257	1.17	0.76	5727	8.53	0.38	6454	8.79	0.38
1.6	200	723	2.32	0.66	2399	0.56	0.99	1385	1.36	0.87	3204	21.64	0.38	4056	21.52	0.38
1.6	500	11656	6.00	0.81	2281	0.54	1.23	1344	1.28	1.17	7635	23.67	0.53	8215	21.87	0.60

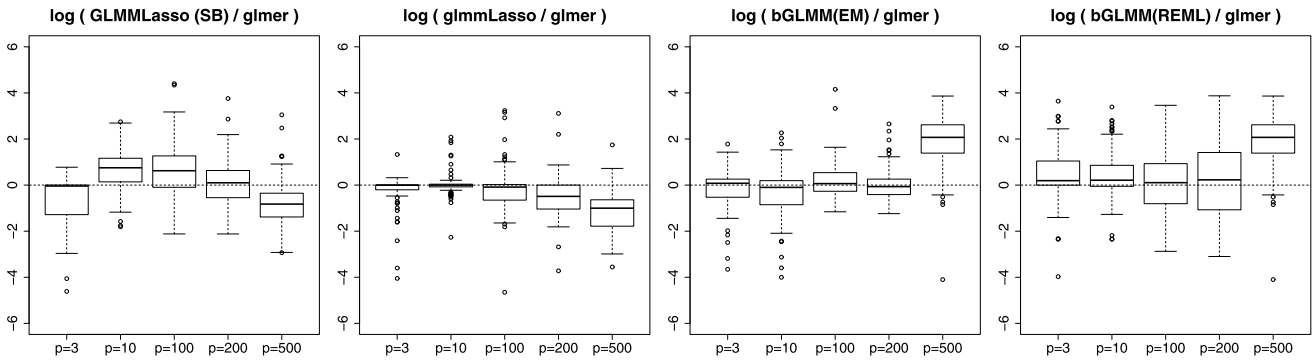

Fig. 3 Boxplots of $\log(\text{mse}_{\boldsymbol{\beta}}(\cdot)/\text{mse}_{\boldsymbol{\beta}}(\texttt{glmer-select}))$ for `glmmLasso` and alternative approaches on Bernoulli data for $\sigma_b = 0.4$

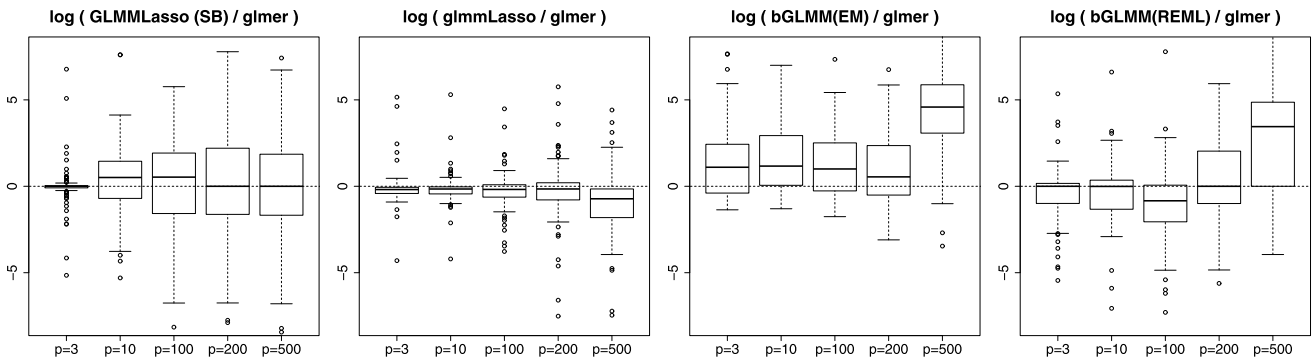

Fig. 4 Boxplots of $\log(\text{mse}_{\sigma_b}(\cdot)/\text{mse}_{\sigma_b}(\texttt{glmer-select}))$ for `glmmLasso` and alternative approaches on Bernoulli data for $\sigma_b = 0.4$

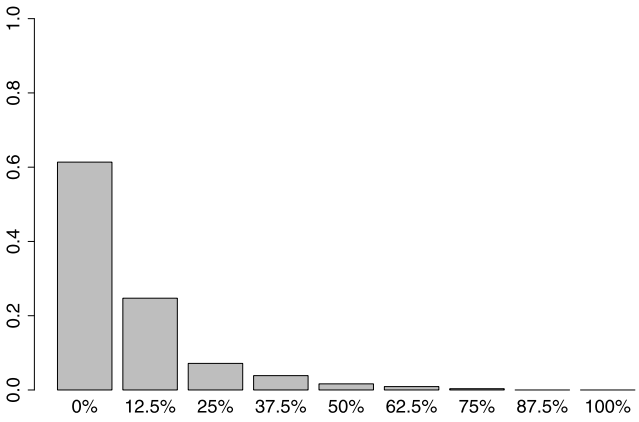

Fig. 5 Relative frequencies of the nine defoliation classes for all observation plots and all time points for the forest health data

