LMU
LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

INSTITUT FÜR STATISTIK

Faisal M. Zahid & Gerhard Tutz

Multinomial Logit Models with Implicit Variable Selection

Technical Report Number 089, 2010
Department of Statistics
University of Munich

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

Multinomial Logit Models with Implicit Variable Selection

Faisal Maqbool Zahid[a,*], Gerhard Tutz[b]

[a]Ludwig-Maximilians-University Munich, Ludwigstrasse 33, D-80539 Munich, Germany
[b]Ludwig-Maximilians-University Munich, Akademiestraße 1, D-80799 Munich, Germany

Abstract

Multinomial logit models which are most commonly used for the modeling of unordered multi-category responses are typically restricted to the use of few predictors. In the high-dimensional case maximum likelihood estimates frequently do not exist. In this paper we are developing a boosting technique called *multinomBoost* that performs variable selection and fits the multinomial logit model also when predictors are high-dimensional. Since in multi-category models the effect of one predictor variable is represented by several parameters one has to distinguish between variable selection and parameter selection. A special feature of the approach is that, in contrast to existing approaches, it selects variables not parameters. The method can distinguish between mandatory predictors and optional predictors. Moreover, it adapts to metric, binary, nominal and ordinal predictors. Regularization within the algorithm allows to include nominal and ordinal variables which have many categories. In the case of ordinal predictors the order information is used. The performance of boosting technique with respect to mean squared error, prediction error and the identification of relevant variables is investigated in a simulation study. For two real life data sets the results are also compared with the Lasso approach which selects parameters.

Key words: Logistic regression, Multinomial logit, Variable selection, Side constraints, Likelihood-based boosting, Penalization, Hit rate, False alarm rate.

1. Introduction

The multinomial logit model is the most frequently used model in regression analysis with categorical response. Typically, the maximum likelihood method is used for estimating the parameters. However, the use of maximum likelihood estimation severely limits the number of predictors in the multinomial logit models. As the number of covariates increases relative to the sample size, problems with the convergence of parameter estimates arise and the usual ML estimates will not exist for $p > n$. To overcome the problem, one alternative is to rely on penalization

[*]Corresponding author. Tel.: ++49 89 2180 6408; fax.: ++49 89 2180 5040.
Email addresses: faisal-maqbool.zahid@stat.uni-muenchen.de (Faisal Maqbool Zahid), tutz@stat.uni-muenchen.de (Gerhard Tutz)

techniques. One of the oldest penalization techniques is ridge regression which was extended to generalized linear models (GLM) by Nyquist (1991), Segerstedt (1992). In contrast to ridge regression which shrinks the parameter estimates towards zero but does not enforce subset selection, Lasso (Tibshirani (1996)) does not only shrink the parameter estimates but also enforces subset selection by setting some of the parameter estimates exactly equal to zero. An extension to GLMs was proposed by Park and Hastie (2007). However, for multicategory responses, not much literature is available. Multinomial logistic regression with Lasso type estimates was considered by Krishnapuram et al. (2005). Friedman et al. (2010) considered L1 (Lasso), L2 (ridge) penalties and the elastic net (mixture of the L1 and L2 penalty). Zahid and Tutz (2009) used ridge regression with symmetric side constraints, which makes the ridge estimates invariant to the choice of a reference category. A general alternative to maximum likelihood estimation is likelihood-based boosting. Boosting was originally developed in the machine learning community to improve classification (e.g., Schapire (1990) and Freund and Schapire (1996)). Friedman et al. (2000) showed that boosting can also be viewed as an approximation to additive modeling using the appropriate likelihood function. Bühlmann and Yu (2003) used the L_2 loss function instead of the LogitBoost cost function within the context of linear models. Bühlmann (2006) showed the relation to Lasso which also does variable selection and shrinkage without making any assumptions about the correlation structure of the predictors. For an overview on boosting see Bühlmann and Hothorn (2007). Likelihood-based boosting based on one step of Fisher scoring for variable selection in generalized additive models (GAM) was proposed by Tutz and Binder (2006).

In this article we are using the likelihood based boosting technique with one step of Fisher scoring for variable selection in multinomial logit models. For the weak learners we are using the ridge penalty. When seeking for a parsimonious model, one frequently considers some of the covariates as an essential part of the model. For example, in a treatment study one is interested in particular in the treatment effect, which is considered a mandatory covariate. The other covariates which might be of relevance are considered as optional. Similar to Tutz and Binder (2007) our method distinguishes between mandatory and optional predictors.

When working with categorical covariates it is essential that selection does not refer to parameters but to covariates (comprising the group of parameters associated with the categorical covariate). Consequently, our approach performs variable selection in terms of covariates rather than parameters. For this purpose our approach differentiates among the categorical predictors that contain a group of parameters (for each logit) in the parameter space and those predictors having only one parameter for each logit of the multinomial logit model e.g., binary or metric predictors. Moreover, in the case of ordinal covariate(s), rather than penalizing the parameters, our approach penalizes the differences between the paramters of the adjacent categories.

In Section 2 the side constraints for the multinomial logit model and the regularization for different types of covariates are discussed . Boosting is discussed in Section 3, Section 4 gives empirical results of a simulation study. The algorithm is applied to real life data set and the results are compared with those obtained from the Lasso approach in Section 5.

2. Predictor Space, Side Constraints and Regularization

In this section we describe the different types of candidate predictors (candidates to become part of a parsimonious logit model) which will be used in Section 3, and how they are incorporated into the boosting algorithm for subset selection. For simplicity, we assume in this section that the multinomial logit model has only one predictor with K parameters to be estimated for each of the k categories of the response variable. If the predictor is metric or binary then $K = 1$, and $K > 1$ if the predictor is a multicategory variable with the $K + 1$ categories labeled as $1, \ldots, K, K + 1$. Although the intercept is part of the model, for simplicity in the following the intercept is omitted.

2.1. Side Constraints for the Multinomial Logit Model

Let the response variable $Y \in \{1, \ldots, k\}$ have k possible categories. The generic form of the multinomial logit model is

$$P(Y = r|\mathbf{x}) = \frac{exp(\mathbf{x}^T\boldsymbol{\beta}_r)}{\sum_{s=1}^{k} exp(\mathbf{x}^T\boldsymbol{\beta}_s)} = \frac{exp(\eta_r)}{\sum_{s=1}^{k} exp(\eta_s)}, \qquad (1)$$

where $\boldsymbol{\beta}_r^T = (\beta_{r1}, \ldots, \beta_{rK})$. Since parameters $\boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_k^T$ are not identifiable, for the identifiability of parameters one has to specify additional constraints. One most commonly used side constraint is to choose one of the response categories as reference category. If category k is chosen as reference, then one sets

$$\boldsymbol{\beta}_k^T = (0, \ldots, 0) \qquad \text{yielding} \qquad \eta_k = 0.$$

Of course, any category can be chosen as a reference category. With category k as the reference category, the model is

$$P(Y = r|\mathbf{x}) = \frac{exp(\mathbf{x}^T\boldsymbol{\beta}_r)}{1 + \sum_{s=1}^{q} exp(\mathbf{x}^T\boldsymbol{\beta}_s)} \qquad \text{for } r = 1, \ldots, q. \qquad (2)$$

with $q = k - 1$. Alternatively one can work with the symmetric side constraint. With $\boldsymbol{\beta}_s^*$ denoting the parameter vector for category s, it is given by

$$\sum_{s=1}^{k} \boldsymbol{\beta}_s^* = \mathbf{0}. \qquad (3)$$

Then the multinomial logit model has the form

$$P(Y = r|\mathbf{x}) = \frac{exp(\mathbf{x}^T\boldsymbol{\beta}_r^*)}{\sum_{s=1}^{k} exp(\mathbf{x}^T\boldsymbol{\beta}_s^*)} = \frac{exp(\eta_r^*)}{\sum_{s=1}^{k} exp(\eta_s^*)} \qquad \text{for } r = 1, \ldots, q \qquad (4)$$

With symmetric side constraint, the median response defined by the geometric mean can be viewed as the reference category. The parameters $\boldsymbol{\beta}^*$ have a different interpretation than $\boldsymbol{\beta}$ obtained with a reference category constraint. Here $\boldsymbol{\beta}_r^*$ reflects the effects of $\mathbf{x}$ on the logits when $P(Y = r|\mathbf{x})$ is compared to the median response $GM(\mathbf{x}) = \sqrt[k]{\prod_{s=1}^{k} P(Y = s|\mathbf{x})}$.

