LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

INSTITUT FÜR STATISTIK

Jan Ulbricht & Gerhard Tutz

Boosting Correlation Based Penalization in Generalized Linear Models

Technical Report Number 009, 2007
Department of Statistics
University of Munich

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

Boosting Correlation Based Penalization in Generalized Linear Models

Jan Ulbricht & Gerhard Tutz

11th December 2007

Abstract

In high dimensional regression problems penalization techniques are a useful tool for estimation and variable selection. We propose a novel penalization technique that aims at the grouping effect which encourages strongly correlated predictors to be in or out of the model together. The proposed penalty uses the correlation between predictors explicitly. We consider a simple version that does not select variables and a boosted version which is able to reduce the number of variables in the model. Both methods are derived within the framework of generalized linear models. The performance is evaluated by simulations and by use of real world data sets.

Keywords: Correlation based estimator, Boosting, Variable selection, Generalized linear models.

1 Introduction

Linear models have a long tradition in statistics as nicely summarized in Toutenburg (1992). When the number of covariates is large the estimation of unknown parameters frequently raises problems. Then the interest usually focusses on data driven subset selection of relevant regressors. The sophisticated monitoring equipment which is now routinely used in many data collection processes makes it possible to collect data with a huge amount of regressors, even with considerably more explanatory variables than observations. One example is the analysis of microarray data of gene expressions. Here the typical tasks are to select variables and to classify samples into two or more alternative categories. Binary responses of this type may be handled within the framework of generalized linear models (Nelder and Wedderburn 1972) and are also considered in Toutenburg (1992).

There are several approaches to attain subset selection in generalized linear models. Shrinkage methods with L_1 norm penalties such as the lasso estimator are one

class of methods. The lasso estimator was introduced by Tibshirani (1996) for the linear model and extended to generalized linear models in Park and Hastie (2007). An alternative approach is componentwise boosting (see Bühlmann and Yu 2003). Boosting uses an ensemble of weak learners to improve the estimator. One obtains subset selection if each learner is restricted to use a subset of covariates.

One aspect in subset selection, highlighted by Zou and Hastie (2005), is the treatment of highly correlated covariates. Instead of choosing only one representative out of a group of highly correlated variables one could encourage strongly correlated covariates to be in or out of the model together. Zou and Hastie (2005) refer to it as the grouping effect.

In this paper we propose a new regularization method and a boosted version of it, which explicitly focus on the selection of groups. To reach this target we consider a correlation based penalty which uses correlation between variables as data driven weights for penalization. See also Tutz and Ulbricht (2006) for a similar approach to linear models. This new method and some of its main properties are described in Section 2. A boosted version of it that will be presented in Section 3 allows for variable selection. In Section 4 we use simulated and real data sets to compare our new methods with existing ones.

2 Penalized Maximum Likelihood Estimation

Consider a set of n independent one-dimensional observations $y_1, \ldots, y_n$ with densities from a simple exponential family type

$$f(y|\theta, \phi) = \exp\left\{\frac{y\theta - b(\theta)}{\phi} + c(y, \phi)\right\}, \tag{1}$$

where θ is the natural scalar parameter of the family, $\phi > 0$ is a nuisance or dispersion parameter, $b(\cdot)$ and $c(\cdot)$ are measurable functions. For each observation, also values of a set of p explanatory variables $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\top$ are recorded. They form a linear predictor $\eta_i = \beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta}^*$, where β_0 is a constant and $\boldsymbol{\beta}^* = (\beta_1, \ldots, \beta_p)^\top$ is a p dimensional parameter vector. It is assumed that the expectation of y_i is given by $\mu_i = h(\eta_i)$, where $h(\cdot)$ is a differentiable monotone response function and μ_i is the expectation of y_i.

Assuming that the dispersion parameter ϕ is known, we are interested in finding the unknown parameter vector $\boldsymbol{\beta} = (\beta_0, \boldsymbol{\beta}^{*\top})^\top$, which maximizes the corresponding log likelihood function

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left\{\frac{y_i\theta[h(\beta_0 + \mathbf{x}_i^\top\boldsymbol{\beta}^*)] + b(\theta[h(\beta_0 + \mathbf{x}_i^\top\boldsymbol{\beta}^*)])}{\phi_i} + c(y_i, \phi_i)\right\}. \tag{2}$$

Simple derivation yields the score function

$$s(\boldsymbol{\beta}) = \frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} \frac{y_i - b'(\theta_i)}{Var(y_i)} \frac{\partial h(\eta_i)}{\partial \eta} \mathbf{x}_i = \mathbf{X}^\top \mathbf{D} \boldsymbol{\Sigma}^{-1} (\mathbf{y} - \boldsymbol{\mu}), \tag{3}$$

where $\mathbf{X}^\top = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$,

$$\mathbf{D} = \operatorname{diag} \left\{ \frac{\partial h(\eta_1)}{\partial \eta}, \ldots, \frac{\partial h(\eta_n)}{\partial \eta} \right\}, \; \boldsymbol{\Sigma} = \operatorname{diag} \left\{ Var(y_1), \ldots, Var(y_n) \right\}.$$

