LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

INSTITUT FÜR STATISTIK

Gerhard Tutz, Wolfgang Pößnecker & Lorenz Uhlmann

Variable Selection in General Multinomial Logit Models

Technical Report Number 126, 2012
Department of Statistics
University of Munich

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

Variable Selection in General Multinomial Logit Models

Gerhard Tutz, Wolfgang Pößnecker & Lorenz Uhlmann

Ludwig-Maximilians-Universität München
Akademiestraße 1, 80799 München

October 2, 2012

Abstract

The use of the multinomial logit model is typically restricted to applications with few predictors, because in high-dimensional settings maximum likelihood estimates tend to deteriorate. In this paper we are proposing a sparsity-inducing penalty that accounts for the special structure of multinomial models. In contrast to existing methods, it penalizes the parameters that are linked to one variable in a grouped way and thus yields variable selection instead of parameter selection. We develop a proximal gradient method that is able to efficiently compute stable estimates. In addition, the penalization is extended to the important case of predictors that vary across response categories. We apply our estimator to the modeling of party choice of voters in Germany including voter-specific variables like age and gender but also party-specific features like stance on nuclear energy and immigration.

Keywords: Logistic regression, Multinomial logit model, Variable selection, Lasso, Group Lasso, CATS Lasso.

1. Introduction

The multinomial logit model is the most frequently used model in regression analysis for unordered multi-category responses. The maximum likelihood (ML) method, which is typically used for estimation, has the drawback that it requires more observations than parameters to be estimated. The amount of parameters, however, increases rapidly when the number of predictors grows, as multinomial logit models employ several coefficients for each explanatory variable. Therefore, ML estimates tend to deteriorate quickly and interpretability suffers as well, so that the number of predictors in the model is severely limited. For these reasons,

variable selection is necessary to obtain multinomial logit models that are both interpretable and reliable.

To illustrate these points, we consider data from the German Longitudinal Election Study (GLES) about party choice of voters during the 2009 parliamentary elections for the German Bundestag. Modeling the decision of voters for specific political parties and determining the major factors behind their preference are of great interest in political sciences. The available parties to choose from are the Christian Democratic Union (CDU), the Social Democratic Party (SPD), the Green Party (Bündnis 90/ Die Grünen), the Liberal Party (FDP) and the Left Party (Die Linke). As explanatory variables, various individual characteristics of the voter are considered, like, for example, gender, age or education, see Section 5 for a complete list. The main goal of our analysis is to select those predictors that influence party choice and to remove the rest. Besides improving interpretability, this is beneficial for polling firms and in opinion research. If, for example, gender was found to be irrelevant for party preference, one could save time and money while performing opinion polls as one would not have to care about a representative gender ratio among the interviewed persons.

While variable selection for such individual-specific predictiors requires new methodology due to the particular structure of multinomial logit models, this dataset offers another challenge in the form of predictors that are party-specific. For various topics like immigration or nuclear energy, participants of the study were asked how they perceive the parties' stance on this issue. Additionally, they stated their personal position on the topic. From this information, the distance between the personal point of view and the perceived position of the parties can be computed. These distances are then included into the model and are an example of so-called category-specific predictors. To be able to deal with the challenges of such a dataset, a method for variable selection in multinomial logit models is developed in this paper that accounts for the categorical and multivariate structure of these models and that works with both global and category-specific predictors.

The standard method for variable selection are forward/ backward strategies which have been used for a long time, but are notoriously unstable and thus cannot be recommended, see, for example, Hastie et al. (2009). An established alternative are penalty approaches for regularized variable selection. For linear and generalized linear models (GLMs), a variety of such methods has been proposed. The most prominent example is the Lasso (Tibshirani, 1996) and its extensions to Fused Lasso (Tibshirani et al., 2005) and Group Lasso (Yuan & Lin, 2006). Alternative regularized estimators that enforce variable selection are the Elastic Net (Zou & Hastie, 2005), SCAD (Fan & Li, 2001), the Dantzig selector (Candes & Tao, 2007) and boosting approaches (Bühlmann & Yu, 2003; Bühlmann & Hothorn, 2007; Tutz & Binder, 2006).

These methods, however, were developed for models with univariate response. Because the multinomial logit model is not a common univariate GLM, these methods cannot be applied directly. As mentioned previously, the effect of one predictor variable is represented by sev-

eral parameters. Therefore, one has to distinguish between variable selection and parameter selection, where variable selection is only achieved if all effects/ parameters that belong to one variable are simultaneously removed from the model. The available methods for multinomial logit models (Krishnapuram et al., 2005; Friedman et al., 2010) use L_1-type penalties that shrink all the parameters individually. Thus, they do not use the natural grouping of coefficients that is available, with each group containing the parameters that belong to the same explanatory variable. In particular, they pursue the goal of parameter selection and cannot directly promote variable selection.

In Section 2 we briefly introduce the multinomial logit model and propose a novel penalty that yields proper variable selection. Extensions to incorporate category-specific variables and categorical predictors both in the model and in the penalization are discussed separately. Regularized estimation is considered in Section 3, where a proximal gradient algorithm is derived that efficiently solves the corresponding estimation problem. The performance of our estimator is investigated in a simulation study in Section 4. Then, in Section 5, the real data example from the German Longitudinal Election Study is analysed using the developed methodology.

2. Model and Regularization

2.1. The Multinomial Logit Model with Category-Specific Covariates

For data $(y_i, \boldsymbol{x}_i), i = 1, \ldots, n$, with y_i denoting an observation of the categorical response variable $Y \in \{1, \ldots, k\}$ and $\boldsymbol{x}_i$ the p-dimensional vector of predictors, the multinomial logit model in its generic form specifies

$$\pi_{ir} = P(Y = r|\boldsymbol{x}_i) = \frac{\exp(\beta_{r0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_r)}{\sum_{s=1}^{k} \exp(\beta_{s0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_s)} = \frac{\exp(\eta_{ir})}{\sum_{s=1}^{k} \exp(\eta_{is})}, \tag{1}$$

where $\boldsymbol{\beta}_r^T = (\beta_{r1}, \ldots, \beta_{rp})$. Since parameters $\beta_{10}, \ldots, \beta_{k0}$, $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_k$ are not identifiable, additional constraints are needed. Typically, one of the response categories is chosen as reference category. We use category k as the reference category by setting $\beta_{k0} = 0$, $\boldsymbol{\beta}_k = \mathbf{0}$. With this choice, the linear predictors η_{ir}, $r = 1, \ldots, k-1$, correspond to the log odds between category r and the reference category k.

The model given in (1) is the most commonly used form of the multinomial logit model. However, it only uses "global" predictors that do not vary over response categories, a restriction that is not always appropriate in practice. In particular in the modeling of choice, when an individual chooses among alternatives $1, \ldots, k$, one wants to model the effects of characteristics of the individual like age and gender, but also account for measured attributes of the alternatives $1, \ldots, k$. In the modeling of preference for parties the alternatives are characterized by positions on policy dimensions. When the choice refers to transportation mode, the

potential attributes are price and duration, which vary across the alternatives and therefore are category-specific. Then, in addition to the global predictors $\boldsymbol{x}_i$, a set of category-specific predictors $\mathbf{w}_{i1}, \ldots, \mathbf{w}_{ik}$ is available, where $\mathbf{w}_{ir}$ contains the attributes of category r.