In the following, let $\boldsymbol{\beta}^T = (\boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_q^T, \mathbf{0})$ and $\boldsymbol{\beta}^{*T} = (\boldsymbol{\beta}_1^{*T}, \ldots, \boldsymbol{\beta}_k^{*T})$ represent the parameter vectors for the multinomial logit model under the reference category side constraint ($\boldsymbol{\beta}_k = \mathbf{0}$) and symmetric side constraint ($\sum_{s=1}^{k} \boldsymbol{\beta}_s^* = \mathbf{0}$), respectively. There is a one-to-one correspondence between the parameters $\boldsymbol{\beta}$ and $\boldsymbol{\beta}^*$. One way of giving the transformation is by considering $\boldsymbol{\beta}^T_{.j} = (\beta_{1j}, \ldots, \beta_{k-1,j})$, and $\boldsymbol{\beta}^{*T}_{.j} = (\beta^*_{1j}, \ldots, \beta^*_{k-1,j})$, $j = 1, \ldots, K$, which denote parameter vectors for a particular variable with reference category k or symmetric side constraints respectively, then

$$\boldsymbol{\beta}^*_{.j} = \mathbf{T}\boldsymbol{\beta}_{.j} \qquad \text{for } j = 1, \ldots, K, \tag{5}$$

where $\mathbf{T}$ is a $q \times q$ matrix (q=k-1) with diagonal elements $\frac{q}{k}$ and off-diagonal elements as $-\frac{1}{k}$. The matrix $\mathbf{T}^{-1}$, then, has the diagonal enteries as 2 and all off-diagonal elements are 1. For likelihood (or penalized likelihood) estimation, the complete design matrix $\mathbf{X}$ of order $q(n \times K)$ is given by $\mathbf{X}^T = [\mathbf{X}_1 \ \mathbf{X}_2 \ \ldots \ \mathbf{X}_n]$. The matrix $\mathbf{X}_i$ is a $q \times K$ design matrix composed of $\mathbf{x}_i$ and is given by

$$\mathbf{X}_i = \begin{bmatrix} \mathbf{x}_i^T & & & \\ & \mathbf{x}_i^T & & \\ & & \ddots & \\ & & & \mathbf{x}_i^T \end{bmatrix}.$$

Since the parameters $\boldsymbol{\beta}^*$ is a reparameterization of the parameters $\boldsymbol{\beta}$, the computation of maximum likelihood estimates of $\boldsymbol{\beta}^*$ needs a transformation of the design matrix $\mathbf{X}$ as $\mathbf{X}^* = \mathbf{X}\mathbf{T}^*$, for a $q(K \times K)$ matrix $\mathbf{T}^*$ given by $\mathbf{T}^* = \mathbf{T}^{-1}_{q \times q} \otimes \mathbf{I}_{K \times K}$, where $\otimes$ is the Kronecker matrix product. If not mentioned otherwise we will use the symmetric side constraint.

2.2. Regularization and Type of Predictors

In the version of componentwise boosting used in Section 3, the effect of one predictor variable, that is all the parameters linked to that variable, will be updated within one step of the algorithm. Updating of the predictor will be performed by regularized estimates with the regularization depending on the type of predictor.

Nominal Predictors

Let the predictor X take values $1, \ldots, K, K+1$ and X be the only variable in the predictor. The parameter values for response category r have length K and are given by $\boldsymbol{\beta}^{*T}_r = (\beta^*_{r1}, \ldots, \beta^*_{rK})$. Regularization will be based on ridge type estimates. For the multinomial logit model it is advisable to use the symmetric side constraint, otherwise shrinkage is determined by the choice of the reference category (see Zahid and Tutz (2009)). The corresponding ridge estimators can be motivated by maximization of the penalized log-likelihood

$$l_p(\boldsymbol{\beta}) = \sum_{i=1}^{n} l_i(\boldsymbol{\beta}) - \frac{\lambda}{2} \sum_{r=1}^{k} \sum_{j=1}^{K} \beta^{*2}_{rj},$$

where $l_i(\boldsymbol{\beta})$ is the log-likelihood contribution of the ith observation, λ is a tuning parameter and $\sum_{r=1}^{k} \beta^*_{rj} = 0$. The underlying penalty can also be given by

$$J(\boldsymbol{\beta}^*) = \sum_{r=1}^{k} \sum_{j=1}^{K} \beta^{*2}_{rj} = \sum_{j=1}^{K} \boldsymbol{\beta}^{*T}_{.j} \mathbf{T}^{-1} \boldsymbol{\beta}^*_{.j}, \tag{6}$$

with shortened vector $\boldsymbol{\beta}^{*T}_{.j} = (\beta^*_{1j}, \ldots, \beta^*_{qj})$. It should be noted that the use of matrix $\mathbf{T}^{-1}$ in place of an identity matrix $\mathbf{I}$, implicitly penalizes the size of parameters for all k response categories while working with the $q = k - 1$ logits. In matrix notation, one obtains the ridge penalty with symmetric side constraint, for a complete design matrix for the multinomial logit model in the form

$$J(\boldsymbol{\beta}^*) = \boldsymbol{\beta}^{*T}\mathbf{T}^*\boldsymbol{\beta}^*,$$

where $\boldsymbol{\beta}^*$ is a $qK \times 1$ vector given by $\boldsymbol{\beta}^{*T} = (\boldsymbol{\beta}^{*T}_1, \ldots, \boldsymbol{\beta}^{*T}_q)$ and the matrix $\mathbf{T}^* = \mathbf{T}^{-1}_{q\times q} \otimes \mathbf{I}_{K\times K}$, is same as discussed in Section 2.1.

Ordinal Predictors

In regression analysis, ordinal predictors are often part of the predictor space but proper treatment is found rarely. If the multinomial logit model has some ordinal predictors, penalization should account for the order of the categories. With ordinal predictors, it is advantageous to penalize the differences between the coefficients of adjacent categories rather than penalizing the size of the parameters themselves. By penalizing such differences, one gets a smoother coefficient vector and avoid the high jumps among the parameter estimates corresponding to the ordinal covariate (see Gertheiss and Tutz (2009)). Let again the ordinal predictor take $K + 1$ categories $1, \ldots, K, K + 1$ and let the first category serve as reference category such that $\beta_{.1} = 0$. Then for ordinal predictor with $K + 1$ categories an appropriate penalty with symmetric side constraint is

$$J(\boldsymbol{\beta}^*) = \sum_{r=1}^{k}\sum_{j=2}^{K+1}(\beta^*_{rj} - \beta^*_{r,j-1})^2. \tag{7}$$

If one works with a reference category constraint then the penalty is simply given as $J(\boldsymbol{\beta}) = \sum_{r=1}^{k}\sum_{j=2}^{K+1}(\beta_{rj} - \beta_{r,j-1})^2$, or in matrix notation $J(\boldsymbol{\beta}) = \sum_{j=2}^{K+1}\boldsymbol{\beta}^T_{.j}\boldsymbol{\Omega}\boldsymbol{\beta}_{.j}$, where $\boldsymbol{\Omega} = \mathbf{U}^T\mathbf{U}$ with $\mathbf{U}$, a $K \times K$ matrix, given by

$$\mathbf{U} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ -1 & 1 & \ddots & & \vdots \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix}$$

But we are proceeding with symmetric side constraint where we have to penalize the differences between the parameters of adjacent categories for all k categories of the response variable, while working with q logits. In such case the penalty term for the complete design matrix is given as

$$J(\boldsymbol{\beta}^*) = \boldsymbol{\beta}^{*T}\boldsymbol{\Omega}^*\boldsymbol{\beta}^*, \tag{8}$$

where $\boldsymbol{\beta}^*$ is a $qK \times 1$ vector given by $\boldsymbol{\beta}^{*T} = (\boldsymbol{\beta}_1^{*T}, \ldots, \boldsymbol{\beta}_q^{*T})$ and the matrix $\boldsymbol{\Omega}^* = \mathbf{T}_{q\times q}^{-1} \otimes \boldsymbol{\Omega}_{K\times K}$. The matrix $\boldsymbol{\Omega}^*$ will handle implicitly the penalization of differences between the adjacent categories for k categories while working with the q logits.

In the next section where we are using the ridge type penalties to obtain weak learners in the boosting algorithm. Two types of penalty matrices are used for the expression $J(\boldsymbol{\beta}^*)$. If the candidate predictor is ordinal, the penalty matrix $\boldsymbol{\Omega}^*$ is used to penalize the differences among the coefficients of adjacent predictor categories for k response categories and if the candidate predictor is nominal, the penalty matrix $\mathbf{T}^*$ is used to penalize the size of the parameters. Although we are working with q logits both penalty matrices implicity perform the penalization for the k logits under symmetric side constraint.