The Fisher matrix is given by

$$F(\boldsymbol{\beta}) = -E \left[\frac{\partial^2 l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} \right] = E[s(\boldsymbol{\beta}) s(\boldsymbol{\beta})^\top] = \mathbf{X}^\top \mathbf{W} \mathbf{X}, \tag{4}$$

where $\mathbf{W} = \mathbf{D} \boldsymbol{\Sigma}^{-1} \mathbf{D}^\top$. The unknown parameter vector can be found iteratively by applying numerical methods for solving nonlinear equation systems, such as Newton-Raphson. Under weak assumptions the maximum likelihood estimator $\hat{\boldsymbol{\beta}}$ is consistent and asymptotically normal with asymptotic covariance matrix $Cov(\hat{\boldsymbol{\beta}}) = (\mathbf{X}^\top \mathbf{W} \mathbf{X})^{-1}$, see Fahrmeir and Kaufmann (1985).

In their seminal paper, Hoerl and Kennard (1970) show that the least squares estimate in the linear regression model tends to overestimate the length of the true parameter vector if the prediction vectors are not mutually orthogonal. Segerstedt (1992) shows similar effects when estimating generalized linear models. Early attempts of a generalizing ridge estimation were limited to logistic regression, see e.g. Anderson and Blair (1982), Schaefer, Roi, and Wolfe (1984) and Duffy and Santner (1989). Nyquist (1991) introduces ridge estimation of generalized linear models in the context of restricted estimation.

Since the maximum likelihood estimator of the unknown parameter vector has the tendency to overestimate length, it is advisable to fix its squared length. This restriction is formulated as constraint, so that we can use the Lagrangian approach. Formally, we solve the optimization problem

$$\hat{\boldsymbol{\beta}} = \arg \max_{\boldsymbol{\beta}} \left\{ l(\boldsymbol{\beta}) - P(\boldsymbol{\beta}) \right\}, \tag{5}$$

where

$$P(\boldsymbol{\beta}) = \lambda \|\boldsymbol{\beta}\|_2^2 = \lambda \sum_{j=1}^{p} \beta_j^2 \tag{6}$$

with $\|\boldsymbol{\beta}\|_2^2$ denoting the the squared L_2 norm of $\boldsymbol{\beta}$ and $\lambda > 0$ is a tuning parameter. Let $\hat{\boldsymbol{\beta}}_{ridge}(\lambda)$ denote the resulting GLM ridge estimator for given λ. Hence, $\hat{\boldsymbol{\beta}}_{ridge}(\lambda)$ is based on an L_2 penalty term.

Typically there exists a tuning parameter λ, so that the asymptotic mean squared error of the GLM ridge estimator is smaller than the asymptotic variance of the maximum likelihood estimator, for the proof see Segerstedt (1992). Nevertheless, the major drawback of $\hat{\boldsymbol{\beta}}_{ridge}(\lambda)$ is its lack in producing sparse solutions.

In the linear model setting the most important penalized regression approach that automatically includes subset selection is the lasso, as introduced by Tibshirani (1996). The L_1 based lasso penalty

$$P(\boldsymbol{\beta}) = \lambda \|\boldsymbol{\beta}\|_1 = \lambda \sum_{j=1}^{p} |\beta_j| \tag{7}$$

leads to regression fits that are sparse and interpretable, in the sense that many variables are "pruned" from the model. Shevade and Keerthi (2003) propose an L_1 penalization for logistic regression. Park and Hastie (2007) introduce a corrector-predictor algorithm for generalized linear models with lasso penalty. The main problem in using L_1 penalties within the GLM framework is the instability of coefficient estimates when some explanatory variables are strongly correlated. Furthermore, the solution might not be unique if some regressors are multicollinear. Therefore Park and Hastie (2007) modify the lasso penalty term to

$$P(\boldsymbol{\beta}) = \lambda_1 \|\boldsymbol{\beta}\|_1 + \frac{\lambda_2}{2} \|\boldsymbol{\beta}\|_2^2, \tag{8}$$

where $\lambda_1 > 0$ is an arbitrary tuning parameter and λ_2 is a fixed small positive constant. The elastic net penalty as introduced in Zou and Hastie (2005) is algebraically identical to (8), up to a rescaled tuning parameter of the L_2 penalty term. Using (8) in the way of Zou and Hastie (2005) requires simultaneous tuning parameter selection, e.g. by cross-validation, in two dimensions. This can be computationally cumbersome. One motivation Zou and Hastie (2005) give for the elastic net is its property to include groups of variables which are highly correlated. If variables are highly correlated, as for example gene expression in microarray data, the lasso selects only one out of the group whereas the elastic net catches "all the big fish", meaning that it selects the whole group.