With k as the reference category, the set of linear predictors generalizes to

$$\eta_{ir} = \beta_{r0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_r + (\mathbf{w}_{ir} - \mathbf{w}_{ik})^T \boldsymbol{\alpha}, \qquad r = 1, \ldots, k-1. \tag{2}$$

The second term specifies the effect of the global variables, and the third term specifies the effect of the difference $\mathbf{w}_{ir} - \mathbf{w}_{ik}$ on the choice between category r and the reference category. For the interpretation of parameters it is often useful to consider the link between the latent utilities and the linear predictor. Let the latent utility of person i be specified by $u_{ir} = \gamma_{r0} + \boldsymbol{x}_i^T \boldsymbol{\gamma}_r + \mathbf{w}_{ir}^T \boldsymbol{\alpha}$ and the corresponding random utility be given as $U_{ir} = u_{ir} + \varepsilon_{ir}$, where ε_{ir} follows the Gumbel distribution, which has distribution function $F(\varepsilon) = \exp(-\exp(-\varepsilon))$. It is assumed that each individual chooses the alternative that yields maximum utility, that is, the link between the observable choice Y_i and the unobservable random utility is given by $Y_i = r \;\Leftrightarrow\; U_{ir} = \max_{j=1,\ldots,k} U_{jr}$. For the probabilities of the alternatives, one then obtains the multinomial logit model with predictor

$$\eta_{ir} = u_{ir} - u_{ik} = (\gamma_{r0} - \gamma_{k0}) + \boldsymbol{x}_i^T (\boldsymbol{\gamma}_r - \boldsymbol{\gamma}_k) + (\mathbf{w}_{ir} - \mathbf{w}_{ik})^T \boldsymbol{\alpha},$$

which has the form given in (2) with $\beta_{r0} = (\gamma_{r0} - \gamma_{k0})$, $\boldsymbol{\beta}_r = (\boldsymbol{\gamma}_r - \boldsymbol{\gamma}_k)$ (see Yellott, 1977; McFadden, 1973). An extensive discussion of the multinomial logit model as a multivariate GLM is given in Tutz (2012).

2.2. Regularization: the Categorically Structured Lasso

We first focus on regularization in multinomial logit models with only global predictors in which the overall parameter vector is given by $\boldsymbol{\beta}^T = (\beta_{10}, \ldots, \beta_{k-1,0}, \boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_{k-1}^T)$. A common way to regularize a model are penalty approaches in which one maximizes the penalized log-likelihood

$$l_p(\boldsymbol{\beta}) = l(\boldsymbol{\beta}) - \lambda J(\boldsymbol{\beta}),$$

where $l(\boldsymbol{\beta})$ denotes the usual log-likelihood, λ is a tuning parameter and $J(\boldsymbol{\beta})$ is a functional that typically penalizes the size of the parameters. While the tuning parameter determines the strength of the regularization, the functional determines the properties of the penalized estimator. Tibshirani (1996) introduced the Lasso that penalizes the L_1-norm of coefficients, that is, $J(\boldsymbol{\beta}) = ||\boldsymbol{\beta}||_1$, and showed that it facilitates sparse solutions in which coefficients of weak predictors are set to exactly zero. For the multinomial logit model without category-

specific predictors, direct application of the Lasso yields the L_1-norm of $\boldsymbol{\beta}$:

$$J(\boldsymbol{\beta}) = ||\boldsymbol{\beta}||_1 = \sum_{r=1}^{k-1} ||\boldsymbol{\beta}_r||_1 = \sum_{r=1}^{k-1} \sum_{j=1}^{p} |\beta_{rj}|. \tag{3}$$

Friedman et al. (2010) used the slightly more general Elastic Net penalty, which is an adaptation of the original Elastic Net (Zou & Hastie, 2005) to multinomial response models. In terms of variable selection, Elastic Net and Lasso both share the same drawback that selection focuses on parameters but not on variables. In univariate regression models, for which these penalties were developed, this distinction is irrelevant if only continuous or binary predictors are used because then each predictor influences the response through only one coefficient. By contrast, multinomial logit models use a whole vector $\boldsymbol{\beta}_{\bullet j}^T = (\beta_{1j}, \ldots, \beta_{k-1,j})$ of parameters to capture the influence of predictor x_j. The ordinary Lasso penalty from equation (3), however, only encourages selection of single parameters β_{rj}. Thus, variable selection is only achieved if $k-1$ coefficients are simultaneously shrunk to zero, but the Lasso does not enforce such behaviour.

Therefore, the ordinary Lasso is not ideal for variable selection in multinomial models. Although one might suspect that setting many coefficients to zero still improves interpretability, it is seen from equation (1) that the probability of all response categories is influenced by a predictor x_j if just one of the corresponding coefficients β_{rj}, $r = 1, \ldots, k-1$ is non-zero. Hence, there is a strong incentive in multinomial models to perform true variable selection by simultaneously removing all effects of a predictor from the model.

The alternative penalty that we propose penalizes the groups $\boldsymbol{\beta}_{\bullet j}$ of parameters that are linked to one variable. For simplicity, we assume all predictors to be metric, standardized and centered around zero. Extensions to categorical predictors follow in Section 2.3. If no category-specific predictors are included, we will use the penalty

$$J(\boldsymbol{\beta}) = \sum_{j=1}^{p} ||\boldsymbol{\beta}_{\bullet j}|| = \sum_{j=1}^{p} (\beta_{1j}^2 + \cdots + \beta_{k-1,j}^2)^{1/2}, \tag{4}$$

where $||\boldsymbol{u}|| = ||\boldsymbol{u}||_2 = \sqrt{\boldsymbol{u}^T\boldsymbol{u}}$ denotes the L_2-norm. The penalty enforces variable selection, that is, all the parameters in $\boldsymbol{\beta}_{\bullet j}$ are simultaneously shrunk toward zero. It is strongly related to the Group Lasso (Yuan & Lin, 2006; Meier et al., 2008). However, in the Group Lasso the grouping refers to the parameters that are linked to a categorical predictor within a univariate regression model whereas in the present model grouping arises from the multivariate response structure. In other words: the use of Group Lasso is *sometimes* necessary in regression models if the predictors have a specific structure. In multinomial logit models, on the other hand, it is the multivariate nature of the response that *always* leads to a model in which a grouped penalization across response categories is required for variable selection - irrespective of the structure of the predictors. In this sense, the penalty in (4) is the result of combining the idea of the original Lasso with the penalization across response categories that is mandatory

for variable selection in multi-category response models. Therefore, we call the corresponding penalized estimator *Categorically Structured Lasso (CATS Lasso)*.

It should be noted that variable selection in multinomial logit models could also be achieved by other penalties as long as they adequately account for the inherent structure of this model class. For example, the methodology that was developed for multivariate linear regression in Turlach et al. (2005) could be extended to multinomial logit models, even though the technical details would be cumbersome. Overall, each penalty term that yields direct variable selection in multinomial logit models must penalize the parameter vectors $\boldsymbol{\beta}_{\bullet j}$ in a groupwise fashion. We believe that CATS Lasso, given by (4), is the most simple and intuitive way to construct such a penalty.

In the previous discussion, we showed that CATS Lasso induces explicit variable selection while the ordinary Lasso does not, but there is another advantage of our approach: the sparsity induced by the Lasso depends on the choice of the reference category while that of CATS Lasso does not. As all coefficients β_{rj} in multinomial logit models are related to the log odds between category r and the reference category, shrinking one coefficient to zero actually means that x_j has the same effect on both category r and the reference category. If one changes the reference category, the coefficient β_{rj} may switch from zero to non-zero or vice versa. Therefore, interpretability and sparsity of Lasso solutions depend on the choice of the reference category, which often times is completely arbitrary. It is easy to see that the variable selection induced by (4) does not suffer from this drawback.