3. Boosting

The method proposed in this section is based on likelihood-based boosting with quadratic penalties for regularization to obtain weak learners. Let us consider the multinomial logit model with side constraint given by (3) and the penalty term given by (6) or (8) according to the nature of predictors. Then the penalized log-likelihood is

$$l_p(\boldsymbol{\beta}^*) = \sum_{i=1}^{n} l_i(\boldsymbol{\beta}^*) - \frac{\lambda}{2} J(\boldsymbol{\beta}^*) = \sum_{i=1}^{n} l_i(\boldsymbol{\beta}^*) - \frac{\lambda}{2} \boldsymbol{\beta}^{*T} \mathbf{P}^* \boldsymbol{\beta}^*$$

with penalty matrix $\mathbf{P}^*$ which will be replaced by $\mathbf{T}^*$ or $\boldsymbol{\Omega}^*$ depending on the nature of the predictors. The corresponding penalized score function $s_p(\boldsymbol{\beta}^*)$ is given by

$$\begin{aligned} s_p(\boldsymbol{\beta}^*) &= \sum_{i=1}^{n} \mathbf{X}_i^{*T} \mathbf{D}_i(\boldsymbol{\beta}^*) \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{\beta}^*)[\mathbf{y}_i - h(\boldsymbol{\eta}_i^*)] - \lambda \mathbf{P}^* \boldsymbol{\beta}^* \\ &= \mathbf{X}^{*T} \mathbf{D}(\boldsymbol{\beta}^*) \boldsymbol{\Sigma}^{-1}(\boldsymbol{\beta}^*)[\mathbf{y} - h(\boldsymbol{\eta}^*)] - \lambda \mathbf{P}^* \boldsymbol{\beta}^*, \end{aligned}$$

where $\boldsymbol{\beta}^*$ is a vector of parameters of length $q \times (p+1)$ and $\mathbf{X}^*$ is the transformation of the actual design matrix as discussed in Section 2.2. The matrix $\mathbf{D}_i(\boldsymbol{\beta}^*) = \frac{\partial h(\boldsymbol{\eta}_i^*)}{\partial \boldsymbol{\eta}^*}$ is the derivative of $h(\boldsymbol{\eta}^*)$ evaluated at $\boldsymbol{\eta}_i^* = \mathbf{X}_i^* \boldsymbol{\beta}^*$ and $\boldsymbol{\Sigma}(\boldsymbol{\beta}^*) = \text{cov}(\mathbf{y}_i)$ is the covariance matrix of ith observation of $\mathbf{y}$ given parameter vector $\boldsymbol{\beta}^*$. For the full design matrix, in matrix notation $\mathbf{y}$ and $h(\boldsymbol{\eta}^*)$ are given by $\mathbf{y}^T = (\mathbf{y}_1^T, \ldots, \mathbf{y}_n^T)$ and $h(\boldsymbol{\eta}^*)^T = (h(\boldsymbol{\eta}_1^*)^T, \ldots, h(\boldsymbol{\eta}_n^*)^T)$ respectively. The matrices have block diagonal form $\boldsymbol{\Sigma}(\boldsymbol{\beta}^*) = \text{diag}(\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{\beta}^*))$, $\mathbf{W}(\boldsymbol{\beta}^*) = \text{diag}(\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{\beta}^*))$, and $\mathbf{D}(\boldsymbol{\beta}^*) = \text{diag}(\mathbf{D}_i(\boldsymbol{\beta}^*))$.

For the boosting algorithm, let the the whole predictor space be dichotomized into two non-overlapping groups. One group contains the mandatory/obligatory covariates (including the intercept) which are the essential part of the model and are necessarily re-estimated in each of m boosting iterations. The second group consists of the candidate predictors each of which is a candidate to become a part of the final parsimonious model decided after m boosting iterations. Let the predictor variable indices $V = \{1, \ldots, p\}$ be partitioned into disjoint sets as $V = V_o \cup V_1 \cup \ldots \cup V_g$, where V_o represents the obligatory predictors (each predictor may have one or more parameters associated with it) and $V_1, \ldots, V_g$ are g predictors, among which we want to make a subset selection. Let K_j denote the number of parameters/dummies for one logit associated with the predictor V_j, $(j = 1, \ldots, g)$. So the total predictor space is partitioned into two disjoint sets of obligatory and candidate predictors i.e., $V = V_o \cup V_c$, where $V_c = V_1 \cup \ldots \cup V_g$, and the split of complete parameter vector is then given as $\boldsymbol{\beta}^{*T} = (\boldsymbol{\beta}_o^{*T} \;\; \boldsymbol{\beta}_c^{*T})$. For the re-fitting process a combination of mandatory and some candidate predictor is considered i.e., $V_o \cup V_j,\ j \in \{1, \ldots, g\}$, will be considered in a re-fitting process in a particular boosting iteration. If among the candidate predictors, V_j is considered for re-fitting in a boosting iteration then for likelihood/penalized-likelihood estimation we use the $q(n \times K_j)$ design matrix $\mathbf{X}_j$ from the full design matrix of order $q(n \times (\sum_{j=1}^{p} K_j + 1))$. The design matrix $\mathbf{X}_j$ is based on the parameters/columns associated with V_j and is given as

$$\mathbf{X}_j^T = [\mathbf{X}_{1(j)} \;\; \mathbf{X}_{2(j)} \;\; \ldots \;\; \mathbf{X}_{n(j)}] \;\;\text{with}\;\; \mathbf{X}_{i(j)} = \begin{bmatrix} \mathbf{x}_{i(j)}^T & & & \\ & \mathbf{x}_{i(j)}^T & & \\ & & \ddots & \\ & & & \mathbf{x}_{i(j)}^T \end{bmatrix} \tag{9}$$

The *multinomBoost* algorithm can be described as follows:

Algorithm: *mulinomBoost*

Step 1: (Initialization)

Fit the intercept model $\boldsymbol{\mu}_0^* = h(\boldsymbol{\eta}_0^*)$ by maximizing the likelihood fucnction to obtain $\hat{\boldsymbol{\eta}}_0^*$ and $h(\hat{\boldsymbol{\eta}}_0^*)$.

Step 2: Boosting iterations

For $m = 1, 2, \ldots$

Step 2A: For obligatory/mandatory predictors

(i) Fit the model $\boldsymbol{\mu} = h(\hat{\boldsymbol{\eta}}_{m-1}^* + \mathbf{X}_o^* \boldsymbol{\beta}_o^{*F1})$, where $\hat{\boldsymbol{\eta}}_{m-1}^*$ is treated as an offset and $\mathbf{X}_o^*$ is the design matrix based on the parameters/columns corresponding to V_o. $\boldsymbol{\beta}_o^{*F1}$ is computed with one-step Fisher scoring as

$$\boldsymbol{\beta}_o^{*F1} = (\mathbf{X}_o^{*T} \mathbf{W}(\hat{\boldsymbol{\eta}}_{m-1}^*) \mathbf{X}_o^*)^{-1} \, \mathbf{X}_o^{*T} \; W(\hat{\boldsymbol{\eta}}_{m-1}^*) \, \mathbf{D}^{-1} \, (\mathbf{y} - \hat{\boldsymbol{\mu}}_{m-1}^*).$$

(ii) set $\hat{\boldsymbol{\eta}}_m^* = \hat{\boldsymbol{\eta}}_{m-1}^* + \mathbf{X}_o^* \boldsymbol{\beta}_o^{*F1}$.