In this paper we propose an alternative regularization procedure which aims at the selection of groups of correlated variables. In the simpler version it is based on a penalty that explicitly uses correlation between variables as weights. In the extended version boosting techniques are used for groups of variables. The correlation based

penalty is introduced as

$$\begin{aligned} P_c(\boldsymbol{\beta}) &= \lambda \sum_{i=1}^{p-1} \sum_{j>i} \left\{ \frac{(\beta_i - \beta_j)^2}{1-\varrho_{ij}} + \frac{(\beta_i + \beta_j)^2}{1+\varrho_{ij}} \right\} \\ &= 2\lambda \sum_{i=1}^{p-1} \sum_{j>i} \frac{\beta_i^2 - 2\varrho_{ij}\beta_i\beta_j + \beta_j^2}{1-\varrho_{ij}^2} \end{aligned} \tag{9}$$

where ϱ_{ij} denotes the (empirical) correlation between the ith and the jth predictor. It is designed to focus on the grouping effect, that is highly correlated effects show comparable values of estimates ($|\hat{\beta}_i| \approx |\hat{\beta}_j|$) with the sign being determined by positive or negative correlation. For strong positive correlation ($\varrho_{ij} \to 1$) the first term becomes dominant having the effect that estimates for β_i, β_j are similar ($\hat{\beta}_i \approx \hat{\beta}_j$). For strong negative correlation ($\varrho_{ij} \to -1$) the second term becomes dominant and $\hat{\beta}_i$ will be close to $-\hat{\beta}_j$. Consequently, for weakly correlated data the performance is quite close to the ridge penalty. The correlation based penalty (9) can be written as a quadratic form

$$P_c(\boldsymbol{\beta}) = \lambda \boldsymbol{\beta}^\top \mathbf{M} \boldsymbol{\beta}, \tag{10}$$

where $\mathbf{M} = (m_{ij})$ is given by

$$m_{ij} = \begin{cases} 2\sum_{s \neq i} \frac{1}{1-\varrho_{is}^2}, & i = j, \\ -2\frac{\varrho_{ij}}{1-\varrho_{ij}^2}, & i \neq j. \end{cases}$$

We denote the resulting penalized maximum likelihood estimator of the unknown coefficient vector as $\hat{\boldsymbol{\beta}}_c$ and refer to it in the following as GLM PenalReg estimator.

Due to the additive structure between log likelihood function and the penalty term the computation of the correlation based penalized estimator, abbreviated by GLM PenalReg, is easily done by using the score function and Fisher matrix of the log likelihood function. For the penalized log likelihood with $P_c(\boldsymbol{\beta}) = \lambda \boldsymbol{\beta}^\top \mathbf{M} \boldsymbol{\beta}$ one obtains

$$l_p(\boldsymbol{\beta}) = l(\boldsymbol{\beta}) - \frac{\lambda}{2} \boldsymbol{\beta}^\top \mathbf{M} \boldsymbol{\beta}, \tag{11}$$

where we use a rescaling of λ for computational simplicity. Hence, the penalized score is

$$s_p(\boldsymbol{\beta}) = \frac{\partial l_p(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = s(\boldsymbol{\beta}) - \lambda \mathbf{M} \boldsymbol{\beta} = \mathbf{X}^\top \mathbf{D} \boldsymbol{\Sigma}^{-1} (\mathbf{y} - \boldsymbol{\mu}) - \lambda \mathbf{M} \boldsymbol{\beta}, \tag{12}$$

and the penalized Fisher matrix is given by

$$F_p(\boldsymbol{\beta}) = -E \left[\frac{\partial s_p(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}^\top} \right] = \mathbf{X}^\top \mathbf{W} \mathbf{X} + \lambda \mathbf{M}, \tag{13}$$

As in non-penalized maximum likelihood estimation we need to solve a nonlinear system of equations. In the same way as the GLM ridge estimator the GLM PenalReg estimator can be written as an iteratively re-weighted least squares estimator, given by

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{c}}{}^{(k+1)} = (\mathbf{X}^\top \mathbf{W}\mathbf{X} + \lambda \mathbf{M})^{-1}\mathbf{X}^\top \mathbf{W}\tilde{\mathbf{y}}^{(k)}, \tag{14}$$

where $\tilde{\mathbf{y}}^{(k)} = \mathbf{X}\hat{\boldsymbol{\beta}}_{\boldsymbol{c}}{}^{(k)} + \mathbf{D}^{-1}(\mathbf{y} - \boldsymbol{\mu})$.

Based on a first order Taylor approximation one obtains the asymptotic covariance matrix

$$Cov[\hat{\boldsymbol{\beta}}_{\boldsymbol{c}}(\lambda)] = (\mathbf{X}^\top \mathbf{W}\mathbf{X} + \lambda \mathbf{M})^{-1}\mathbf{X}^\top \mathbf{W}\mathbf{X}(\mathbf{X}^\top \mathbf{W}\mathbf{X} + \lambda \mathbf{M})^{-1}. \tag{15}$$

Note that we get similar results for the generalized ridge estimator $\boldsymbol{\beta}_{ridge}(\lambda)$ when we substitute the identity matrix for the penalty matrix $\mathbf{M}$, see Segerstedt (1992) for details. A systematic report on mean squared error comparisons of competing biased estimators for the linear model is given in Trenkler and Toutenburg (1990). For performance comparisons in several simulation and practical data situations we refer to section 4.

3 Generalized Blockwise Boosting

The main drawback of the correlation based penalized estimator is its lack of sparsity. In particular when high dimensional data such as microarray data are considered one wants to select an appropriate subset of regressors. One method that is able to overcome this disadvantage is componentwise boosting as introduced by Bühlmann and Yu (2003). They propose to update in one boosting step only the component that maximally improves the fit.