If category-specific predictors are present, the parameter vector $\boldsymbol{\alpha}$ has to be included in the penalty. For category-specific predictors there is one coefficient for each predictor, so that an ordinary Lasso penalty is appropriate for $\boldsymbol{\alpha}$. For a simple notation, let the overall parameter vector of the model be given by $\boldsymbol{\theta}^T = (\boldsymbol{\beta}_{\bullet 0}^T, \boldsymbol{\beta}_{\bullet 1}^T, \ldots, \boldsymbol{\beta}_{\bullet p}^T, \boldsymbol{\alpha}^T)$, where $\boldsymbol{\beta}_{\bullet 0}^T = (\beta_{10}, \ldots, \beta_{k-1,0})$ denotes the intercept vector. The parameter $\boldsymbol{\theta}$ has length $d = \big((p+1)(k-1) + L\big)$ in a model with an intercept, p global and L category-specific predictors. For this general case, we extend the CATS Lasso penalty to

$$J(\boldsymbol{\theta}) = \psi \sum_{j=1}^{p} \phi_j \, ||\boldsymbol{\beta}_{\bullet j}|| \; + \; (1-\psi) \sum_{l=1}^{L} \varphi_l \, |\alpha_l|, \tag{5}$$

where ψ is an additional tuning parameter that balances the penalty on the global and the category-specific variables. Unless stated otherwise, we always use $\psi = 0.5$. The parameters ϕ_j and φ_l are weights that assign different amounts of penalization to different parameter groups. Following the arguments of Yuan & Lin (2006), $\phi_j = \sqrt{k-1}$ and $\varphi_l = 1$ is used as a default, which guarantees that it is "fair" to use the same λ for both the parameter groups $\boldsymbol{\beta}_{\bullet j}$ and the single parameters α_l despite their different size.

2.3. Categorical Predictors

The general penalty (5), which enforces variable selection in multinomial logit models, was constructed under the restriction that all predictors are metric - more generally, it is suitable for all predictors that would enter a standard univariate GLM with one degree of freedom. This includes metric predictors and binary ones with dummy coding. For the more general case in which a predictor x_j enters a GLM with p_j parameters, extensions of our penalty are needed. The most prominent example for this situation are categorical predictors with p_j+1 categories which typically enter a GLM through p_j dummy variables.

For this case, Yuan & Lin (2006) argued that a grouped penalization and thus selection is more suitable than the selection of single coefficients performed by the ordinary Lasso. For this purpose, they proposed the Group Lasso, which was extended to GLMs in Meier et al. (2008). If such multi-category predictors are to be included in a multinomial logit model one has a vector of parameters for each response category and each predictor, that is, for response category r and variable j, one has $\boldsymbol{\beta}_{rj\bullet}^T = (\beta_{rj_1}, \ldots, \beta_{rj_{p_j}})$. Thus, there are two different kinds of grouping to consider: First, all coefficients that belong to one actual predictor should enter or leave the model jointly, that is, the whole vector $\boldsymbol{\beta}_{rj\bullet}$ should be set to zero if x_j is found to be irrelevant for category r. Second, the underlying principle of our categorically structured approach says that all the influences $\boldsymbol{\beta}_{rj\bullet}$, $r = 1, \ldots, k-1$ of x_j on the different response categories must be penalized in a grouped way to obtain variable selection. This means that the vector $\boldsymbol{\beta}_{\bullet j\bullet}^T = (\boldsymbol{\beta}_{1j\bullet}^T, \ldots, \boldsymbol{\beta}_{k-1,j\bullet}^T)$ of all coefficients that are linked to x_j should be treated as one big parameter group. Hence, for multi-category response models the concept of the Group Lasso for univariate models has to be combined with the concept of our Categorically Structured Lasso. The same idea applies to the vector of the category-specific variables $\boldsymbol{\alpha}$; if the effect of variable l is determined by the parameter vector $\boldsymbol{\alpha}_{l\bullet}^T = (\alpha_{l_1}, \ldots, \alpha_{l_{p_l}})$, the whole vector is penalized.

Hence, including categorical predictors yields CATS Lasso in its most general form:

$$J(\boldsymbol{\theta}) = \psi \sum_{j=1}^{p} \phi_j \, ||\boldsymbol{\beta}_{\bullet j\bullet}|| \; + \; (1-\psi) \sum_{l=1}^{L} \varphi_l \, ||\boldsymbol{\alpha}_{l\bullet}||, \tag{6}$$

where the weights ϕ_j and φ_l now default to $\phi_j = \sqrt{(k-1)p_j}$ and $\varphi_l = \sqrt{p_l}$. For notational simplicity, we consider metric or binary predictors for the rest of this paper and focus our exposition on the penalty from (5). However, software for the general case is available and has been used in the application.

2.4. Improved Variable Selection

Like Lasso, Group Lasso, Fused Lasso and other estimators with sparsity-inducing penalties, CATS Lasso is biased. In particular for large values of λ, the sparsity of solutions comes with

biased estimates of selected variables. Since a suitable value of λ is usually unknown, it is typically chosen by optimizing some appropriate criterion as, for example, cross-validation or AIC. In practice, however, one can frequently observe that a tuning parameter is chosen for which some weak predictors are not removed. In general, it is desirable to apply a high degree of penalization to weak predictors and mild penalization to strong ones. Two approaches to achieve this goal are discussed next.

Adaptive CATS Lasso

For the ordinary Lasso, Zou (2006) showed that the induced variable selection is inconsistent in certain scenarios and offered a remedy called adaptive Lasso. The same issue and a similar solution were discussed for the Group Lasso by Wang & Leng (2008). The CATS penalty from (5) will suffer from the same problem if simple weights $\phi_j = \sqrt{k-1}$ and $\varphi_l = 1$ are employed. Therefore, we use the idea of adaptive penalties to obtain adaptive CATS Lasso, in which the weights ϕ_j and φ_l are replaced by

$$\phi_j^a = \frac{\sqrt{k-1}}{||\hat{\boldsymbol{\beta}}_{\bullet j}^{\mathrm{ML}}||}, \quad \varphi_l^a = \frac{1}{|\hat{\alpha}_l{}^{\mathrm{ML}}|}, \tag{7}$$

where $\hat{\boldsymbol{\beta}}_{\bullet j}^{\mathrm{ML}}$ and $\hat{\alpha}_l{}^{\mathrm{ML}}$ denote the respective ML estimates. The basic idea of this adaptive penalization is that the norm of ML estimates of parameter groups belonging to irrelevant predictors asymptotically converges to zero, yielding a strong penalization, while relevant predictors are penalized less severely. In the simulation studies in Section 4, we demonstrate that these adaptive weights improve the quality of both variable selection and predictive performance of CATS Lasso. It should be noted that the ML estimates can be replaced by arbitrary $\sqrt{n}$-consistent estimates, for example an asymptotically vanishing ridge penalty. Such a ridge penalty is used in our implementation whenever ML estimates do not exist, for example in the $d > n$ case. More technical details are given in the appendix.