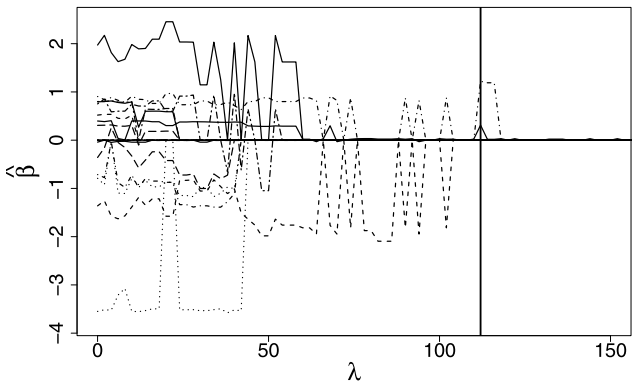

Fig. 6 Coefficient built-ups for the `glmmLasso` for the forest health data; the optimal value of the penalty parameter λ is shown by the vertical line

4 Applications to real data

In the following sections we will apply our lasso method on different real data sets and compare the results with other approaches. The tuning parameters λ have been chosen via BIC, see Appendix A. Standard errors for fixed effects and random effects variance components can be obtained by simulation-based parametric bootstrap evaluations, see Appendix C.

4.1 Forest health data

The forest health data has been considered in previous studies, for example in Kneib et al. (2009) and Tutz and Groll (2012). In this application, the health status of beeches at 83 observation plots located in a northern Bavarian forest district has been assessed in visual forest health inventories carried out between 1983 and 2004. Originally, the health status is classified on an ordinal scale, where the nine possible categories denote different degrees of defoliation. Figure 5 shows a histogram of the nine defoliation classes indicating that no trees were observed in the last two categories. We are now only interested in whether a tree is healthy or not, so we model the dichotomized response variable defoliation with categories 1 (not healthy; defoliation above or equal 12.5 %) and 0 (healthy; no defoliation; 0.0 %). In Kneib et al. (2009) a brief description of the covariates in the data set is presented, which is found in Table 7.

Table 7 Description of covariates for the forest health data

Covariate	Description
age	age of the tree in years (continuous, $7 \leq \text{age} \leq 234$)
elevation	elevation above sea level in meters (continuous, $250 \leq \text{elevation} \leq 480$)
inclination	inclination of slope in percent (continuous, $0 \leq \text{inclination} \leq 46$)
soil	depth of soil layer in centimeters (continuous, $9 \leq \text{soil} \leq 51$)
canopy	density of forest canopy in percent (continuous, $0 \leq \text{canopy} \leq 1$)
stand	type of stand (categorical, $1 =$ deciduous forest, $-1 =$ mixed forest)
fertilisation	fertilisation (categorical, $1 =$ yes, $-1 =$ no)
humus	thickness of humus layer in 5 categories (ordinal, higher categories represent higher proportions)
moisture	level of soil moisture (categorical, $1 =$ moderately dry, $2 =$ moderately moist, $3 =$ moist or temporary wet)
saturation	base saturation (ordinal, higher categories indicate higher base saturation)

As Kneib et al. (2009) identified a nonlinear effect of "age", we include some higher powers of "age" into our model, which results in the following predictor:

$$\begin{aligned} g(\pi_{it}) = {} & \beta_0 + \text{age}_{it}\beta_1 + \text{age}_{it}^2\beta_2 + \text{age}_{it}^3\beta_3 + \text{age}_{it}^4\beta_4 \\ & + \text{elevation}_{it}\beta_5 + \text{inclination}_{it}\beta_6 + \text{soil}_{it}\beta_7 \\ & + \text{canopy}_{it}\beta_8 + \text{fertilisation}_{it}\beta_9 + \text{stand}_{it}\beta_{10} \\ & + \text{humus0}_{it}\beta_{11} + \text{humus2}_{it}\beta_{12} + \text{humus3}_{it}\beta_{13} \\ & + \text{humus4}_{it}\beta_{14} + \text{saturation1}_{it}\beta_{15} \\ & + \text{saturation3}_{it}\beta_{16} + \text{saturation4}_{it}\beta_{17} \\ & + \text{moisture1}_{it}\beta_{18} + \text{moisture3}_{it}\beta_{19} + b_i, \end{aligned}$$

where $\pi_{it} = \mu_{it}$ denotes the expected probability of defoliation for observation area i at time t and $b_i \sim N(0, \sigma_b^2)$ represent cluster-specific random intercepts. We fit a binomial model with logit-link, building groups for the categorial variables "humus", "moisture" and "saturation". For this purpose we use the extended algorithm for categorical predictors from Sect. 3.3. Results for the parameter estimates

Table 8 Estimates for the forest health data (standard errors in brackets)

	`glmer-select`	`GLMMLasso` (SB)	`glmmLasso`
Intercept	−0.948 (0.688)	−0.867	−7.642 (1.904)
age	0.029 (0.004)	–	0.311 (0.092)
age^2	–	–	−0.006 (0.002)
age^3	–	0.000	0.000 (0.000)
age^4	–	0.000	0.000 (0.000)
elevation	–	–	–
inclination	–	–	–
soil	–	–	–
canopy	−3.497 (0.525)	−1.299	–
fertilisation	−1.905 (0.615)	–	–
stand	–	–	1.220 (1.138)
humus0	–	0.646	–
humus2	–	–	–
humus3	–	–	–
humus4	–	–	–
saturation1	–	–	–
saturation3	–	–	–
saturation4	–	–	–
moisture1	–	–	–
moisture3	–	–	–
$\hat{\sigma}_b$	1.865	2.178	1.895 (0.116)

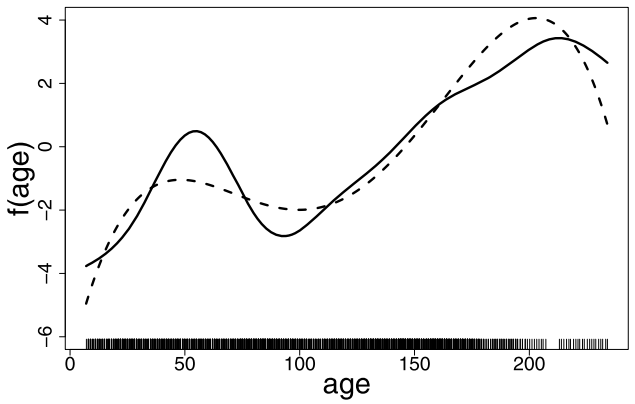

Fig. 7 Smoothed age effect for the forest health data with `gamm` (*solid line*) and `glmmLasso` (*dashed line*)

are found in Table 8 and the corresponding coefficient built-ups are illustrated in Fig. 6.