Boosting methods are multiple prediction schemes that average estimated predictions from re-weighted data. With its origins in the machine learning community the first major field of applications was binary classification. The link between boosting and a gradient descent optimization technique in function space as outlined in Breiman (1998) provided the application of boosting methods in other contexts than classification. Friedman (2001) developed the L_2Boost algorithm for a linear base learner, an optimization algorithm with squared error loss function for application in regression, which provides the foundations of componentwise boosting. For a detailed overview on boosting see e.g. Meir and Rätsch (2003). Componentwise likelihood based boosting applied to the generalized ridge estimator is described in Tutz and Binder (2007). The base learner of this boosting algorithm is the first step of the Fisher scoring algorithm.

Let $S^{(m)} \subset \{0, 1, \ldots, p\}$ denote the index set of the variables considered in the m-th step, where the index 0 refers to the intercept term of the predictor. The input data

to the base learner are $\{(\mathbf{x}_1, r_1), \ldots, (\mathbf{x}_n, r_n)\}$, where $r_i = y_i - \hat{\mu}_i^{(m-1)}$ $(i = 1, \ldots, n)$ denotes the residual between the origin response y_i and the estimated response from the previous boosting step.

The basic concept is to choose within the m-th step of the iterative procedure the subset of variables which provides the best improvement to the fit. In componentwise maximum likelihood based boosting it is common to use the deviance as a measure of goodness-of-fit. We choose the Akaike information criterion (AIC) rather than the deviance, because it includes an automatic penalization of large subsets.

The following algorithm GenBlockBoost is a boosted version of the correlation based penalized estimate.

Algorithm GenBlockBoost

Step 1: (Initialization)

Fit the model $\mu_i = h(\beta_0)$ by iterative Fisher scoring yielding $\hat{\boldsymbol{\beta}}^{(0)} = (\hat{\beta}_0, 0, \ldots, 0)^\top$. Set $\hat{\boldsymbol{\eta}}^{(0)} = X\hat{\boldsymbol{\beta}}^{(0)}, \hat{\boldsymbol{\mu}}^{(0)} = h(\hat{\boldsymbol{\eta}}^{(0)})$.

Step 2: (Iteration)

For $m = 1, 2, \ldots$

(a) *Find an appropriate order of regressors according to their improvements of fit*

For $j \in \{0, \ldots, p\}$ compute the estimates based on one step Fisher scoring

$$\hat{b}_{\{j\}} = (\mathbf{x}_{\{j\}}^\top W(\hat{\boldsymbol{\eta}}^{(m-1)})\mathbf{x}_{\{j\}} + \lambda)^{-1}\mathbf{x}_{\{j\}}^\top W(\hat{\boldsymbol{\eta}}^{(m-1)})D(\hat{\boldsymbol{\eta}}^{(m-1)})^{-1}(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(m-1)}),$$

yielding $\hat{b}_{j_0}, \ldots, \hat{b}_{j_p}$ such that $Dev(\hat{b}_{j_0}) \leq \ldots \leq Dev(\hat{b}_{j_p})$, where

$$Dev(\hat{b}_{j_k}) = 2\sum_{i=1}^{n}\left\{l_i(y_i) - l_i\left[h(\hat{\eta}_i^{(m-1)} + x_{ij_k}\hat{b}_{j_k})\right]\right\}, \quad k = 0, 1, \ldots, p.$$

(b) *Find a suitable number of regressors to update*

For $r = 0, \ldots, p$

With $S_r = \{j_0, \ldots, j_r\}$ we compute the estimates based on one step Fisher scoring

$$\begin{aligned}\hat{\mathbf{b}}_{S_r} &= (\mathbf{X}_{S_r}^\top W(\hat{\boldsymbol{\eta}}^{(m-1)})\mathbf{X}_{S_r} + \lambda_{|S_r|}\mathbf{M}_{S_r})^{-1}\mathbf{X}_{S_r}^\top W(\hat{\boldsymbol{\eta}}^{(m-1)}) \\ &\quad \times D(\hat{\boldsymbol{\eta}}^{(m-1)})^{-1}(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(m-1)}),\end{aligned}$$

yielding estimates $\hat{\mathbf{b}}_{S_r}$ and AIC criterion $AIC(\hat{\mathbf{b}}_{S_r})$.

(c) *Selection*

Select the subset of variables which has the best fit, yielding

$$S^{(m)} = \arg\min_{S_r} AIC(\hat{\mathbf{b}}_{S_r}).$$

(d) *Refit*

The parameter vector is updated by

$$\hat{\beta}_j^{(m)} = \begin{cases} \hat{\beta}_j^{(m-1)} + \hat{b}_j, & \text{if } j \in S^{(m)}, \\ \hat{\beta}_j^{(m-1)}, & \text{otherwise,} \end{cases}$$

yielding $\hat{\boldsymbol{\beta}}^{(m)} = (\hat{\beta}_1^{(m)}, \ldots, \hat{\beta}_p^{(m)})^\top, \hat{\boldsymbol{\eta}}^{(m)} = \mathbf{X}\hat{\boldsymbol{\beta}}^{(m)}, \hat{\boldsymbol{\mu}}^{(m)} = h(\hat{\boldsymbol{\eta}}^{(m)})$.