Refitting

Another concept to improve the variable selection of penalized estimators is to use them for selection purposes only and to perform an unpenalized refit on the set of active predictors, that is, those with non-zero coefficients. This refitting technique was mentioned, for example, in Efron et al. (2004) under the name "Lars-OLS hybrid" and in Candes & Tao (2007) as "Gauss-Dantzig selector". Our simulation studies show that the variable selection performance of CATS Lasso can be improved drastically by such an ML refit. This can be explained by the decoupling of bias and variable selection that is achieved via refitting: If the final estimator is obtained by an unpenalized refit, λ only steers variable selection, so that it can be chosen as large as necessary without having to worry about bias. If refitting is used, one

therefore typically observes that higher values of λ are chosen than for estimators without refit. The simulation study in Section 4 will investigate the improvement provided by refitting and adaptive weights and also compare them with each other.

3. Estimation

For the computation of the estimator proposed in the previous section, we consider maximization of the general penalized log-likelihood given by $l_p(\boldsymbol{\theta}) = l(\boldsymbol{\theta}) - \lambda J(\boldsymbol{\theta})$ with a d-dimensional parameter vector $\boldsymbol{\theta}$, a concave and continuously differentiable log-likelihood $l(\boldsymbol{\theta})$ and a convex penalty term $J(\boldsymbol{\theta})$. The penalized maximum likelihood estimator is defined by

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\operatorname{argmax}}\, l_p(\boldsymbol{\theta}) = \underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\operatorname{argmin}}\, \left(-l(\boldsymbol{\theta}) + \lambda J(\boldsymbol{\theta})\right). \tag{8}$$

CATS Lasso estimates are obtained as a special case if the log-likelihood $l(\boldsymbol{\theta})$ is that of multinomial logit models and the penalty term $J(\boldsymbol{\theta})$ has the form given in (5). In the following, we briefly show how this general maximization problem can be solved with the fast iterative shrinkage-thresholding algorithm (FISTA) of Beck & Teboulle (2009). Technically speaking, FISTA belongs to the class of proximal gradient methods in which only the log-likelihood and its gradient, but no higher-order derivatives are used.

This is particularly useful for multinomial models because they are inherently higher dimensional than univariate GLMs of comparable size. Consider a multinomial logit model with $p = 100$ and $k = 10$. Although this model is not excessively large compared to, for example, gene expression data where $p > 10000$ is quite common, the corresponding Fisher matrix would be of size 1000×1000. Inverting such a matrix or even storing it, as is required for the traditional Fisher scoring method, is very costly and outright impossible for large models. Although the Fisher matrix is avoided and the problem is non-smooth, FISTA achieves quadratic convergence, which is known to be optimal among black-box first-order methods for smooth convex optimization (Nemirovski, 1994; Nesterov, 2004). Thus, it combines quick convergence with cheap iterates that are well-suited for the specific challenges of multinomial logit models. In particular, it turned out to be much faster than the block coordinate descent algorithm of Meier et al. (2008) which we tested first, exploiting a complicated representation of multinomial logit models as multivariate GLMs.

For a clearer presentation, we first review the proximal gradient method in the context of the maximization of general penalized log-likelihoods. Then, we give an analytical and concise form of the building blocks that are required to compute CATS Lasso with it. Technical details of the algorithm and our implementation, which are not necessary for the underlying concept, are relegated to the appendix.

3.1. Proximal Gradient Methods for Penalized Log-likelihood Problems

With $\nu > 0$ denoting an inverse stepsize parameter, the iterations of proximal gradient methods are defined by (Beck & Teboulle, 2009)

$$\hat{\boldsymbol{\theta}}^{(t+1)} = \underset{\boldsymbol{\theta}\in\mathbb{R}^d}{\operatorname{argmin}} \left(-l(\hat{\boldsymbol{\theta}}^{(t)}) - \nabla l(\hat{\boldsymbol{\theta}}^{(t)})^T(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}^{(t)}) + \frac{\nu}{2}||\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}^{(t)}||^2 + \lambda J(\boldsymbol{\theta})\right), \tag{9}$$

which consists of a linear approximation of the negative log-likelihood at the current value $\hat{\boldsymbol{\theta}}^{(t)}$, a proximity term and the penalty. If one sets $\lambda = 0$, the scheme in (9) yields a sequence of unpenalized estimators, denoted here by $\tilde{\boldsymbol{\theta}}^{(t)}$, for which the explicit form

$$\tilde{\boldsymbol{\theta}}^{(t+1)} = \tilde{\boldsymbol{\theta}}^{(t)} + \frac{1}{\nu}\nabla l(\tilde{\boldsymbol{\theta}}^{(t)}) \tag{10}$$

is available. This is the standard formula for the iterates of gradient methods for smooth optimization and intuitively appealing because of the interpretation of the gradient as the direction of steepest ascent. Note that the sequence $\{\tilde{\boldsymbol{\theta}}^{(t)}\}$ converges to the ML estimator. Hence, the update in (10) can be considered a one-step approximation to the ML estimator based on the current iterate.

However, we require solutions to (9) with an active penalty. To obtain these solutions it is helpful to consider a more explicit and tractable formulation of (9) that can be derived using standard convex optimization theory: Via Lagrange duality (Bertsekas et al., 2003), equation (8) can equivalently be expressed by

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\in\mathcal{C}}{\operatorname{argmin}} \left(-l(\boldsymbol{\theta})\right), \tag{11}$$

where $\mathcal{C} = \{\boldsymbol{\theta} \in \mathbb{R}^d \,|\, J(\boldsymbol{\theta}) \leq \kappa(\lambda)\}$ is the constraint region corresponding to $J(\boldsymbol{\theta})$ and $\kappa(\lambda)$ is a tuning parameter that is linked to λ by a one-to-one mapping. Given a search point $\boldsymbol{u} \in \mathbb{R}^d$, the so-called proximal operator associated with $J(\boldsymbol{\theta})$ is defined by

$$\mathcal{P}_\lambda(\boldsymbol{u}) = \underset{\boldsymbol{\theta}\in\mathbb{R}^d}{\operatorname{argmin}} \left(\frac{1}{2}||\boldsymbol{\theta} - \boldsymbol{u}||^2 + \lambda J(\boldsymbol{\theta})\right) \tag{12}$$

and yields the projection of $\boldsymbol{u}$ onto $\mathcal{C}$, that is, for any $\boldsymbol{u} \in \mathbb{R}^d$, one has $\mathcal{P}_\lambda(\boldsymbol{u}) \in \mathcal{C}$. Using this notation and simple algebra, the proximal gradient iterates defined in (9) can also be expressed by the projection

$$\hat{\boldsymbol{\theta}}^{(t+1)} = \mathcal{P}_{\frac{\lambda}{\nu}}\left(\hat{\boldsymbol{\theta}}^{(t)} + \frac{1}{\nu}\nabla l(\hat{\boldsymbol{\theta}}^{(t)})\right). \tag{13}$$

Keeping equations (10) and (11) in mind, the basic idea of proximal gradient methods becomes obvious: First, the penalty is ignored and a step toward the ML estimator via first-order methods for smooth optimization creates a search point. This search point then is projected onto the constraint region $\mathcal{C}$ in order to account for the non-smooth penalty term. As long as

the computation of the proximal operator is cheap, this method has the potential to be highly efficient. Beck & Teboulle (2009) proposed FISTA, a proximal gradient method that has the optimal convergence rate for this kind of algorithm. In order to achieve this rate, it uses a slightly different search point than the one in equation (13), but the core idea remains the same. Further technical details of FISTA (and our implementation) are given in the appendix.

3.2. Log-likelihood, score function and proximal operator

The main building blocks required to apply any proximal method to a particular penalized log-likelihood problem are formulas for the log-likelihood and its gradient as well as an efficient way of computing the proximal operator associated with the chosen penalty. We now derive them for the special case of CATS Lasso, that is, for multinomial logit models with the penalty given in (5).