(iii) set $\boldsymbol{\beta}_{o(m)}^* = \boldsymbol{\beta}_{o(m-1)}^* + \boldsymbol{\beta}_o^{*F1}$

Step 2B: For candidate predictors

(i) For $j = 1, \ldots, g$, fit the model $\boldsymbol{\mu} = h(\hat{\boldsymbol{\eta}}_m^* + \mathbf{X}_j^* \boldsymbol{\beta}_j^{*F1})$, with offset $\hat{\boldsymbol{\eta}}_m^*$ and $\mathbf{X}_j^*$ is the design matrix corresponding to V_j. With one-step Fisher scoring by maximizing penalized log-likelihood, $\boldsymbol{\beta}_j^{*F1}$ is computed as

$$\boldsymbol{\beta}_j^{*F1} = (\mathbf{X}_j^{*T} \mathbf{W}(\hat{\boldsymbol{\eta}}_m^*) \mathbf{X}_j^* + \nu \, \boldsymbol{\Lambda})^{-1} \, \mathbf{X}_j^{*T} \; W(\hat{\boldsymbol{\eta}}_m^*) \, \mathbf{D}^{-1} \, (\mathbf{y} - \hat{\boldsymbol{\mu}}_m^*).$$

where $\nu = \sqrt{df_j}\lambda$ with ridge penalty λ. The penalty matrix $\boldsymbol{\Lambda}$ is given as:

$$\boldsymbol{\Lambda} = \begin{cases} \boldsymbol{\Omega}^* & \text{if } V_j \text{ is ordinal} \\ \mathbf{T}^* & \text{otherwise.} \end{cases}$$

(ii) From the candidate predictors $V_1, \dots, V_g$, select the predictor say V_{best}, which improves the fit maximally and set

$$\boldsymbol{\beta}_c^{*F1} = \begin{cases} \boldsymbol{\beta}_j^{*F1} & \text{if } j \in V_{\text{best}} \\ 0 & \text{if } j \notin V_{\text{best}}. \end{cases}$$

(iii) set $\hat{\boldsymbol{\eta}}_m^* \leftarrow \hat{\boldsymbol{\eta}}_m^* + \mathbf{X}_c^* \boldsymbol{\beta}_c^{*F1}$.

(iv) set $\boldsymbol{\beta}_{c(m)}^* = \boldsymbol{\beta}_{c(m-1)}^* + \boldsymbol{\beta}_c^{*F1}$

In the above algorithm, for regularization, multiplying the ridge penalty λ with $\sqrt{df_j}$ accounts for the number of parameters, df_j, involved in the candidate predictor. The parameter λ is chosen large in order to obtain a weak learner. If it is chosen large enough, as usually in boosting, the performance does not depend on the choice of the value of λ, it only influences the needed number of iterations. In step 2B of the algorithm, the deviance can be used for selecting a candidate predictor for refit. In the mth boosting iteration, that candidate predictor will be considered for the refit which has minimum deviance i.e., $\text{Dev}(\hat{\boldsymbol{\eta}}_m^*)$. As the candidate predictors may have a varying number of parameters, an alternative criterion for predictor selection is Akaike's information criterion (AIC) or Bayesian information criterion (BIC) because both of these measures also take the the number of parameters into account. The AIC criterion is given by

$$\text{AIC} = \text{Dev}(\hat{\boldsymbol{\eta}}_m^*) + 2\, df_m$$

whereas the BIC criterion is given by

$$\text{BIC} = \text{Dev}(\hat{\boldsymbol{\eta}}_m^*) + \log(qn)\, df_m$$

where df_m is the effective degrees of freedom given by the trace of the hat matrix. But if the deviance as a criterion for the selection of a covariate makes the fitting procedure much faster, especially for large samples that is an advantage. The stopping criteria can be based on deviance based cross-validation, which we are using. Alternative but more time-consuming options are AIC or BIC.

One possible drawback of boosting is that the parameters corresponding to some predictors may be updated only once or twice within the boosting iterations. It is recommended to select only those variables whose estimates are not too small compared to the other estimates. The *multinomBoost* algorithm sets all the parameter estimates corresponding to the ith predictor equal to zero, if

$$\frac{\frac{1}{k.K_i}\sum_{j=1}^{K_i}\sum_{l=1}^{k}|\beta_{ijl}|}{\sum_{i=1}^{p}\frac{1}{k.K_i}\sum_{j=1}^{K_i}\sum_{l=1}^{k}|\beta_{ijl}|} < \frac{1}{p}, \tag{10}$$

after m boosting iterations. For boosting, by using approximate hat matrix $\mathbf{H}_m$ at the end of mth boosting iteration, AIC and BIC criteria are given as AIC = $\text{Dev}(\hat{\boldsymbol{\eta}}_m^*) + 2\,\text{tr}(\mathbf{H}_m)$ and BIC = $\text{Dev}(\hat{\boldsymbol{\eta}}_m^*) + \log(qn)\,\text{tr}(\mathbf{H}_m)$ respectively. The approximate hat matrix used in the mth boosting iteration is discussed in the following proposition.

Proposition: In the mth boosting iteration, an approximate hat matrix for which $\hat{\boldsymbol{\mu}}^*_m \approx \mathbf{H}_m\mathbf{y}$ is given by

$$\mathbf{H}_m = \sum_{j=0}^{m} \mathbf{M}_j \prod_{i=0}^{j-1} (\mathbf{I} - \mathbf{M}_0),$$

where for the multinomial logit models with $\mathbf{W}_m = \mathbf{D}_m$ (for $\mathbf{W}_m = \mathbf{W}(\hat{\boldsymbol{\eta}}^*_m)$ and $\mathbf{D}_m = \mathbf{D}(\hat{\boldsymbol{\eta}}^*_m)$), $\mathbf{M}_m = \mathbf{W}_m(\mathbf{X}_m^T\mathbf{W}_m\mathbf{X}_m + \nu\boldsymbol{\Lambda})^{-1}\mathbf{X}_m$.

Proof: At the end of mth boosting iteration, let $V_j = V_{\text{best}}$ is selected. For multinomial logit models with $\mathbf{W}_m = \mathbf{D}_m$, where $\mathbf{W}_m = \mathbf{W}(\hat{\boldsymbol{\eta}}^*_m)$ and $\mathbf{D}_m = \mathbf{D}(\hat{\boldsymbol{\eta}}^*_m)$, we have $\hat{\boldsymbol{\eta}}_m - \hat{\boldsymbol{\eta}}_{m-1} = (\mathbf{X}_j^{*T}\mathbf{W}_m\mathbf{X}_j^* + \nu\,\boldsymbol{\Lambda})^{-1}\ \mathbf{X}_j^{*T}\ (\mathbf{y} - \hat{\boldsymbol{\mu}}^*_{m-1})$. By using the first order Taylor approximation of first order i.e., $h(\hat{\boldsymbol{\eta}}) \approx h(\boldsymbol{\eta}) + (\partial h(\boldsymbol{\eta})/\partial\boldsymbol{\eta}^T)(\hat{\boldsymbol{\eta}} - \boldsymbol{\eta})$, we obtain $\hat{\boldsymbol{\mu}}^*_m \approx \hat{\boldsymbol{\mu}}^*_{m-1} + \mathbf{W}_m(\hat{\boldsymbol{\eta}}^*_m - \hat{\boldsymbol{\eta}}^*_{m-1}) = \hat{\boldsymbol{\mu}}^*_{m-1} + \mathbf{W}_m\mathbf{X}^*_j\hat{\boldsymbol{\beta}}^{*F1} = \hat{\boldsymbol{\mu}}^*_{m-1} + \mathbf{W}_m\mathbf{X}^*_j(\mathbf{X}_j^{*T}\mathbf{W}_m\mathbf{X}_j^* + \nu\,\boldsymbol{\Lambda})^{-1}\ \mathbf{X}_j^{*T}\ (\mathbf{y} - \hat{\boldsymbol{\mu}}^*_{m-1})$. So we have $\hat{\boldsymbol{\mu}}^*_m \approx \hat{\boldsymbol{\mu}}^*_{m-1} + \mathbf{M}_m(\mathbf{y} - \hat{\boldsymbol{\mu}}^*_{m-1})$ with $\mathbf{M}_m = \mathbf{W}_m\mathbf{X}^*_j(\mathbf{X}_j^{*T}\mathbf{W}_m\mathbf{X}_j^* + \nu\,\boldsymbol{\Lambda})^{-1}\ \mathbf{X}_j^{*T}$. We can write $\hat{\boldsymbol{\mu}}^*_m$ as $\hat{\boldsymbol{\mu}}^*_m \approx \hat{\boldsymbol{\mu}}^*_{m-1} + \mathbf{M}_m(\mathbf{y} - \hat{\boldsymbol{\mu}}^*_{m-1}) = \mathbf{H}_{m-1}\mathbf{y} + \mathbf{M}_m(\mathbf{I} - \mathbf{H}_{m-1})\mathbf{y}$. Expanding in the same way, for mth boosting iteration, the general form of the approximate hat matrix is $\mathbf{H}_m = \sum_{j=0}^{m}\mathbf{M}_j\prod_{i=0}^{j-1}(\mathbf{I} - \mathbf{M}_0)$, with $\hat{\boldsymbol{\mu}}^*_m \approx \mathbf{H}_m\mathbf{y}$ and the starting value $\hat{\boldsymbol{\mu}}^*_0 = \mathbf{M}_0\mathbf{y}$.