The number of possible combinations of regressors is 2^p. Due to computational limitation we cannot apply a full search for the best subset. Therefore in a first step of each boosting iteration we order the regressors according to their individual potential improvement to the fit. This improvement is measured by the (potential) deviance

$$Dev(\hat{b}_j) = 2\sum_{i=1}^{n} \left\{ l_i(y_i) - l_i\left[h(\hat{\eta}_i^{(m-1)} + x_{ij}\hat{b}_j)\right]\right\}, \quad j = 0, \ldots, p,$$

where $x_{i0} = 1$ for all $i = 1, \ldots, n$.

For making the base learner a weak learner, so that only a small change in parameter estimates occurs within one boosting iteration, the tuning parameter λ is chosen very large. This also leads to more stable estimates. The price to pay for this choice is an increase in computation time when the value of the tuning parameter becomes larger.

For subsets S that contain only one variable the correlation based penalty (10) cannot be used directly. In those cases we define the penalty by the ridge type penalty $P_{c,\{j\}} = \lambda\beta_j^2$.

Within the algorithm the correlation based estimator is used for subsets of varying size. The tuning parameter λ that is used has to be adapted to the number of refitted regressors. If one considers the case of uncorrelated variables the penalty for all variables reduces to $P_c(\boldsymbol{\beta}) = 2\lambda(p-1)\sum_{i=1}^{p}\beta_i^2$ which equals the ridge penalty with tuning parameter $2\lambda(p-1)$. Thus $\lambda_{|S_r|}$ in step 2b of the GenBlockBoost algortihm is chosen by $\lambda_{|S_r|} = \lambda(|S_r| - 1)$, where $|S_r|$ denotes the number of refitted regressors.

In order to avoid overfitting, a stopping criterion is needed for estimating the optimal number of boosting iterations. We use the AIC criterion

$$AIC(\hat{\boldsymbol{\beta}}^{(m)}) = Dev_m + 2tr(\mathbf{H}_m), \tag{16}$$

with

$$Dev_m = 2\sum_{i=1}^{n}\left[l_i(y_i) - l_i(\hat{\mu}_i^{(m)})\right].$$

An approximation of the hat matrix is given by

$$\mathbf{H}_m = \sum_{j=0}^{m}\mathbf{M}_j\prod_{i=0}^{j-1}(I - \mathbf{M}_i),$$

so that $\hat{\boldsymbol{\mu}}^{(m)} \approx \mathbf{H}_m\mathbf{y}$, where

$$\mathbf{M}_l = \boldsymbol{\Sigma}_m^{1/2}\mathbf{W}_m^{1/2}\mathbf{X}_{S^{(m)}}(\mathbf{X}_{S^{(m)}}^\top\mathbf{W}_m\mathbf{X}_{S^{(m)}} + \lambda\mathbf{M}_{S^{(m)}})^{-1}\mathbf{X}_{S^{(m)}}^\top\mathbf{W}_{(m)}^{1/2}\boldsymbol{\Sigma}_m^{-1/2}$$

and $\mathbf{M}_0 = \frac{1}{n}1_n1_n^\top$. See Tutz and Leitenstorfer (2007) for the derivation of this approximation. An estimate of the sufficient number of boosting iterations is

$$m^* = \arg\min_m AIC(\hat{\boldsymbol{\beta}}^{(m)}).$$

In the next section we investigate the performance of the correlation based penalized estimator for GLMs and the GenBlockBoost algorithm in several simulation and data settings.

4 Simulations and real data example

In the simulations, we consider predictors which are given in 10 blocks, each block contains q variables, resulting in $p = 10q$ variables. All variables have unit variances. The correlations between x_i and x_j are $\varrho^{|i-j|}$ if x_i and x_j belong to the same block, otherwise they are given by a truncated $N(0, 0.1^2)$ distribution. For the true predictor η we choose the set V of all covariates that belong to three randomly chosen blocks so that

$$\eta = \mathbf{x}^\top\boldsymbol{\beta},$$

where $\mathbf{x} = (x_1, \ldots, x_p)^\top$ and $\boldsymbol{\beta} = c \cdot (\beta_1, \ldots, \beta_p)^\top$ is determined by

$$\beta_j \sim N(1, 1) \text{ for } j \in V, \quad \beta_j = 0 \text{ otherwise.}$$

That means each variable included in one of the chosen blocks is considered as relevant. Note that $\beta_0 = 0$ in all simulations, but all methods are allowed to include a nonzero intercept in their vector of estimated coefficients. The final response y corresponding to the expected value of the response $\mu = E(y|\mathbf{x}) = h(\eta)$, where $h(\eta) = \exp(\eta)/(1 + \exp(\eta))$ is drawn from a binomial distribution $B(\mu, 1)$. The constant c is chosen so that the signal-to-noise ratio

$$\text{signal-to-noise ratio} = \frac{\sum_{i=1}^n(\mu_i - \bar{\mu})^2}{\sum_{i=1}^n Var(y_i)},$$

with $\bar{\mu} = \frac{1}{n}\sum_{i=1}^{n}\mu_i$, is (approximately) equal to one. We use the Newton algorithm to find c in this case. The estimation of unknown parameters is based on 100 training data observations. The evaluation uses 1000 test data observations. We use an additional independent validation data set consisting of 100 observation to determine the tuning parameters.