Log-likelihood of multinomial logit models

For each actual observation $y_i \in \{1, \dots, k\}$, we define, for $r = 1, \dots, k-1$, a set of pseudo-observations y_{ir}, given by $y_{ir} = 1$ if $y_i = r$ and $y_{ir} = 0$ otherwise. With linear predictors η_{ir} of the form

$$\eta_{ir} = \beta_{r0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_r + (\mathbf{w}_{ir} - \mathbf{w}_{ik})^T \boldsymbol{\alpha}, \qquad r = 1, \dots, k-1,$$

the log-likelihood can conveniently be written as

$$l(\boldsymbol{\theta}) = \sum_{i=1}^{n} \left(\sum_{r=1}^{k-1} y_{ir} \eta_{ir} - \log(1 + \sum_{s=1}^{k-1} \exp(\eta_{is})) \right). \tag{14}$$

Note that the logit link used in multinomial logit models is the canonical link for the multinomial distribution. Following arguments of Fahrmeir & Kaufmann (1985), this automatically yields concavity of (14).

Score function

We partition the score function, which is defined as the gradient of the log-likelihood, in the following way:

$$\nabla l(\boldsymbol{\theta})^T = s(\boldsymbol{\theta})^T = \Big(s(\boldsymbol{\beta}_{\bullet 0})^T \,|\, s(\boldsymbol{\beta}_{\bullet 1})^T \,|\, \dots \,|\, s(\boldsymbol{\beta}_{\bullet p})^T \,|\, s(\boldsymbol{\alpha})^T \Big).$$

To be able to give $s(\boldsymbol{\theta})$ in a concise form, we use the following notation: For $r = 1, \dots, k-1$, let $\mathbf{y}_r^T = (y_{1r}, \dots, y_{nr})$ and $\boldsymbol{\pi}_r^T = (\pi_{1r}, \dots, \pi_{nr})$ denote vectors that pool the observations and estimated probabilities for category r across all observations. Additionally, let $\boldsymbol{x}_j^T =$

$(x_{1j}, \ldots, x_{nj})$ denote the vector of all observations of the j-th predictor. Furthermore, define

$$\underset{n\times L}{\boldsymbol{V}_r} = \begin{pmatrix} v_{11r} & \cdots & v_{1Lr} \\ \vdots & \ddots & \vdots \\ v_{n1r} & \cdots & v_{nLr} \end{pmatrix}, \quad \underset{n(k-1)\times 1}{\boldsymbol{y}} = \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_{k-1} \end{pmatrix} \quad \text{and} \quad \underset{n(k-1)\times 1}{\boldsymbol{\pi}} = \begin{pmatrix} \boldsymbol{\pi}_1 \\ \vdots \\ \boldsymbol{\pi}_{k-1} \end{pmatrix},$$

where $v_{ilr} = w_{ilr} - w_{ilk},\ r = 1, \ldots, k-1$ denotes the effect with which the category-specific predictors enter the model. The overall model matrix of size $n(k-1) \times \big((p+1)(k-1) + L\big)$ is

$$\boldsymbol{Z} = \left(\begin{array}{ccc|ccc|c|ccc|c} \mathbf{1}_n & & & \boldsymbol{x}_1 & & & \ldots\ldots & \boldsymbol{x}_p & & & \boldsymbol{V}_1 \\ & \ddots & & & \ddots & & & & \ddots & & \vdots \\ & & \mathbf{1}_n & & & \boldsymbol{x}_1 & & & & \boldsymbol{x}_p & \boldsymbol{V}_{k-1} \end{array}\right).$$

With these definitions, the score function, partitioned as above, is given by

$$s(\boldsymbol{\theta}) = \boldsymbol{Z}^T(\boldsymbol{y} - \boldsymbol{\pi}). \tag{15}$$

Proximal operator

Let $\tilde{\boldsymbol{\theta}} \in \mathbb{R}^d$ denote a generic search point that is equivalently partioned as the overall parameter vector $\boldsymbol{\theta}$ of the considered multinomial logit model, that is, $\tilde{\boldsymbol{\theta}}^T = (\tilde{\boldsymbol{\beta}}_{\bullet 0}^T, \tilde{\boldsymbol{\beta}}_{\bullet 1}^T, \ldots, \tilde{\boldsymbol{\beta}}_{\bullet p}^T, \tilde{\boldsymbol{\alpha}}^T)$. Additionally, define $\lambda_1 = \lambda\psi$ and $\lambda_2 = \lambda(1-\psi)$. Then the CATS-penalty from (5) can be written as

$$\lambda J(\boldsymbol{\theta}) = \lambda_1 \sum_{j=1}^{p} \phi_j \, ||\boldsymbol{\beta}_{\bullet j}|| \ + \ \lambda_2 \sum_{l=1}^{L} \varphi_l \, |\alpha_l|.$$

The projection of the search point $\tilde{\boldsymbol{\theta}}$ on the constraint region belonging to this penalty is then given by the proximal operator

$$\mathcal{P}_\lambda(\tilde{\boldsymbol{\theta}}) = \underset{\boldsymbol{\theta} \,\in\, \mathbb{R}^d}{\operatorname{argmin}} \ \left(\frac{1}{2} ||\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}||^2 + \lambda J(\boldsymbol{\theta}) \right).$$

Due to the block-separability of this specific penalty, the proximal operator can be written as the sum

$$\mathcal{P}_\lambda(\tilde{\boldsymbol{\theta}}) = \sum_{j=0}^{p} \mathcal{P}_{\lambda_1}(\tilde{\boldsymbol{\beta}}_{\bullet j}) + \sum_{l=1}^{L} \mathcal{P}_{\lambda_2}(\tilde{\alpha}_l),$$

where

$$\mathcal{P}_{\lambda_1}(\tilde{\boldsymbol{\beta}}_{\bullet 0}) = \underset{\boldsymbol{\beta}_{\bullet 0} \in \mathbb{R}^{k-1}}{\operatorname{argmin}} \frac{1}{2}||\boldsymbol{\beta}_{\bullet 0} - \tilde{\boldsymbol{\beta}}_{\bullet 0}||^2 = \tilde{\boldsymbol{\beta}}_{\bullet 0},$$

$$\mathcal{P}_{\lambda_1}(\tilde{\boldsymbol{\beta}}_{\bullet j}) = \underset{\boldsymbol{\beta}_{\bullet j} \in \mathbb{R}^{k-1}}{\operatorname{argmin}} \left(\frac{1}{2}||\boldsymbol{\beta}_{\bullet j} - \tilde{\boldsymbol{\beta}}_{\bullet j}||^2 + \lambda_1 \phi_j ||\boldsymbol{\beta}_{\bullet j}|| \right), \quad j = 1, \ldots, p, \tag{16}$$

$$\mathcal{P}_{\lambda_2}(\tilde{\alpha}_l) = \underset{\alpha_l \in \mathbb{R}}{\operatorname{argmin}} \left(\frac{1}{2}(\alpha_l - \tilde{\alpha}_l)^2 + \lambda_2 \, \varphi_l |\alpha_l| \right), \quad l = 1, \ldots, L. \tag{17}$$

With $(u)_+ = \max(u, 0)$, the following analytic solutions to (16) and (17) can easily be derived from the Karush-Kuhn-Tucker conditions:

$$\mathcal{P}_{\lambda_1}(\tilde{\boldsymbol{\beta}}_{\bullet j}) = \left(1 - \frac{\lambda_1 \phi_j}{||\tilde{\boldsymbol{\beta}}_{\bullet j}||}\right)_+ \tilde{\boldsymbol{\beta}}_{\bullet j}, \tag{18}$$

$$\mathcal{P}_{\lambda_2}(\tilde{\alpha}_l) = \left(1 - \frac{\lambda_2 \, \varphi_l}{|\tilde{\alpha}_l|}\right)_+ \tilde{\alpha}_l. \tag{19}$$

3.3. Choice of tuning parameters

We choose the main tuning parameter λ based on cross-validation. But in the general case, the additional tuning parameter ψ, which balances the penalization between parameters belonging to global and to category-specific predictors, has to be chosen for given λ. We performed various simulation studies (not shown) in which we simultaneously cross-validated over both λ and ψ. The ψ estimated by cross-validation always remained within the interval $[0.35, 0.65]$. In addition, the difference between the model with cross-validated ψ and the model using the default value of $\psi = 0.5$ was very minor. It seems that the default values for the weights ϕ_j and φ_l are already balancing out the penalty well enough. For efficiency reasons, we therefore recommend to choose only λ via cross-validation and to keep ψ fixed at 0.5.