4. Simulation Study

The performance of *multinomBoost* algorithm is evaluated using simulated data. For a response variable with three categories (unordered), the covariates are drawn from a $p-$dimensional multivariate normal distribution with mean $\mathbf{0}$ and the covariance among the covariates (among the columns of covariate matrix) $\mathbf{x}_j$ and $\mathbf{x}_k$ is $\rho^{|j-k|}$. Two values of ρ, 0.3 and 0.7 are considered in the study. For each value of ρ we draw a samples of sizes 50 and 100 for a design space of 20 covariates (16 continuous and four binary covariates). Among the 20 covariates, six covariates (five continuous and one binary covariate) are informative i.e., we have six covariates with non-zero parameters and the rest having zero value. For the true parameter values $\boldsymbol{\beta}$ the total $q.\sum_{j=1}^{p_{\text{info}}} K_j$ values (where p_{info} is the total number of informative covariates) are obtained by the formula $(-1)^j\exp(-2(j-1)/20)$ for $j = 1, \ldots, q\sum_{j=1}^{p_{\text{info}}} K_j$, and are randomly allotted to the parameters corresponding to the informative covariates. The true parameter vector, then, is $\boldsymbol{\beta}^T = c_{snr}(\boldsymbol{\beta}_1 \quad \boldsymbol{\beta}_2)$, where the constant c_{snr} is chosen so that the signal-to-noise ratio is 3.0. The performance of the *multinomBoost* with respect to the variable selection in high dimensional data sets, where the categorical covariates are also involved is also evaluated. For this purpose, we use four additional settings, for which the total number of predictors used with their type and the number of informative predictors (within the brackets) is as follows:

Type of predictor	Setting 5	Setting 6	Setting 7	Setting 8
Metric (with $\rho = 0.3$ & $\rho = 0.7$ for setting 1 & 2 respectively):	90 (5)	90 (5)	–	–
Binary:	10 (1)	10 (1)	10 (2)	20 (4)
Categorical (with three unordered categories):	–	–	2 (1)	8 (2)
Categorical (with four unordered categories):	–	–	2 (1)	8 (2)
Categorical (with three ordered categories):	–	–	2 (1)	8 (2)
Categorical (with four ordered categories):	–	–	2 (1)	8 (2)

For the first four settings we performed $S = 50$ simulations per setting and $S = 30$ simulations per setting for the last four high dimensional settings. For the subset selection in boosting we use three criteria, deviance, AIC and BIC. For each criterion, in each setting, a fixed value of the rigde penalty λ is used for all S simulations. We chose that value for which the optimal number of boosting iterations was between 50 and 200. The optimal number of iterations in each case are decided on the basis of 10−fold cross-validation. The results obtained with these three criteria are compared to the MLE(oracle), which refers to the usual ML estimate for the model that contains the informative covariates only. Therefore MLE(oracle) has the big advantage of using the variables that carry information. In addition we consider ridge estimation (ssc-Ridge) with symmetric side constraint obtained for the model that contains all covariates. The performance of *multinomBoost* algorithm is evaluated in terms of mean squared error of the parameter estimates $\hat{\boldsymbol{\beta}}$, mean deviance of the fit, i.e. deviance($\hat{\boldsymbol{\pi}}$), prediction error and the identification of influential observations. The MSE of $\hat{\boldsymbol{\beta}}$ is computed using the estimates of all k logits as:

$$\text{MSE}(\hat{\boldsymbol{\beta}}) = \frac{1}{S} \sum_{s} \|\hat{\boldsymbol{\beta}}_s - \boldsymbol{\beta}\|^2.$$

Prediction performance is evaluated by drawing a new sample (test data) of 1000 observations. In generalized linear models, the mean deviance is an appropriate measure than mean squared prediction error. For the test data set the Mean Prediction Error (MPE) based on the deviance measure is given as:

$$\text{MPE} = \frac{1}{S} \sum_{s} D_s = \frac{1}{S} \sum_{s} \Big[\sum_{i=1}^{n} \sum_{j=1}^{k} \pi_{ij}^{test} \log\Big(\frac{\pi_{ij}^{test}}{\hat{\pi}_{ij}^{test}}\Big) \Big]$$

The deviance for the fit is computed as $\sum_{i=1}^{n} \sum_{j=1}^{k} y_{ij} \log\Big(\frac{y_{ij}}{\hat{\pi}_{ij}}\Big)$ with $y_{ij} \log\Big(\frac{y_{ij}}{\hat{\pi}_{ij}}\Big) = 0$ for $y_{ij} = 0$.

4.1. MSE and Prediction Performance

Figure 1 shows the results for the low dimensional settings in terms of MSE. Figure 2 shows the corresponding results for the high-dimensional settings where the model involves categorical covariates also. For setting 8, which involves only categorical covariates, the results of MLE(oracle) are not given because of its non-existence. The solid circles within the boxes represent the mean over observations for which the box-plots were drawn. It is seen that boosting strongly outperforms ridge (exception setting 8 for MSE). More surprisingly mean and median values are smaller than the corresponding values of the oracle. The oracle shows much larger variation, for some data sets its performance is very bad for some rather good. But given that it uses information that is not available in practice the performance is weak.

In setting 8 which contains only categorical covariates, the results of boosting regarding MSE($\hat{\boldsymbol{\beta}}$) and the fit are disturbed but still give better prediction performance. It should be noted that the results for boosting procedure in this case (with all covariates categorical) could be improved by using smaller value of signal-to-noise ratio. Table 1 shows the mean of log(MSE($\hat{\boldsymbol{\beta}}$)), mean deviance of the fit i.e., deviance($\hat{\pi}$) and the mean prediction error (MPE). The values appearing in boldface indicate the best results among all considered methods. In summary, Table 1 shows that boosting is a much better technique than its competitors not only when there is small correlation but also with high correlation among the covariates. We are not comparing the results of *multinomBoost* with estimates of Lasso or elastic net such as given by Friedman et al. (2010), because our algorithm is working in a different way and performs predictor selection (selecting a group of parameters at a time associated with a predictor) rather than parameter selection.

4.2. Identification of Informative Predictors

In addition to small prediction error (MPE) a selection procedure should yield a parsimonious model that includes all informative covariates. To identify whether the correct informative covariates are part of the final parsimonious model or not, we are using ”hit rate” and ”false alarm rate” as criteria. The hit rate is defined as the proportion of correctly identified informative predictors, given as

$$\text{hit rate} = \frac{\sum_{j=1}^{p} I(\boldsymbol{\beta}_{true,j} \neq \mathbf{0}).I(\hat{\boldsymbol{\beta}}_j \neq \mathbf{0})}{\sum_{j=1}^{p} I(\boldsymbol{\beta}_{true,j} \neq \mathbf{0})}.$$

The false alarm rate is defined as the proportion of non-informative predictors dubbed as informative, and is given as

$$\text{false alarm rate} = \frac{\sum_{j=1}^{p} I(\boldsymbol{\beta}_{true,j} = \mathbf{0}).I(\hat{\boldsymbol{\beta}}_j \neq \mathbf{0})}{\sum_{j=1}^{p} I(\boldsymbol{\beta}_{true,j} = \mathbf{0})}.$$

Here $\boldsymbol{\beta}_{true,j},\ j = 1,\dots,p$ is a vector that comprises the true parameter values for the k logits associated with the jth predictor and $\hat{\boldsymbol{\beta}}_j$ are the corresponding estimates. The indicator function I(expression) assumes the value 1, if ”expression” is true and 0 otherwise. To evaluate the performance of *multinomBoost* algorithm concerning selection hit rates and false alarm rates are given in Table 2 for all settings considered in the simulation study. It is seen that the procedure performs very well. Even is setting 8, which contains only categorical variables, hit rate is high and false alarm rate low.

The last four columns of Table 2 give a relative comparison of each method with MLE(oracle). For comparing the relative efficiency in terms of MPE we computed $\frac{1}{S}\sum_s(\text{MPE}_s^{\text{method}}/\text{MPE}_s^{\text{ML}})$, where $\text{MPE}_s^{\text{method}}$ represents the MPE for a boosting approach or ridge regression and MPE_s^{ML} is the MPE of MLE(oracle) for the sth sample. For setting 8 with all categorical predictors, these values are missing because MLE(oracle) estimates do not exist for any sample. The values with boldface represent the best result among the competitors in a particular setting. The deviance as a variable selection criterion is performing best for selecting the relevant covariates in almost all settings even when we have a small sample size relative to the number of covariates and with high correlations. AIC is a strong competitor of deviance than BIC and giving almost the same level of accuracy regarding the selection of the relevant predictors. In high dimensional setting when the model contains only binary or categorical predictors, hit rate with BIC as selection criteria is not so much appealing because it is ignoring most of the relevant predictors. But as for as the inclusion of non-relevant predictors is concerned, BIC is giving the best ”false alarm rate” in all situations. From the result of Table 2, AIC seems to be a good choice as criterion for variable selection when the both measures i.e., hit rate and false alarm rate for identification of the influential predictors are taken into account.