We compare the GLM PenalReg estimator and the GenBlockBoost algorithm with the maximum likelihood estimator (ML), L_2 penalized maximum likelihood estimation (ridge), L_1 penalized maximum likelihood estimation (lasso) and a boosted version of the L_2 penalized maximum likelihood estimator (GenRidgeBoost). For further details on the GenRidgeBoost algorithm see Tutz and Binder (2007). The computation of the L_1 penalized maximum likelihood estimator is done with the R package glmpath by Mee Young Park and Trevor Hastie.

The performance of data fitting is measured by the deviance and the deviation between estimated and true parameter vector. The latter is defined as

$$MSE_\beta = |\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}|^2. \tag{17}$$

Besides the prediction performance as an important criterion for comparison of methods the variables included into the final model are of special interest to practitioners. The final model should be as parsimonious as possible but all relevant variables should be included. We use the criteria *hits* and *false positives* to evaluate the identification of relevant variables. Hits refers to the number of correctly identified influential variables, false positives is the number of non-influential variables dubbed influential.

The simulation results are given in Table 1, 2, 3 and Figures 1 and 2. GenBlockBoost has the best prediction performance almost all the time. Considering the fit of true parameters PenalReg performs very good, but GenBlockBoost shows good results among the variable selecting procedures for small and medium sized blocks. GlmPath performs better for huge blocks. In the hits and false positives analysis GenBlockBoost clearly outperforms GlmPath and also chooses more relevant covariates than GenRidgeBoost. GenRidgeBoost generally tends to more parsimonious models, hence its median number of false positives is smaller in comparison to GenBlockBoost.

For an application to real data we use the leukemia cancer gene expression data set as described in Golub et al. (1999). In cancer treatment it is important to target specific therapies to pathogenetically distinct tumor types, to gain a maximum of efficacy and a minimum of toxicity. Hence, distinguishing different tumor types is critical for successful treatment. The challenge of the leukemia data set is to classify acute leukemia into those arising from lymphoid precursors (acute lymphoblastic leukemia, ALL) and those arising from myeloid precursors (acute myeloid leukemia, AML), based on the simultaneous expression monitoring of 7129 genes using DNA microarrays. The data set consists of 72 samples, out of which 47 observations are ALL and 25 are AML. We use 20 random splits into a training and an independent test sample of sizes 38 and 34, respectively.

		ML	Ridge	PenalReg	GenRidgeBoost	GenBlockBoost	GlmPath (Lasso)
$q = 3$	$\varrho = 0.95$	17983.21	875.22	**866.16**	965.81	901.05	904.78
	$\varrho = 0.8$	16791.35	928.75	923.54	916.89	**907.23**	940.89
	$\varrho = 0.5$	15497.62	965.58	966.91	890.67	**881.59**	936.87
$q = 5$	$\varrho = 0.95$	20035.41	894.60	892.68	891.95	**851.78**	908.84
	$\varrho = 0.8$	21152.08	939.29	934.00	906.33	**897.05**	949.74
	$\varrho = 0.5$	19842.48	1005.99	1011.70	993.47	**958.38**	1007.23
$q = 10$	$\varrho = 0.95$	-	871.39	**854.36**	868.69	859.35	907.84
	$\varrho = 0.8$	-	970.10	947.54	937.15	**915.58**	982.49
	$\varrho = 0.5$	-	1099.91	1085.18	1119.54	1110.80	**1083.11**

TABLE 1: *Median deviances for simulated data based on 20 replications.*

		ML	Ridge	PenalReg	GenRidgeBoost	GenBlockBoost	GlmPath (Lasso)
$q = 3$	$\varrho = 0.95$	423640.00	2.19	**1.80**	2.69	2.56	3.55
	$\varrho = 0.8$	106086.80	1.98	1.68	1.89	**1.62**	1.92
	$\varrho = 0.5$	47861.17	2.00	2.04	**1.30**	1.43	1.63
$q = 5$	$\varrho = 0.95$	345348.10	1.60	**1.51**	3.62	2.07	3.71
	$\varrho = 0.8$	77118.91	2.35	1.97	2.27	**1.95**	2.80
	$\varrho = 0.5$	33738.83	2.15	2.19	2.12	**1.78**	2.18
$q = 10$	$\varrho = 0.95$	-	1.43	**1.08**	2.87	2.40	2.22
	$\varrho = 0.8$	-	2.02	1.55	2.72	**2.49**	2.46
	$\varrho = 0.5$	-	2.51	**2.38**	2.67	2.79	2.58