For all simulations and the real data example in this paper, we used 10-fold cross-validation based on the deviance. Alternatively, model selection criteria like AIC can be employed, using the effective degrees of freedom given in Yuan & Lin (2006). For CATS Lasso, the formula is

$$\hat{\mathrm{df}} = (k-1) + \sum_{j=1}^{p} \left(\mathcal{I}(||\hat{\boldsymbol{\beta}}_{\bullet j}|| > 0) + \frac{||\hat{\boldsymbol{\beta}}_{\bullet j}||}{||\hat{\boldsymbol{\beta}}_{\bullet j}^{ML}||}(k-2) \right) + \sum_{l=1}^{L} \mathcal{I}(|\hat{\alpha}_l| > 0),$$

where the first term corresponds to the unpenalized intercept vector and $\mathcal{I}(\cdot)$ denotes the indicator function.

4. Simulation Study

Before analyzing the German Longitudinal Election Study, we first give the results of a small simulation study which illustrates the performance of our approach. In particular, four different

versions of CATS Lasso can be derived by using or not using adaptive weights and/or ML refitting. The simulations demonstrate the performance of these four variants under various settings and provide a guideline on when to use which of the alternative versions. The second aim of the simulation study is to compare our categorically structured approach with the unstructured Lasso. In order to simplify the presentation, we always refer to CATS Lasso as "CATS" and to the ordinary, unstructured Lasso as "Lasso" for the remainder of this section. All simulations were performed in `R` (R Development Core Team, 2011) using a self-written implementation that can be obtained from the authors.

4.1. Simulation Settings

Scenarios

We consider a small and a large model. The small model consists of 5 response categories with 4 relevant and 4 noise variables, giving a total of 8 global predictors. Additionally, 4 category-specific predictors are available of which 2 are relevant. The large model consists of 10 response categories, 60 global and 20 category-specific predictors of which 20 and 8 are relevant, respectively. For each model, the coefficients of the relevant predictors are independently drawn at random from the set $\{-1, -0.5, 0.5, 1, 1.5, 2, 2.5, 3\}$, yielding a true coefficient vector $\boldsymbol{\theta}^*$. The coefficients of noise variables are always zero.

The global and category-specific predictors were drawn from a multivariate Gaussian with an equi-correlation of 0.2 (small model) or 0.6 (large model) between all predictors. Using the predictors and true coefficient vector $\boldsymbol{\theta}^*$, the true probabilities for each observation were computed and then used to draw the response from a multinomial distribution.

The small model is tested for $n = 40$ and $n = 200$, the large one for $n = 500$ and $n = 3000$. Including intercepts, the size of the overall parameter vector to be estimated is $d = 40$ for the small model and $d = 569$ for the large model. Hence, both models are tested for the two cases of data-rich situations and of less observations than parameters. For all settings, the true coefficient vector is drawn once, followed by 100 replications of data generation and model estimation.

Comparison of Methods

To compare the methods, their estimation and prediction accuracy as well as variable selection performance are evaluated. First, with $\hat{\boldsymbol{\theta}}^{(i)}$ denoting the estimator for the i-th replication, we define the squared error for this replication as $(\hat{\boldsymbol{\theta}}^{(i)} - \boldsymbol{\theta}^*)^T(\hat{\boldsymbol{\theta}}^{(i)} - \boldsymbol{\theta}^*)/d$. From these squared errors, we compute the mean squared error (MSE). Second, prediction accuracy is evaluated by drawing a test set of $n_{\text{test}} = 3n$ new observations from the true model and then computing the predictive deviance on this test set. Third and last, we report the false positive and false negative rates (in terms of variable selection). The false positive rate (FPR) for variable selection is the percentage of noise variables whose coefficient vector is incorrectly estimated

as non-zero. The false negative rate (FNR) for variable selection is the percentage of relevant predictors whose estimated coefficient vector is falsely set to zero.

4.2. Results

Small Model

The simulation results for the small model are summarized in Figure 1. The methods compared are the ML estimator, CATS in simple, adaptive ("ada"), refitted ("rf") and adaptive plus refitted form as well as the unstructured Lasso in the same four variants as CATS. The left column of Figure 1 gives the results for the case with more observations than parameters, the right column contains results for the $n \leq d$ case. The first row shows squared errors and the second one the predictive deviance on a test set of, respectively, $n_{\text{test}} = 600$ (left column) and $n_{\text{test}} = 120$ (right column) new observations. Both quantities are given as boxplots to visualize variability and outlier behavior of these criteria for the various estimators. Extreme values were omitted for better clarity of the plots. In the third row, false positve (gray) and false negative (black) rates are shown.

For $n = 200$ (left column), one can clearly see that all regularized estimators perform better than the ML estimator in terms of MSE. Moreover, the large box indicates that the ML estimator is rather instable. For both CATS and Lasso, the adaptive version without refitting performs best in terms of MSE and prediction accuracy. Comparing each of the four variants between CATS and Lasso, CATS always comes out slightly ahead. However, the biggest advantage of CATS over Lasso is visible in the false positive and false negative rates, shown in the bottom left plot. While plain CATS performs quite poorly, both refitting and adaptive weights improve the variable selection of CATS by a large margin. By contrast, Lasso shows substantially worse variable selection properties. With refitting, the best method in terms of variable selection for both Lasso and CATS, about 25% of irrelevant variables were not detected by Lasso, while CATS missed only about 6%.

For $n = 40$ (right column), a model is fitted with as many parameters as observations. In this case, "true" ML estimates do not exist anymore, so we added a very small ridge penalty whenever an unpenalized fit for this model was required. This is the case for the ML estimator as well as all methods with refit. It can immediately be seen from Figure 1 that for $n = 40$, all these methods in which the final model is obtained by an unpenalized ML fit perform significantly worse. In terms of MSE and prediction accuracy, differences between Lasso, CATS and their respective adaptive versions are minor. When it comes to variable selection, however, adaptive CATS shows by far the best performance of all considered methods and includes only half as many noise variables as the best version of Lasso. It is noteworthy that the false negative rates are much higher throughout all methods than for $n = 200$.