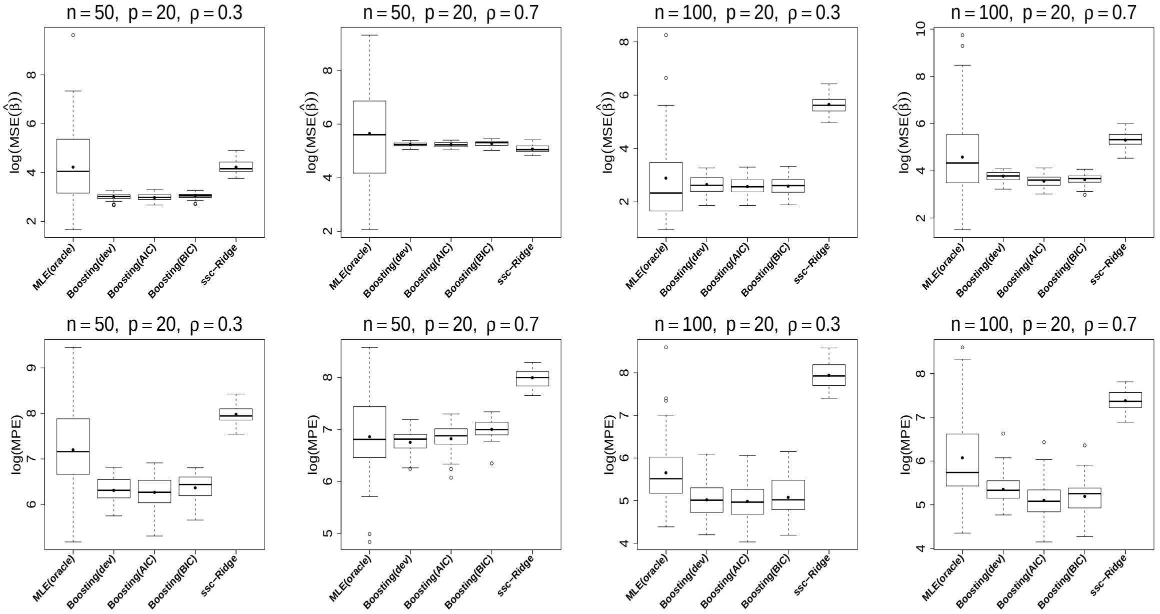

FIGURE 1: *Illustration of the simulation study for first four settings without categorical covariates: Box plots for comparing Boosting (using the criteria deviance, AIC and BIC) with MLE(oracle) and ssc-Ridge in terms of log(MSE($\hat{\boldsymbol{\beta}}$)) (top panel) and in terms of Mean Prediction Error i.e., log(MPE) (bottom panel). The solid circles within the boxes represent the mean of the data for which box plot is drawn.*

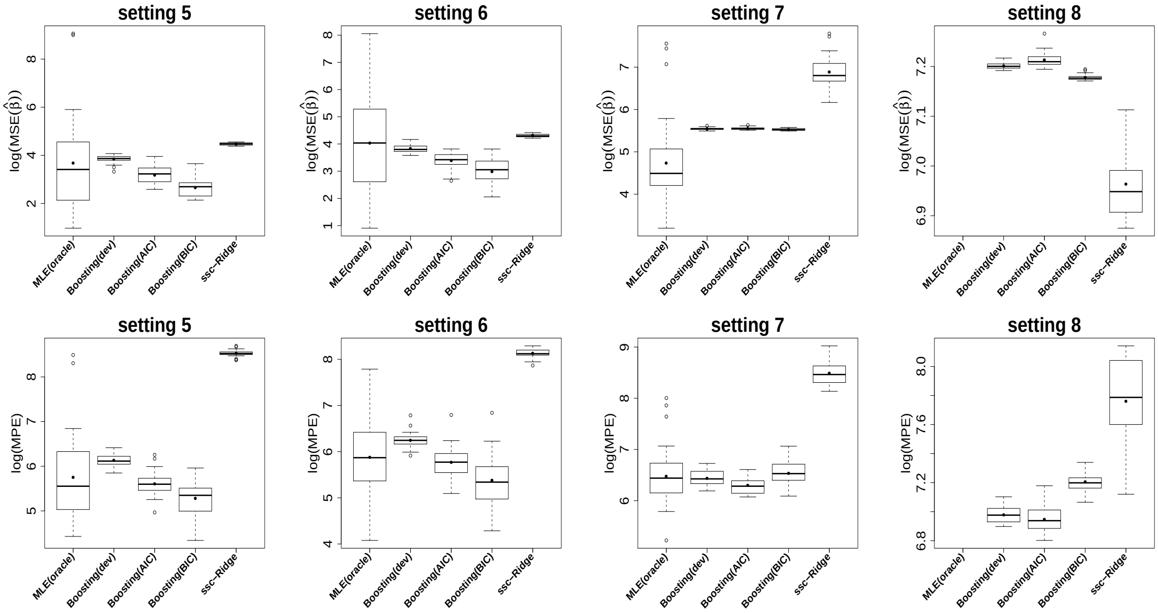

Figure 2: *Illustration of the simulation study for high dimensional settings: Box plots for comparing Boosting (using the criteria deviance, AIC and BIC) with MLE(oracle) and ssc-Ridge in terms of log(MSE($\hat{\boldsymbol{\beta}}$)) (top panel) and in terms of Mean Prediction Error i.e., log(MPE) (bottom panel). The solid circles within the boxes represent the mean of the data for which box plot is drawn.*

Table 1: *Comparison of Boosting approach with MLE(oracle) and ridge regression with symmetric side constraint (ssc-Ridge) in terms of log(MSE($\hat{\boldsymbol{\beta}}$)), deviance of the fit i.e., deviance($\hat{\boldsymbol{\pi}}$) and Mean Prediction Error (MPE). The values with boldface represent the best result with a particular method among all competitors.*

	log(MSE($\hat{\boldsymbol{\beta}}$))					deviance($\hat{\boldsymbol{\pi}}$)					MPE				
		Boosting					Boosting					Boosting			
	MLE	deviance	AIC	BIC	ssc-Ridge	MLE	deviance	AIC	BIC	ssc-Ridge	MLE	deviance	AIC	BIC	ssc-Ridge
Setting 1	4.2276	4.2186	3.0089	**2.9797**	3.0325	28.9138	12.0440	**11.7320**	13.3471	39.1327	1022.3555	286.2253	**278.3627**	302.9525	1485.1743
Setting 2	5.6426	5.2404	5.2310	5.2795	**5.0777**	23.2785	**12.5991**	12.8019	16.0280	35.9622	612.9043	**439.2145**	472.5820	554.9007	1492.8805
Setting 3	2.8980	2.6479	**2.5798**	2.5915	5.6423	26.0146	**15.7188**	16.0922	18.0862	103.0675	229.9102	83.3936	**80.6702**	89.6679	1466.4339
Setting 4	4.5806	3.7675	**3.5734**	3.6326	5.3012	36.4438	**25.1230**	26.4086	28.9922	49.6566	378.0500	112.3644	**90.5228**	98.3068	821.2240
Setting 5	3.6859	3.8458	3.1852	**2.6561**	4.4734	33.1327	17.2544	13.3362	**11.0415**	119.1908	303.9448	231.7965	141.2749	**104.9711**	2544.0830
Setting 6	4.0363	3.8262	3.3816	**2.9946**	4.3078	53.9725	39.5517	30.7109	**29.5848**	184.2309	527.4985	525.9714	340.9851	**258.2625**	3433.4700
Setting 7	**4.7309**	5.5441	5.5521	5.5289	6.8844	159.6963	152.8985	154.5399	**130.9688**	552.0475	398.8496	314.6842	**275.1543**	352.9394	2495.1473
Setting 8	–	7.2015	7.2136	**7.1781**	6.9634	–	216.9942	235.3204	244.9447	**96.7519**	–	1073.2970	**1043.7080**	1348.9930	2462.4450