TABLE 2: *Median MSE_β for simulated data based on 20 replications.*

		ML	Ridge	PenalReg	GenRidgeBoost	GenBlockBoost	GlmPath (Lasso)
$q = 3$	$\varrho = 0.95$	9/22	9/22	9/22	4/1	6/3	5/7
	$\varrho = 0.8$	9/22	9/22	9/22	5/1	5/2	6/8
	$\varrho = 0.5$	9/22	9/22	9/22	6/2	6/3	7/10
$q = 5$	$\varrho = 0.95$	15/36	15/36	15/36	6/3	12/4	6/9
	$\varrho = 0.8$	15/36	15/36	15/36	7/3	11/6	8/9
	$\varrho = 0.5$	15/36	15/36	15/36	8/3	9/5	9/9
$q = 10$	$\varrho = 0.95$	-	30/71	30/71	9/2	17/8	8/5
	$\varrho = 0.8$	-	30/71	30/71	10/2	16/5	12/10
	$\varrho = 0.5$	-	30/71	30/71	12/2	17/9	14/12

TABLE 3: *Median hits/false positives for simulated data based on 20 replications.*

Besides the test deviance

$$Dev_{test} = 2 \sum_{i=1}^{n_{test}} \left[l_i(y_{i,test}) - l_i(\hat{\mu}_{i,test}) \right], \tag{18}$$

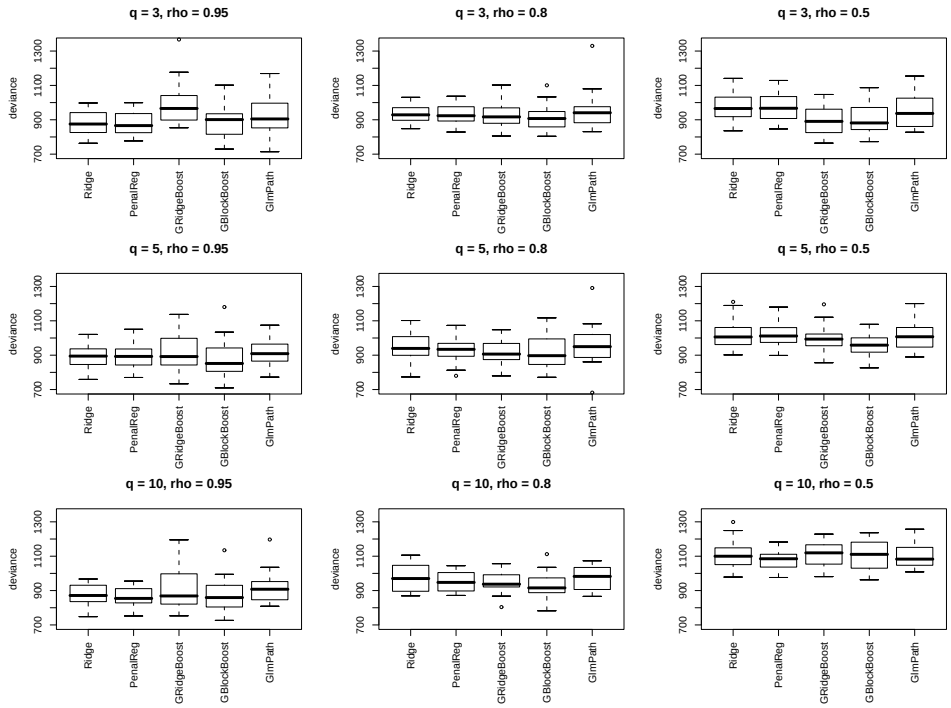

FIGURE 1: *Deviances for various estimators for the simulations.*

which is based on the test sample, we consider the number of genes identified as relevant variables. Since the main focus is on classification we focus on the numbers of correctly classified respective misclassified observations in the test data set as performance measures for discrimination.

Due to the 20 random splits we consider the median performance results which are given in Table 4. All three considered algorithms show quite similar performances. At the median number of correctly classified types of leukemia, GenRidgeBoost is slightly better for the ALL class, GenBlockBoost is slightly better for the AML class. When considering the overall misclassification GlmPath has the best discrimination power. Due to the test deviance, the test data fits best to the model estimated by GenBlockBoost. Here, the GenRidgeBoost estimator is only poor. When we consider the number of selected genes GenRidgeBoost is slightly more sparse than the competitors.