The penalty parameter λ for the `glmmLasso` again was determined by BIC on the interval [0; 300]. The chosen parameter was rather high, $\lambda_{\text{opt}} = 112$, indicating that penalization improves the fit compared to ordinary fitting procedures which are obtained for $\lambda = 0$ and consequently only few of the variables are included. The smooth effect of age on tree defoliation for our binomial model with logit-link is shown in Fig. 7. For comparison, the smooth effect obtained by a penalized basis function approach, which is implemented in the `gamm` function of the R-package `mgcv` (Wood 2006), is shown. Obviously with increasing age of the trees the probability of defoliation increases in a nonlinear fashion.

Table 9 Description of covariates for the Jimma data

Covariate	Description
age	age of the child in days (continuous, $0 \leq \text{age} \leq 385$)
ageM	age of the mother in years (continuous, $14 \leq \text{ageM} \leq 50$)
education	educational level of the mother (categorical, 1 = illiterate, 2 = read and write, 3 = elementary school, 4 = junior high school, 5 = high school, 6 = college and above)
delivery	place of delivery (categorical, 1 = hospital, 2 = health center, 3 = home)
visits	number of antenatal visits (categorical, 0, ≥ 1)
month	month of birth (categorical, 1 = Jan.–June, 0 = July–Dec.)
sex	sex of the child (categorical, 1 = male, 0 = female)
marital	marital status of mother (categorical, 1 = married, 2 = divorced, 3 = widowed, 4 = never married)
status	occupational status of mother (categorical, 1 = unemployed, 0 = employed)

4.2 Jimma Infant Survival Study

The Jimma Infant Survival Differential Longitudinal Study is a cohort study investigating the live births which took place in the town of Jimma in Ethiopia during a one year period from September 1992 until September 1993. An extensive description can be found in Lesaffre et al. (1999). The study covers 8000 households with live births in the said period. Following Lesaffre et al. (1999) and Tutz and Reithinger (2007), 495 singleton live births have been considered and monitored for a one year period in order to determine the risk factors for infant mortality. A good indicator of a child's health status is the body weight. Hence, to determine possible influence factors on growth of the children, we use the (logarithmic) body weight (in kg) as response variable together with some socio-economic and demographic as well as some prenatal and delivery-related covariates. A brief description of all considered covariates can be found in Table 9.

Tutz and Reithinger (2007) identified a nonlinear effect of "age", therefore we include also "age^2" into our model, resulting in the following predictor:

$$g(\mu_{it}) = \beta_0 + \text{age}_{it}\beta_1 + \text{age}_{it}^2\beta_2 + \text{ageM}_{it}\beta_3$$

Table 10 Estimates for the standard deviations of the random effects for the Jimma data with `glmer-select`, with `lmmlasso` function and `glmmLasso` algorithm (bootstrap standard errors in brackets)

	`glmer-select`	`lmmlasso`	`glmmLasso`
$\hat{\sigma}_{b_0}$	0.294	0.400	0.078 (0.001)
$\hat{\sigma}_{b_1}$	0.001	0.003	0.000 (0.003)
$\hat{\sigma}_{b_2}$	0.000	0.000	0.000 (0.002)

$$
\begin{aligned}
&+ \text{education1}_{it}\beta_4 + \text{education2}_{it}\beta_5 \\
&+ \text{education3}_{it}\beta_6 + \text{education4}_{it}\beta_7 \\
&+ \text{education5}_{it}\beta_8 + \text{delivery1}_{it}\beta_9 \\
&+ \text{delivery2}_{it}\beta_{10} + \text{visits}_{it}\beta_{11} + \text{month}_{it}\beta_{12} \\
&+ \text{sex}_{it}\beta_{13} + \text{marital1}_{it}\beta_{14} + \text{marital2}_{it}\beta_{15} \\
&+ \text{marital3}_{it}\beta_{16} + \text{status}_{it}\beta_{17} \\
&+ b_{0i} + \text{age}_{it}b_{1i} + \text{age}_{it}^2 b_{2i},
\end{aligned}
$$

where μ_{it} denotes the expected body weight of child i at time t and $\mathbf{b}_i = (b_{0i}, b_{1i}, b_{2i})^T \sim N(\mathbf{0}, \mathbf{Q})$ represent child-specific random intercepts and random slopes on age and squared age. The continuous variables age, squared age and age of the mother have been standardized. We fit a normal distribution model with log-link, building groups for the categorial variables "education", "delivery" and "marital". So again the extended algorithm for categorical predictors from Sect. 3.3 is required. For `GLMMLasso` (SB) only Poisson and binomial model are implemented, so we use the `lmmlasso` from the corresponding R-package (see Schelldorfer 2011) as well as `glmer-select` for comparison. Note here, that for the `lmmlasso` function a warning message is reported, that covariance parameters are redundant. The estimates for the standard deviations of the random effects for the standardized model are presented in Table 10.