Table 2: *Hit rates (HR) and false alarm rates (FAR) for identifying the informative predictors when deviance, AIC and BIC are used as criteria for selecting a predictor in a boosting iteration. Deviance is used as stopping criterion with* 10−*fold cross-validation. Last four columns represent the relative efficiency (R.E.) of different approaches with respect to MLE(oracle) in terms of Mean Prediction Error (MPE). A value less than one means that a particular approach is performing better than MLE(oracle). The boldface figures represent the best result among all competitor methods.*

	Deviance		AIC		BIC		Relative Efficiency			
	HR	FAR	HR	FAR	HR	FAR	RE(ridge)	RE(dev)	RE(AIC)	RE(BIC)
Setting 1	**0.7733**	0.0943	0.7267	0.0486	0.6067	**0.0100**	3.6742	0.6474	**0.6191**	0.6750
Setting 2	**0.6840**	0.1339	0.5694	0.1205	0.4583	**0.0892**	4.2538	**1.2758**	1.3823	1.5941
Setting 3	**0.9900**	0.0171	**0.9900**	0.0143	0.9633	**0.0000**	14.8751	0.7153	**0.6936**	0.7463
Setting 4	0.9700	0.0386	**0.9733**	0.0171	0.9267	**0.0057**	5.9219	0.7030	**0.5470**	0.5936
Setting 5	**0.9111**	0.1486	0.8722	0.0482	0.8111	**0.0096**	23.1847	2.1491	1.3049	**0.9094**
Setting 6	**0.9944**	0.1333	**0.9944**	0.0447	0.9611	**0.0085**	13.9645	2.1677	1.3102	**0.9011**
Setting 7	**0.9611**	0.1111	0.9389	0.0611	0.8278	**0.0167**	9.3696	1.1523	**1.0146**	1.3058
Setting 8	**0.8083**	0.1333	0.7444	0.0717	0.2944	**0.0492**	–	–	–	–

5. Application

In this section two data sets are used to illustrate variable selection by use of the *multinomBoost* algorithm and to show how it differs from the Lasso approach which focuses on parameter selection rather than variable selection. Both of the data sets are taken from the UCI machine learning repository.

5.1. Glass Identification Data

The first data set concerns the identification of type of glass (Blake et al. (1998)). The data comprises 214 observations. The response variable is the type of glass with six response categories given as: BFP (building windows float processed), BFNP (building windows non float processed), VFP (vehicle windows float processed), Con (containers), TW (tableware) and HL (headlamps). The nine continuously valued covariates are RI (refractive index), Na (Sodium), Mg (Magnesium), Al (Aluminum), Si (Silicon), K (potassium), Ca (calcium), Ba (Barium) and Fe (iron). The unit of measurement for all covariates other than refractive index is the weight percent in corresponding oxide.

For the identification of influential covariates, the *multinomBoost* algorithm is used. Three measures i.e., deviance, AIC and BIC are used to select a variable for updating in a boosting iteration. With all these three criteria, while using 10−fold cross-validation for deciding the optimal number of boosting iterations, only three covariates i.e., Na, Mg and Al are identified as potential covariates whereas the rest of the covariates are found non-informative with zero parameter estimates for all six response categories. The same data set is also analyzed by using the lasso approach. For this purpose the *glmnet* package of R (Friedman et al. (2010)) is used. The optimal value of L1-penalty decided on

Table 3: *Parameter estimates for six categories of "Type of Glass" with Lasso approach and the multinomBoost. For the boosting estimates AIC is used for variable selection and deviance is used as stopping criterion based on* 10−*fold cross-validation.*

	Lasso						Boosting					
Predictor	BFP	BNFP	VFP	Con	TW	HL	BFP	BNFP	VFP	Con	TW	HL
RI	0	0	−299.4145	0	0	0	0	0	0	0	0	0
Na	0	−0.0216	0	−0.0798	1.9375	1.3650	−0.5331	−0.6059	0.1177	−0.7570	1.0011	0.7773
Mg	1.0135	0.0700	0.0949	−0.4171	−0.0700	−0.6507	1.4192	0.2458	0.8044	−0.9832	−0.5732	−0.9130
Al	−3.0152	0	−2.6973	2.7338	0	1.2600	−1.4020	0.0541	−0.7351	1.2081	−0.0279	0.9028
Si	0	−0.4491	−1.4466	0	0.2272	0.4268	0	0	0	0	0	0
K	0	0	0	0.3329	−2.0816	0	0	0	0	0	0	0
Ca	0	0	0	0.3050	0	−0.1285	0	0	0	0	0	0
Ba	0	0	0	0	−0.1092	1.9513	0	0	0	0	0	0
Fe	0	2.5488	0	0	−0.8011	0	0	0	0	0	0	0

the basis of 10−fold cross validation was 0.009993293. The parameter estimates for lasso approach using this penalty term along with those from the boosting approach with AIC as variable selection criterion and deviance based on 10−fold cross-validation as stopping criterion are given in Table 3. The results in Table 3 show that the lasso approach does parameter selection but not variable selection. With lasso, all predictors are found relevant because at least one estimate for some response category is non-zero for each predictor. The boosting approach suggests only three predictors as relevant and the rest of the predictors as non-informative. In contrast to lasso, the boosting approach is selecting (or ignoring) the whole block of category-specific parameter estimates for each predictor with non-zero (or zero) values of the parameter estimates for all response categories. The coefficients build-up for relevant predictors for each response category resulting from boosting are plotted in Figure 3. The names of all non-informative predictors are overlapped against zero value on the right side of each plot. Figure 3 is showing the coefficients build-up only for the informative covariates because the *multinomBoost* algorithm sets parameter estimates to zero for all those predictors which fulfill the criteria given in (10) at final boosting iteration.

5.2. Contraceptive Method Choice Data

The second data set considered provided by Lim et al. (2000) is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The sample, 1473 married women who were either not pregnant or did not know if they were at the time of interview. The problem is to analyse the current contraceptive method choice (no use, long-term methods, or short-term methods) of a woman based on 10 demographic and socio-economic characteristics as: Wife's age , Wife's education (wife.edu; 1 =low, 2, 3, 4 =high), Husband's education (husband.edu; 1 =low, 2, 3, 4 =high), Number of children ever born, Wife's religion (0 =Non-Islam, 1 =Islam), Wife's now working? (0 =Yes, 1 =No), Husband's occupation (categorical: $1, 2, 3, 4$), standard of living (sol.index; 1 =low, 2, 3, 4 =high) and media exposure (0 =Good, 1 =Not good).

For this data set, we used the deviance for variable selection among the candidate predictors. For large samples using

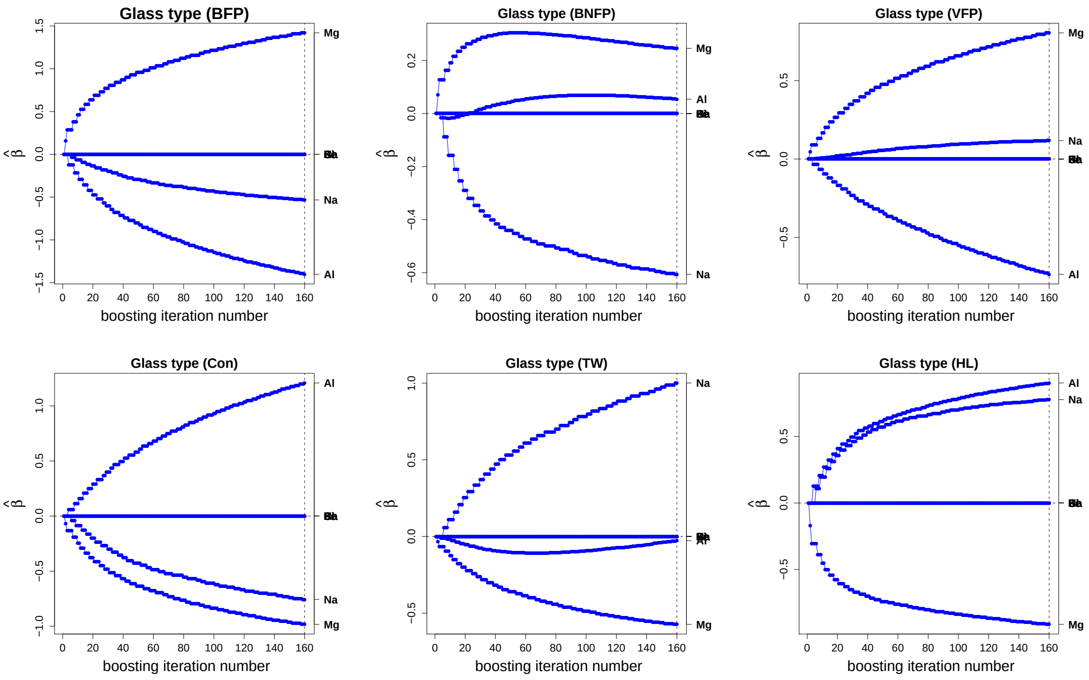

Figure 3: *Coefficient build-up in boosting for "Type of glass" data. The vertical dotted line represents the optimal boosting iteration number decided on the basis of* 10−*fold cross-validation when deviance is used as a stopping criterion. AIC is used for predictor selection. The names of all non-informative predictors are overlapped against zero value on the right side of each plot.*