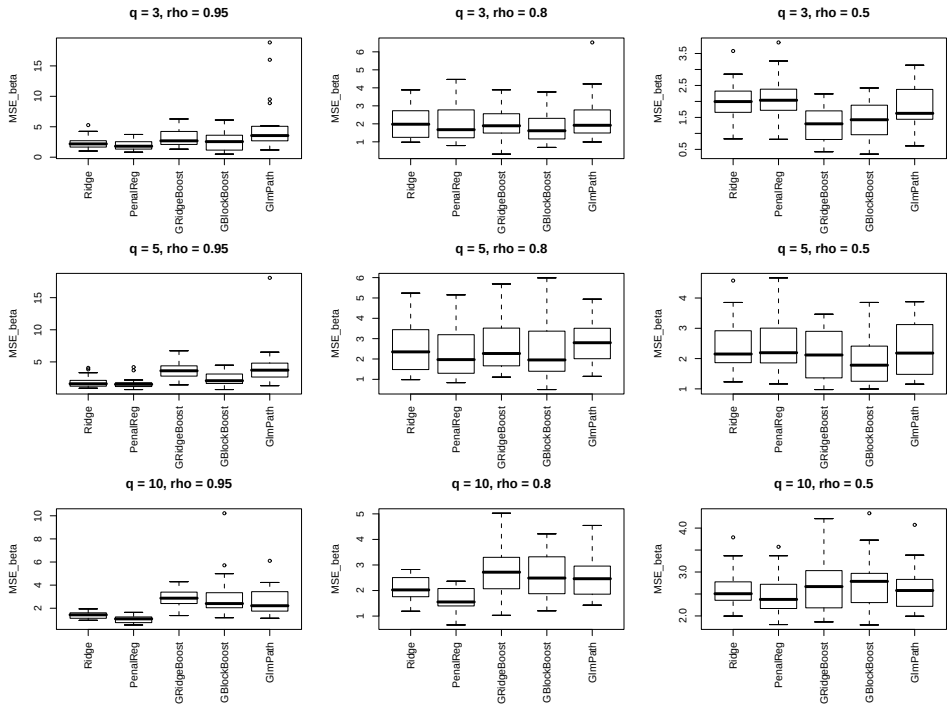

FIGURE 2: *MSE_β for various estimators for the simulations.*

Performance measure	GenBlockBoost	GlmPath	GenRidgeBoost
ALL correctly classified	9	10	11
AML correctly classified	21	20	20
misclassification	5	3	5
Dev_{test}	17.75	19.11	84.98
No. of genes used	11	10	9

TABLE 4: *Median performance results for the leukemia cancer gene expression data for 20 random splits into 38 learning data and 34 test data.*

5 Concluding Remarks

We presented two approaches for parameter estimation in generalized linear models with many covariates. The GLM PenalReg estimator gives special attention to the grouping effect, the GenBlockBoost algorithm moreover put additional attention on subset selection. The simulations demonstrate the competitive data fitting perfor-

mance and the small deviation between estimated and true parameter vectors. The GenBlockBoost algorithm is slightly less sparse than the GenRidgeBoost algorithm but this is a consequence of the more tightly focused grouping effect. Nevertheless the correct identification of relevant variables is quite good. As a result, the GenBlockBoost estimator can be seen as a strong competitor in the field of subset selection in generalized linear models.

Both methods may be extended to the case of multivariate generalized linear models, such as with multinomial response. Furthermore, some further theoretical aspects on MSE comparisons with the GLM ridge estimator might by interesting. Here, Trenkler and Toutenburg (1990) provides an initial point for the challenging application to generalized linear models.

References

Anderson, J. A. and V. Blair (1982). Penalized maximum likelihood estimation in logistic regression and discrimination. *Biometrika 69*, 123–136.

Breiman, L. (1998). Arcing classifiers. *Annals Of Statistics 26*, 801–849.

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

Duffy, D. E. and T. J. Santner (1989). On the small sample properties of restricted maximum likelihood estimators for logistic regression models. *Communication in Statistics, Theory & Methods 18*, 959–989.

Fahrmeir, L. and H. Kaufmann (1985). Consistency and asymptotic normality of the maximum likelihood estimator in generalized linear models. *The Annals of Statistics 13*, 342–368.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Ann. Statist. 29*, 1189–1232.

Golub, T. R., D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science 286*, 531–537.

Hoerl, A. E. and R. W. Kennard (1970). Ridge regression: Bias estimation for nonorthogonal problems. *Technometrics 12*, 55–67.

Meir, R. and G. Rätsch (2003). An introduction to boosting and leveraging. In S. Mendelson and A. Smola (Eds.), *Advanced Lectures on Machine Learning*, pp. 119–184. New York: Springer.

Nelder, J. A. and R. W. M. Wedderburn (1972). Generalized linear models. *Journal of the Royal Statistical Society A 135*, 370–384.

Nyquist, H. (1991). Restricted estimation of generalized linear models. *Applied Statistics 40*, 133–141.

Park, M. Y. and T. Hastie (2007). An l1 regularization-path algorithm for generalized linear models. *JRSS*.

Schaefer, R. L., L. D. Roi, and R. A. Wolfe (1984). A ridge logistic estimate. *Communication in Statistics, Theory & Methods 13*, 99–113.

Segerstedt, B. (1992). On ordinary ridge regression in generalized linear models. *Communication in Statistics, Theory & Methods 21*, 2227–2246.

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

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

Toutenburg, H. (1992). *Lineare Modelle – Theorie und Anwendungen*. Heidelberg: Physica-Verlag.

Trenkler, G. and H. Toutenburg (1990). Mean squared error matrix comparisons between biased estimators – an overview of recent results. *Statistical Papers 31*, 165–179.

Tutz, G. and H. Binder (2007). Boosting ridge regression. *Computational Statistics & Data Analysis (to appear)*.

Tutz, G. and F. Leitenstorfer (2007). Generalized smooth monotonic regression in additive modeling. *Journal of Computational and Graphical Statistics 16*, 165–188.

Tutz, G. and J. Ulbricht (2006). Penalized regression with correlation based penalty. Discussion Paper 486, SFB 386, Universität München.

Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society B 67*, 301–320.