To sum up the results for the small model, CATS substantially outperforms Lasso in terms of variable selection and is better or on par with Lasso in terms of estimation and prediction

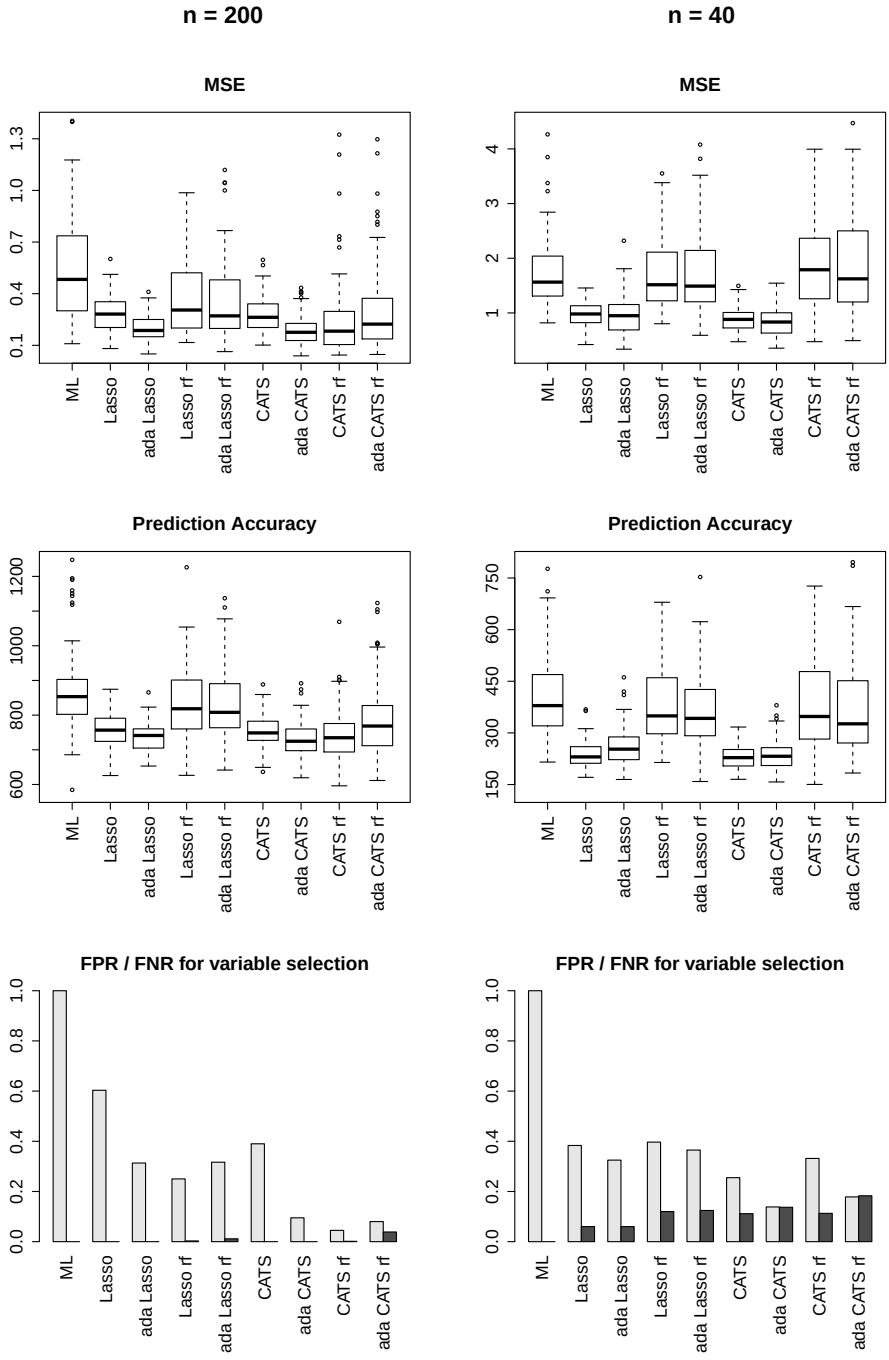

Figure 1: Simulation results for the small model: MSE, predictive deviance and False Positive Rate (FPR, gray) & False Negative Rate (FNR, black) for $n = 200$ (left column) and $n = 40$ (right column).

accuracy. The adaptively weighted estimator shows by far the best performance in the $n \leq d$ case and also gave the highest estimation and prediction accuracy for the case with many ob-

servations. Only in terms of variable selection in a data-rich environment, it was outperformed marginally by refitting.

Large model

The results of the simulations for the large model are shown in Figure 2. The left column contains the $n = 3000$ setting. The differences in MSE between the various regularized estimators are rather small. However, the top left plot of Figure 2 shows that the ML estimator for each coefficient has, on average, a distance of about 0.7 from the true value. Thus, unpenalized estimation is unable to yield accurate estimates even though 3000 observations are available to estimate 569 parameters. The second plot of the left column shows the predictive deviance on $n_{\text{test}} = 9000$ observations. Once again, the ML estimator is vastly outperformed by all regularized estimators. Among the different versions of Lasso and CATS, those with adaptive weights and no refitting show the best prediction accuracy. Comparing these two methods, adaptive CATS slightly outperforms adaptive Lasso. The biggest difference between CATS and Lasso is again seen in their false positive and false negative rates. Unless one uses adaptive weights and refitting, Lasso shows unacceptably high rates of false positives. Turning to CATS, this plot confirms the bias issues of simple CATS that were discussed in Section 2.4 and emphasizes that adaptive weights, refitting or both should be used. Once the bias is accounted for, CATS shows extremely low error rates which are magnitudes below those of Lasso.

The right column of Figure 2 shows the results for the large model with $n = 500$, that is, for a setting in which significantly less observations than parameters to be estimated are available. Once again, a small ridge penalty is added to the ML estimator and during refitting as an unregularized estimator does not exist in this scenario. In terms of MSE, the differences between the regularized estimators are negligible while the ML estimator - as expected - shows an extremely poor performance. When it comes to the predictive deviance on $n_{\text{test}} = 1500$ observations, the ML estimator was so bad that the axis had to be broken to show it within one plot. Additionally, this plot shows that refitting techniques, in which the final model is obtained by an unpenalized ML refit on the active set of the regularized estimator, perform substantially worse. On top of that, one can see that adaptive CATS distinctly outperformed Lasso. The plot on the bottom right of figure 2 shows that all methods with refitting had very high false negative rates. Moreover, all versions of Lasso have both higher false positive and false negative rates than their corresponding counterpart of CATS. For the considered scenario, adaptive CATS offers the best variable selection properties by a large margin. It never removed an important predictor from the model and is able to detect roughly 80% of all noise variables - keeping in mind the massive size of the model and the low number of observations, we consider this a very encouraging and satisfying result.