AIC or BIC as variable selection criterion increases the computational burden and can slow the algorithm because of computation of hat matrix at each boosting iteration for each of the candidate predictors. The optimal number of boosting iterations was decided by deviance with 10−fold cross-validation. For computation of lasso estimates again the *glmnet* package (Friedman et al. (2010)) of R was used. The optimal value of penalty term decided with 10−fold cross-validation was 0.001016358. The lasso estimates with this penalty are given in Table 4 along with corresponding boosting estimates. As in Section 5.1, again the difference between parameter selection (by lasso approach) and variable selection (boosting approach) becomes obvious from Table 4. Here the process is one step more complex in the sense that categorical covariates are included. So with variable selection all parameters associated with dummies of the categorical covariate for all response categories should be selected or rejected at the same time. From the results of Table 4 it is clear that the boosting approach is following that rule but lasso is not. Once again with the lasso approach all the variables are found relevant when at least one predictor (or dummy associated with the categorical predictor) had non-zero estimate(s) for at least one response category. In contrast boosting is recommending just four informative predictors by selecting all of the parameters associated with the predictors for all response categories. The informative covariates include two continuous predictors i.e., wife's age and number of children ever born, and two categorical predictors (with all of their categories) i.e., wife's education and husband's education. The coefficients

Table 4: *Parameter estimates for three categories of the contraceptive methods used (i.e., No-use, Long-term and Short-term) with Lasso approach and the* multinomBoost. *For the boosting estimates deviance is used as stopping criterion with* 10−*fold cross-validation. Deviance is also used for predictor selection.*

	Lasso			Boosting		
Predictor	No-use	Long-term	Short-term	No-use	Long-term	Short-term
wife.age	0.0468	0	−0.0593	0.3683	0.0715	−0.4397
wife.edu2	−0.0371	0.6058	0	−0.1072	0.1303	−0.0231
wife.edu3	−0.3581	1.0610	0	−0.2872	0.3390	−0.0518
wife.edu4	−0.9607	1.3694	0	−0.5445	0.5760	−0.0315
husband.edu2	0	−0.7669	1.0697	−0.0397	−0.2146	0.2543
husband.edu3	0	−0.5997	1.2333	−0.0872	−0.2770	0.3642
husband.edu4	0	−0.5686	1.0176	−0.0764	−0.2628	0.3392
children	−0.3449	0	0	−0.4979	0.2407	0.2572
wife.religion	0.3522	−0.1795	0	0	0	0
wife.working	−0.0130	0	0.1527	0	0	0
husband.job2	0	−0.4072	0.0162	0	0	0
husband.job3	0	−0.2178	0.2897	0	0	0
husband.job4	−0.4590	0	0.0312	0	0	0
sol.index2	−0.3402	0.0219	0	0	0	0
sol.index3	−0.4528	0.2433	0	0	0	0
sol.index4	−0.6841	0.2440	0	0	0	0
media	0.5316	0	0	0	0	0

build-up for boosting is shown in Figure 4. The names of all non-informative predictors/dummies are overlapped against zero value on the right side of each plot. Although some of the predictors appearing in Table 4 or Figure 4 with zero parameter estimates, were considered for updating at some boosting iteration but at the final (optimal) boosting iteration *multinomBoost* algorithm set all those estimates to zero which fulfilled the criteria given in (10).

6. Concluding Remarks

In multinomial logit models, where the response variable has k un-ordered categories, there are $k-1$ parameters (or $(k-1)(K-1)$ parameters in the case of a categorical predictor with K categories) that are associated with one predictor variable. Subset selection means that the selection should not refer to the parameters but to the block/group associated with one predictor. When the predictor is categorical, the complexity goes one step farther and selection refer to the blocks/groups of all parameters associated with all dummies of the predictor for all response categories. Unfortunately existing techniques such as lasso do not work in that way but focus on individual parameters rather than predictors. One more issue with subset selection is that sometimes the experimenter wants to include particular variables in the model in such a way that they must be a part of the selected model. The lasso approach can discard

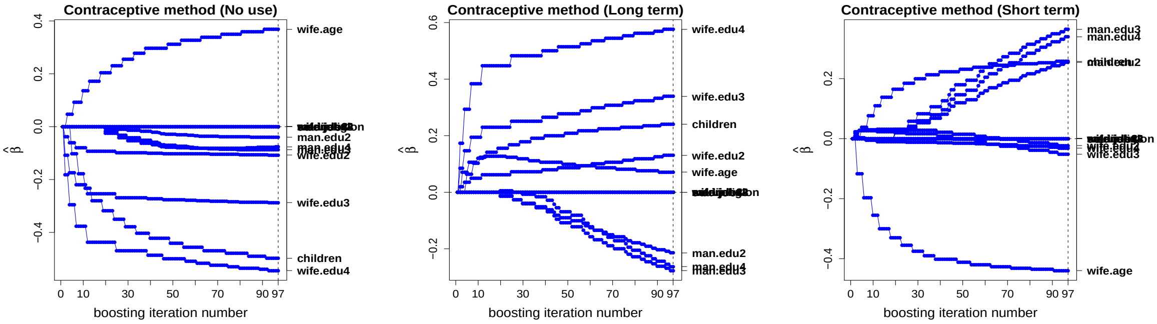

Figure 4: *Coefficient build-up in boosting for "contraceptive method choice" data. The vertical dotted line represents the optimal boosting iteration number decided on the basis of* $10-$*fold cross-validation when deviance is used as a stopping criterion as well as for predictor selection. The names of all non-informative predictors/dummies are overlapped against zero value on the right side of each plot.*

such variable during the subset selection process suggesting them as non-informative. The *multinomBoost* algorithm considered addresses both issues. It allows mandatory covariates to be a necessary part of the sparse model. Also, in contrast to lasso, predictors (not parameters) which form a group of parameters are considered candidates for updating in the next boosting iteration. As a result, in the final estimates either the complete block of parameters comprising a predictor is part of the sparse model or not. In addition our algorithm treats ordinal predictors properly. Instead of penalizing the parameters associated with dummies of the ordinal predictors, the difference between the parameters of adjacent categories is penalized. The effect is that also ordered predictors that contain many categories can be included in the model while simple maximum likelihood frequently fails when many categories are involved.

References

Blake, C., Keogh, E., Merz, C. J., 1998. UCI respository of machine learning databases.

Bühlmann, P., 2006. Boosting for high-dimensional linear models. The Annals of Statistics 34, 559–583.

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

Bühlmann, P., Yu, B., 2003. Boosting with l2 loss: Regression and classification. Journal of the American Statistical Association 98, 324–339.

Freund, Y., Schapire, R. E., 1996. Experiments with a new boosting algorithm. Machine Learning: Proc. of the Thirteenth International Conference, 148–156.

Friedman, J., Hastie, T., Tibshirani, R., 2010. Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software 33, 1–22.

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

Gertheiss, J., Tutz, G., 2009. Penalized regression with ordinal predictors. International Statistical Review 77, 345–365.

Krishnapuram, B., Carin, L., Figueiredo, M. A., Hartemink, A. J., 2005. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 957–968.

Lim, T.-S., Loh, W.-Y., Shih, Y.-S., 2000. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. Machine Learning 40, 203–229.

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

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

Schapire, R. E., 1990. The strength of weak learnability. Machine Learning 5, 197–227.

Segerstedt, B., 1992. On ordinary ridge regression in generalized linear models. Communications in Statistics: Theory and Methods 21, 2227–2246.

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

Tutz, G., Binder, H., 2006. Generalized additive modelling with implicit variable selection by likelihood based boosting. Biometrics 62, 961–971.

Tutz, G., Binder, H., 2007. Boosting ridge regression. Computational Statistics & Data Analysis 51, 6044–6059.

Zahid, F. M., Tutz, G., 2009. Ridge estimation for multinomial logit models with symmetric side constraints. Technical Report No. 67. Institute of Statistics, Ludwig-Maximilians-University Munich, Germany.