The results for the estimated linear effects corresponding to the original scaling of the variables can be found in Table 11 and the corresponding coefficient built-ups are illustrated in Fig. 8. The BIC is plotted against the penalty parameter λ in Fig. 9. Again penalization improves ordinary fitting procedures obtained for $\lambda = 0$ and a sparse model is chosen with a clearly non-linear influence of the child's age. Only the `lmmlasso` function detects a linear influence of the variable "sex".

The child-specific smooth effects of the children's age on the body weight for `glmmLasso` are shown in Fig. 10. As was to be expected, with increasing age of the children their body weight increases, at first relatively fast, but slowing down after the first 200 days. The main feature of the penalized approach is that variables that also turn out to be non-influential are automatically selected.

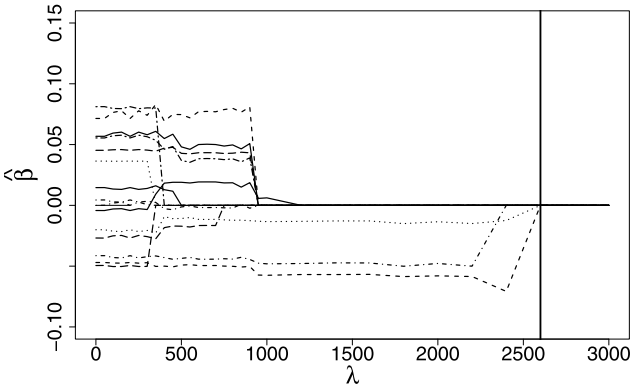

Fig. 8 Coefficient built-ups for the `glmmLasso` for the Jimma data; the optimal value of the penalty parameter λ is shown by the *vertical line*

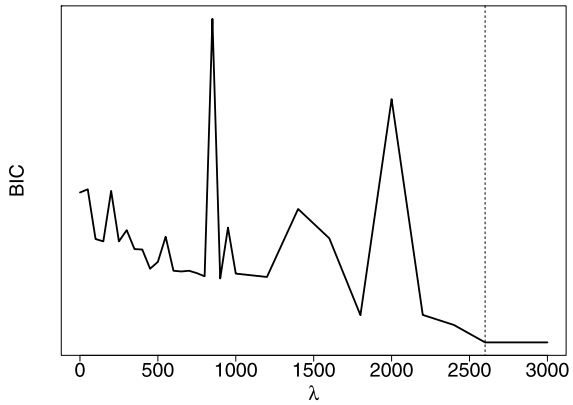

Fig. 9 Results for BIC for the `glmmLasso` as function of penalty parameter λ for the Jimma data; the optimal value of the penalty parameter λ is shown by the *vertical line*

5 Concluding remarks

Several procedures for variable selection based on L_1-penalties have been proposed. The procedures yield stable estimates in cases where methods that do not include variable selection typically fail because of the complexity of the fitting task. The method allows to include categorical predictors that are selected or omitted as a whole predictor in the spirit of the group lasso. It is straightforward to extend the approach to include more complex penalty terms, for example, the elastic net penalty or hierarchical penalty terms as proposed by Zhao et al. (2009). Even though the procedures work in linear mixed models the main focus of this article was on GLMMs.

As suggested by a referee we included a comparison with the approach recently proposed by Schelldorfer and Bühlmann (2011). In contrast to the gradient ascent algorithm proposed here, they suggest a coordinate gradient descent method based on a quadratic approximation of the penalized log-likelihood and perform indirect line search to

Table 11 Estimated linear effects for the Jimma data with `glmer-select`, with `lmmlasso` function and `glmmLasso` algorithm (standard errors in brackets)

	`glmer-select`	`lmmlasso`	`glmmLasso`
Intercept	1.288 (0.007)	3.293 (–)	1.288 (0.010)
age	0.005 (0.000)	0.023 (–)	0.005 (0.000)
age^2	−0.000 (0.000)	−0.000 (–)	−0.000 (0.000)
ageM	–	–	–
education1	–	–	–
education2	–	–	–
education3	–	–	–
education4	–	–	–
education5	–	–	–
delivery1	–	–	–
delivery2	–	–	–
visits	–	–	–
month	–	–	–
sex	–	0.088 (–)	–
marital1	–	–	–
marital2	–	–	–
marital3	–	–	–
occupational	–	–	–

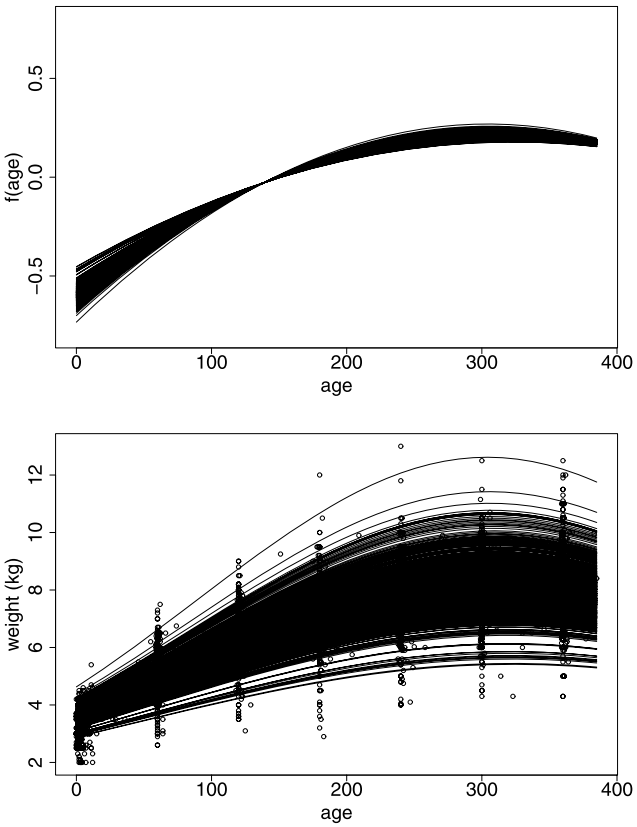

Fig. 10 Individual smoothed age effects for the Jimma data on the predictor level (*upper*) and *versus* body weight (*lower*) for `glmmLasso` with slopes up to second potence of age

ensure that the objective function decreases. Also the computation of variance components is different and our procedure includes a final re-estimation step. It should be noted that the selection of the tuning parameter in our procedure is based on the whole procedure, iterative estimates and final re-estimation. This may account for the difference in performance. The simulation study shows that the new `glmmLasso` algorithm is highly competitive, both with respect to the accuracy of $\boldsymbol{\beta}$-estimates and estimates of the random effects variance parameters as well as the computational efficiency. An advantage of `glmmLasso` is the appropriate treatment of categorical variables. Another advantage of the `glmmLasso` function comes from the implementation side: `glmmLasso` is able to fit many GLM-families and to consider all kinds of random effects covariance matrices, whereas the `GLMMLasso` (SB) method at the moment only allows for binomial and Poisson family and two rather simple structures for the random effects covariance matrix.

After finishing the present paper we found that an alternative approach proposed by Ibrahim et al. (2011) is available online, but in the simulations as well as in the application only linear mixed models were considered. They select both fixed and random effects in a general class of mixed effects models using maximum penalized likelihood (MPL) estimation along with the smoothly clipped absolute deviation (SCAD) and the adaptive least absolute shrinkage and selection operator (ALASSO) penalty functions. They specify penalty parameters as hyperparameters. In contrast to the gradient ascent approach used here they use the expectation-maximization (EM) algorithm to simultaneously optimize the penalized likelihood function and estimate the penalty parameters.

Appendix A: Determination of the tuning parameter λ

First of all, define a fine grid of different values for the tuning parameter, $0 \leq \lambda_1 \leq \cdots \leq \lambda_L \leq \infty$. Next, the optimal tuning parameter is determined using one of the following techniques and finally, the whole data set is fitted again using the `glmmLasso` algorithm with λ_{opt} to obtain the final estimates $\hat{\boldsymbol{\delta}}, \hat{\mathbf{Q}}$ and the corresponding fit $\hat{\boldsymbol{\mu}}$.

One way to determine the tuning parameter is based on information criteria. In the following we focus on Akaike's information criterion (AIC, see Akaike 1973) as well as on the Bayesian information criterion (BIC, see Schwarz 1978), also known as Schwarz's information criterion, given by:

$$AIC_l = -2l\big(\hat{\boldsymbol{\mu}}^{(j)}\big) + 2df(\lambda_j),$$

$$BIC_l = -2l\big(\hat{\boldsymbol{\mu}}^{(j)}\big) + \log(n)df(\lambda_j),$$

$j \in \{1, \ldots, L\}$, where $l(\hat{\boldsymbol{\mu}}^{(j)})$ denotes the approximated log-likelihood from (4) evaluated at the fit corresponding to λ_j and $df(\lambda_j)$ denotes the degrees of freedom, which are equal to the sum of the number of nonzero fixed-effects coefficients and the number of covariance parameters, that is $df(\lambda_j) = \#\{k : 1 \leq k \leq p, \hat{\beta}_k \neq 0\} + \frac{q(q+1)}{2}$ (compare Schelldorfer and Bühlmann 2011). Finally, for the optimal tuning parameter λ_{opt} the chosen information criterion is minimal.

Alternatively to information criteria, the optimal tuning parameter λ_{opt} can be derived using K-fold cross-validation. For this purpose the original sample is randomly partitioned into K subsamples and the model is fitted on $K-1$ subsamples (training data). The remaining subsample (test data) is used for validation. The adequacy of the model for λ_j, $j \in \{1, \ldots, L\}$ can be assessed by evaluating a cross-validation score on the test data, for example, the deviance

$$D_j = -2\phi \sum_{i=1}^{n_{test}} \left[l_i\left(\hat{\mu}_i^{(j)}\right) - l_i(y_i) \right],$$

where $l_i(\cdot)$ denotes the log-likelihood contribution of sample element i. In special situations other measures of fit can also be used, for example the misclassification error rate for binary responses or the mean squared error for continuous responses. The procedure is then repeated K times, with each subsample used exactly once as test data. For the optimal tuning parameter the cross-validation score averaged over all K folds is minimal. The concept of splitting the data into parts has a long history and has already been discussed, for example, by Stone (1974, 1978), Geissler (1975) and Picard and Cook (1984).

Appendix B: Partition of Fisher matrix

According to Fahrmeir and Tutz (2001) the penalized pseudo-Fisher matrix $\mathbf{F}^{\text{pen}}(\boldsymbol{\delta}) = \mathbf{A}^T\mathbf{W}(\boldsymbol{\delta})\mathbf{A} + \mathbf{K}$ can be partitioned into

$$\mathbf{F}^{\text{pen}}(\boldsymbol{\delta}) = \begin{bmatrix} \mathbf{F}_{\boldsymbol{\beta}\boldsymbol{\beta}} & \mathbf{F}_{\boldsymbol{\beta}1} & \mathbf{F}_{\boldsymbol{\beta}2} & \ldots & \mathbf{F}_{\boldsymbol{\beta}n} \\ \mathbf{F}_{1\boldsymbol{\beta}} & \mathbf{F}_{11} & & & 0 \\ \mathbf{F}_{2\boldsymbol{\beta}} & & \mathbf{F}_{22} & & \\ \vdots & & & \ddots & \\ \mathbf{F}_{n\boldsymbol{\beta}} & 0 & & & \mathbf{F}_{nn} \end{bmatrix},$$

with single components

$$\mathbf{F}_{\boldsymbol{\beta}\boldsymbol{\beta}} = -E\left(\frac{\partial^2 l^{\text{pen}}(\boldsymbol{\delta})}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^T}\right) = \mathbf{X}^T\mathbf{D}(\boldsymbol{\delta})\boldsymbol{\Sigma}(\boldsymbol{\delta})^{-1}\mathbf{D}(\boldsymbol{\delta})^T\mathbf{X},$$

$$\mathbf{F}_{\boldsymbol{\beta}i} = \mathbf{F}_{i\boldsymbol{\beta}}^T = -E\left(\frac{\partial^2 l^{\text{pen}}(\boldsymbol{\delta})}{\partial\boldsymbol{\beta}\partial\mathbf{b}_i^T}\right)$$

$$= \mathbf{X}_i^T\mathbf{D}_i(\boldsymbol{\delta})\boldsymbol{\Sigma}_i(\boldsymbol{\delta})^{-1}\mathbf{D}_i(\boldsymbol{\delta})^T\mathbf{Z}_i,$$

$$\mathbf{F}_{ii} = -E\left(\frac{\partial^2 l^{\text{pen}}(\boldsymbol{\delta})}{\partial\mathbf{b}_i\partial\mathbf{b}_i^T}\right)$$

$$= \mathbf{Z}_i^T\mathbf{D}_i(\boldsymbol{\delta})\boldsymbol{\Sigma}_i(\boldsymbol{\delta})^{-1}\mathbf{D}_i(\boldsymbol{\delta})^T\mathbf{Z}_i + \mathbf{Q}^{-1},$$

and $\mathbf{D}_i(\boldsymbol{\delta}) = \partial h(\boldsymbol{\eta}_i)/\partial\boldsymbol{\eta}$, $\boldsymbol{\Sigma}_i(\boldsymbol{\delta}) = \text{cov}(\mathbf{y}_i|\boldsymbol{\beta}, \mathbf{b}_i)$.

Appendix C: Two bootstrap approaches for GLMMs

The general idea of bootstrapping has been developed by Efron (1983, 1986). An extensive overview of the bootstrap and related methods for asserting statistical accuracy can be found in Efron and Tibshirani (1993). For GLMMs two main approaches are found in the literature. The first approach is to resample nonparametrically, which has been proposed e.g. by McCullagh (2000) and Davison and Hinkley (1997). They randomly sample groups of observations with replacement at the first stage and suggest various ways how to sample within the groups at the second stage. They showed that sometimes it can be useful to randomly resample groups at the first stage only and leave groups themselves unchanged, for example if there is a longitudinal structure in the data, see e.g. Shang and Cavanaugh (2008).

The second approach, on which the standard errors in Sect. 4 are based on, is to simulate parametric bootstrap samples following the parametric distribution family of the underlying model (compare Efron 1982). Booth (1996) has extended the parametric approach from Efron (1982) to GLMMs to estimate standard errors for the fitted linear predictor $\hat{\boldsymbol{\eta}} = \mathbf{X}\hat{\boldsymbol{\beta}} + \mathbf{Z}\hat{\mathbf{b}}$ from Sect. 2.

Analogously we can derive standard errors for the fixed effects estimate $\hat{\boldsymbol{\beta}}$ and for the estimated random effects variance components $\hat{\mathbf{Q}}$, respectively. Let $\{F_{\boldsymbol{\xi}} : \boldsymbol{\xi} \in \Xi\}$ denote the parametric distribution family of the underlying model, where $\boldsymbol{\xi}^T = (\boldsymbol{\beta}^T, \text{vec}(\mathbf{Q})^T)$ is unknown. Here $\text{vec}(\mathbf{Q})$ denotes the column-wise vectorization of matrix $\mathbf{Q}$ to a column vector. Let $\hat{\boldsymbol{\xi}} = (\hat{\boldsymbol{\beta}}^T, \text{vec}(\hat{\mathbf{Q}})^T)$ denote the Lasso estimate of $\boldsymbol{\xi}$ for an already chosen penalty parameter λ on a certain data set. Now we can simulate new bootstrap data sets $(\mathbf{y}^*, \mathbf{b}^*)$ with respect to the distribution $F_{\hat{\boldsymbol{\xi}}}$, i.e. $(\mathbf{y}^*, \mathbf{b}^*) \sim F_{\hat{\boldsymbol{\xi}}}$. We repeat this procedure sufficiently often, say $B = 10.000$, and fit every new bootstrap data set $(\mathbf{y}^*_{(r)}, \mathbf{X}, \mathbf{W})$, $r = 1, \ldots, B$, with our `glmmLasso` algorithm. The new fits $\hat{\boldsymbol{\xi}}^*_{(r)}$ corresponding to the r-th new data set serve as bootstrap estimates and can be used to derive standard errors.

Although consistency of straightforward bootstrap in L_1-penalized regression can fail even in the simple case of linear regression (Chatterjee and Lahiri 2011), in the finite dimensional case bootstrap is helpful and we found that it yields reasonable results.

References

Akaike, H.: Information theory and the extension of the maximum likelihood principle. In: Second International Symposium on Information Theory, pp. 267–281 (1973)

Bates, D., Maechler, M.: lme4: linear mixed-effects models using S4 classes. R package version 0.999375-34 (2010)

Bondell, H.D., Krishna, A., Ghosh, S.K.: Joint variable selection of fixed and random effects in linear mixed-effects models. Biometrics **66**, 1069–1077 (2010)

Booth, J.G.: Bootstrap methods for generalized mixed models with applications to small area estimation. In: Seeber, G.U.H., Francis, B.J., Hatzinger, R., Steckel-Berger, G. (eds.) Statistical Modelling, vol. 104, pp. 43–51. Springer, New York (1996)

Booth, J.G., Hobert, J.P.: Maximizing generalized linear mixed model likelihoods with an automated Monte Carlo EM algorithm. J. R. Stat. Soc. B **61**, 265–285 (1999)

Breiman, L.: Heuristics of instability and stabilization in model selection. Ann. Stat. **6**, 2350–2383 (1996)

Breiman, L.: Arcing classifiers. Ann. Stat. **26**, 801–849 (1998)

Breslow, N.E., Clayton, D.G.: Approximate inference in generalized linear mixed model. J. Am. Stat. Assoc. **88**, 9–25 (1993)

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

Broström, G.: glmmML: generalized linear models with clustering. R package version 0.81-6 (2009)

Bühlmann, P., Hothorn, T.: Boosting algorithms: regularization, prediction and model fitting. Stat. Sci. **22**, 477–522 (2007)

Bühlmann, P., Yu, B.: Boosting with the L_2 loss: regression and classification. J. Am. Stat. Assoc. **98**, 324–339 (2003)

Candes, E., Tao, T.: The Dantzig selector: statistical estimation when p is much larger than n. Ann. Stat. **35**, 2313–2351 (2007)

Chatterjee, A., Lahiri, S.N.: Bootstrapping lasso estimators. J. Am. Stat. Assoc. **106**, 608–625 (2011)

Davison, A.C., Hinkley, D.V.: Bootstrap Methods and Their Application. Cambridge University Press, Cambridge (1997)

Efron, B.: The Jackknife, the Bootstrap and Other Resampling Plans. SIAM: CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 38. (1982)

Efron, B.: Estimating the error rate of a prediction rule: improvement on crossvalidation. J. Am. Stat. Assoc. **78**, 316–331 (1983)

Efron, B.: How biased is the apparent error rate of a prediction rule? J. Am. Stat. Assoc. **81**, 461–470 (1986)

Efron, B., Tibshirani, R.: An Introduction to the Bootstrap. Chapman & Hall, New York (1993)

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. Ann. Stat. **32**, 407–499 (2004)

Fahrmeir, L., Lang, S.: Bayesian inference for generalized additive mixed models based on Markov random field priors. Appl. Stat. **50**, 201–220 (2001). doi:10.1111/1467-9876.00229

Fahrmeir, L., Tutz, G.: Multivariate Statistical Modelling Based on Generalized Linear Models, 2nd edn. Springer, New York (2001)

Fan, J., Li, R.: Variable selection via nonconcave penalize likelihood and its oracle properties. J. Am. Stat. Assoc. **96**, 1348–1360 (2001)

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

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

Geissler, S.: The predictive sample reuse method with applications. J. Am. Stat. Assoc. **70**, 320–328 (1975)

Genkin, A., Lewis, D., Madigan, D.: Large-scale Bayesian logistic regression for text categorization. Technometrics **49**, 291–304 (2007)

Goeman, J.J.: L_1 penalized estimation in the Cox proportional hazards model. Biom. J. **52**, 70–84 (2010)

Groll, A.: glmmLasso: Variable Selection for Generalized Linear Mixed Models by L_1-penalized Estimation. R package version 1.0.1 (2011a)

Groll, A.: GMMBoost: Componentwise Likelihood-based Boosting Approaches to Generalized Mixed Models. R package version 1.0.2 (2011b)

Gui, J., Li, H.Z.: Penalized Cox regression analysis in the high-dimensional and low-sample size settings, with applications to microarray gene expression data. Bioinformatics **21**, 3001–3008 (2005)

Hastie, T., Rosset, S., Tibshirani, R., Zhu, J.: The entire regularization path for the support vector machine. J. Mach. Learn. Res. **5**, 1391–1415 (2004)

Ibrahim, J.G., Zhu, H., Garcia, R.I., Guo, R.: Fixed and random effects selection in mixed effects models. Biometrics **67**, 495–503 (2011)

James, G.M., Radchenko, P.: A generalized Dantzig selector with shrinkage tuning. Biometrika **96**(2), 323–337 (2009)

Kim, Y., Kim, J.: Gradient lasso for feature selection. In: Proceedings of the 21st International Conference on Machine Learning. ACM International Conference Proceeding Series, vol. 69, pp. 473–480 (2004)

Kneib, T., Hothorn, T., Tutz, G.: Variable selection and model choice in geoadditive regression. Biometrics **65**, 626–634 (2009)

Lesaffre, E., Asefa, M., Verbeke, G.: Assessing the godness-of-fit of the laird and ware model—an example: the Jimma infant survival differential longitudinal study. Stat. Med. **18**, 835–854 (1999)

Lin, X., Breslow, N.E.: Bias correction in generalized linear mixed models with multiple components of dispersion. J. Am. Stat. Assoc. **91**, 1007–1016 (1996)

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

McCullagh, P.: Re-sampling and exchangeable arrays. Bernoulli **6**, 303–322 (2000)

McCulloch, C.E., Searle, S.R., Neuhaus, J.M.: Generalized, Linear and Mixed Models, 2nd edn. Wiley, New York (2008)

Meier, L., Van de Geer, S., Bühlmann, P.: The group lasso for logistic regression. J. R. Stat. Soc. B **70**, 53–71 (2008)

Ni, X., Zhang, D., Zhang, H.H.: Variable selection for semiparametric mixed models in longitudinal studies. Biometrics **66**, 79–88 (2010)

Osborne, M., Presnell, B., Turlach, B.: On the lasso and its dual. J. Comput. Graph. Stat. (2000)

Park, M.Y., Hastie, T.: L_1-regularization path algorithm for generalized linear models. J. R. Stat. Soc. B **19**, 659–677 (2007)

Picard, R., Cook, D.: Cross-validation of regression models. J. Am. Stat. Assoc. **79**, 575–583 (1984)

Pinheiro, J.C., Bates, D.M.: Mixed-Effects Models in S and S-Plus. Springer, New York (2000)

Radchenko, P., James, G.M.: Variable inclusion and shrinkage algorithms. J. Am. Stat. Assoc. **103**, 1304–1315 (2008)

Schall, R.: Estimation in generalised linear models with random effects. Biometrika **78**, 719–727 (1991)

Schelldorfer, J.: lmmlasso: Linear mixed-effects models with Lasso. R package version 0.1-2. (2011)

Schelldorfer, J., Bühlmann, P.: GLMMLasso: an algorithm for high-dimensional generalized linear mixed models using L_1-penalization. Preprint, ETH Zurich, (2011). http://stat.ethz.ch/people/schell

Schelldorfer, J., Bühlmann, P., van de Geer, S.: Estimation for high-dimensional linear mixed-effects models using L_1-penalization. Scand. J. Stat. **38**(2), 197–214 (2011)

Schwarz, G.: Estimating the dimension of a model. Ann. Stat. **6**, 461–464 (1978)

Segal, M.R.: Microarray gene expression data with linked survival phenotypes: diffuse large-b-cell lymphoma revisited. Biostatistics **7**, 268–285 (2006)

Shang, J., Cavanaugh, J.E.: Bootstrap variants of the Akaike information criterion for mixed model selection. Comput. Stat. Data Anal. **52**, 2004–2021 (2008)

Shevade, S.K., Keerthi, S.S.: A simple and efficient algorithm for gene selection using sparse logistic regression. Bioinformatics **19**, 2246–2253 (2003)

Stone, M.: Cross-validatory choice and assessment of statistical predictions (with discussion). J. R. Stat. Soc. B **36**, 111–147 (1974)

Stone, M.: Cross-validation: A review. Math. Oper.forsch. Stat. **9**, 127–139 (1978)

Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. B **58**, 267–288 (1996)

Tibshirani, R.: The lasso method for variable selection in the Cox model. Stat. Med. **16**, 385–395 (1997)

Tutz, G., Groll, A.: Generalized linear mixed models based on boosting. In: Kneib, T., Tutz, G. (eds.) Statistical Modelling and Regression Structures—Festschrift in the Honour of Ludwig Fahrmeir. Physica, Heidelberg (2010)

Tutz, G., Groll, A.: Likelihood-based boosting in binary and ordinal random effects models. J. Comput. Graph. Stat. (2012). doi:10.1080/10618600.2012.694769

Tutz, G., Reithinger, F.: A boosting approach to flexible semiparametric mixed models. Stat. Med. **26**, 2872–2900 (2007)

Venables, W.N., Ripley, B.D.: Modern Applied Statistics with S, 4th edn. Springer, New York (2002)

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

Wang, D., Eskridge, K.M., Crossa, J.: Identifying QTLs and epistasis in structured plant populations using adaptive mixed lasso. J. Agric. Biol. Environ. Stat. **16**, 170–184 (2010a)

Wang, S., Song, P.X., Zhu, J.: Doubly regularized REML for estimation and selection of fixed and random effects in linear mixed-effects models. Technical Report 89, The University of Michigan, (2010b)

Wolfinger, R.W.: Laplace's approximation for nonlinear mixed models. Biometrika **80**, 791–795 (1994)

Wolfinger, R., O'Connell, M.: Generalized linear mixed models; a pseudolikelihood approach. J. Stat. Comput. Simul. **48**, 233–243 (1993)

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

Yang, H.: Variable selection procedures for generalized linear mixed models in longitudinal data analysis. PhD thesis, North Carolina State University (2007)

Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. J. R. Stat. Soc. B **68**, 49–67 (2006)

Zhao, P., Rocha, G., Yu, B.: The composite absolute penalties family for grouped and hierarchical variable selection. Ann. Stat. **37**, 3468–3497 (2009)

Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. J. R. Stat. Soc. B **67**, 301–320 (2005)

Zou, H., Hastie, T.: The adaptive lasso and its oracle properties. J. Am. Stat. Assoc. **101**, 1418–1429 (2006)