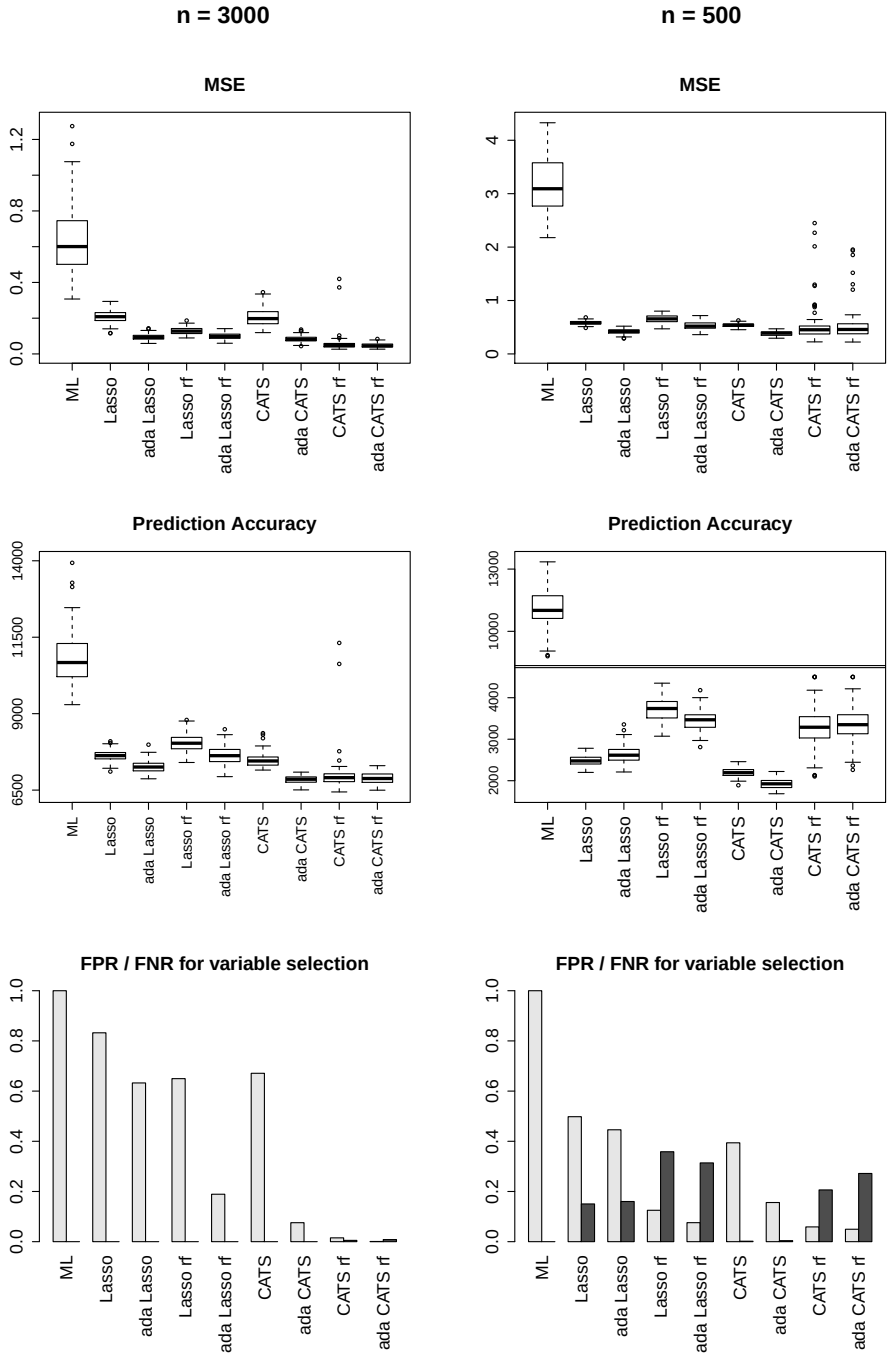

Figure 2: Simulation results for the large model: MSE, predictive deviance and False Positive Rate (FPR, gray) & False Negative Rate (FNR, black) for $n = 3000$ (left column) and $n = 500$ (right column).

Summary

To sum up the simulation study, regularized estimators offer more accurate estimates and prediction than the unpenalized ML estimator - even if substantially more observations than parameters to be estimated are available. CATS outperformed the ordinary Lasso in all settings and in terms of all criteria that were used to compare them. The degree by which CATS outperforms Lasso grows with the number of response categories, in particular in terms of variable selection. Comparing the simple, adaptive and refitted version of CATS, adaptive CATS is clearly recommended if few observations are available, relative to the size of the model. In $d > n$ situations, the ML estimator does not exist and we replace it by a ridge estimate with small tuning parameter, see also Section A.2 in the appendix. This pseudo-unpenalized estimator is used both in the adaptive weights and in the refit. Our simulation results indicate that the adaptive version of CATS (and also Lasso) is less affected by an instability of this estimator than refitting approaches.

In data-rich situations, refitting performed marginally better than adaptive weights when it comes to variable selection, but slightly worse in terms of MSE and prediction accuracy. The combination of adaptive weights and refitting did not offer an advantage, but was more unstable in several cases and is therefore not recommended. Thus, prediction and variable selection are conflicting in the $n > d$ case and none of our considered methods can optimize both at once. Even though the conflict between variable selection and estimation performance is not very pronounced in our simulations, it is common for penalization approaches and was already discussed in the literature, for example by Leng et al. (2006). However, from the boxplots in the left columns of figures 1 and 2 one can see that the refitting approach does not only perform worse than the adaptive one in terms of MSE and prediction but also has a larger variability and produces more outliers. Due to this lower robustness of refitted CATS, we also prefer the adaptive version of CATS for $n > d$ settings.

5. Regularized Analysis of Party Choice in Germany

5.1. Data Description

The data we consider come from the German Longitudinal Election Study. We focus on a dataset that contains the party on which 816 study participants intended to vote during the 2009 election for the German parliament, the Bundestag. To be able to explain their party choice behavior, nine individual characteristics of the voters are available. These global predictor are gender (1: male, 0: female), regional provenance (west; 1: former West Germany, 0: former East Germany), age (mean-centered), union (1: membership in a union, 0: otherwise), high school degree (1: yes, 0: no), unemployment (1: currently unemployed, 0: otherwise), political interest (1: less interested, 0: very interested), satisfaction with the functioning of democracy (democracy; 1: not satisfied, 0: satisfied) and religion (0: Protestant, 1: Catholic,

2: otherwise).

Additionally, the interviewed persons rated the position of the parties on political issues on a scale from 1 to 11. They also rated their own position on these topics on the same scale. From this information, the distance between the voters' own position (the "ideal point") and the perceived position of the party was computed, that is, the absolute value of the difference of the two variables. The resulting distances take values between 0 and 10 and can be considered measures of agreement between voter and party and are used as category-specific predictors. This approach has strong connections to spatial election theory which assumes that each party is characterized by a position in a finite-dimensional space, with each dimension corresponding to one political issue. Spatial election theory then explains the party choice of voters by a utility function that depends on the distance between the voter's own position and that of the parties within the space of policy-dimensions. For further details on spatial election theory, see, for example, Thurner & Eymann (2000). Here, four political issues were considered: the so-called socioeconomic dimension ("do you prefer low taxes and few public spending or high taxes with lots of public spending?"), the attitude toward immigration and nuclear energy as well as the positioning on a political left-right scale.

5.2. Results and Interpretation

To model the party choice behavior of voters in Germany, a multinomial logit model with the chosen party as response variable is used with the explanatory variables described in the previous paragraph. Of the five available parties, the CDU was chosen as reference. We fitted this model using adaptive CATS Lasso, that is, a penalized multinomial logit model with the penalty from (5) and the weights given in (7). The only exception is the religion of the voter which enters the model with two dummies, so that the advanced methodology from Section 2.3 is used to jointly penalize both dummies.

Figure 3 shows the resulting coefficient paths for those variables that were selected by CATS Lasso. They show how the different coefficients change if the tuning parameter λ is varied with λ plotted on a logarithmic scale. The left end of the coefficient paths corresponds to zero penalization and thus shows the ML estimator. The vertical line marks the value of λ that was chosen by 10-fold crossvalidation. The global predictors that were removed from the model are gender, unemployment and political interest. One can immediately see the structured selection performed by CATS Lasso. The variable high school was selected by our method, but appears to be on the edge of being removed from the model. Removing the variables democracy, age, west and union would require substantially more penalization. The estimated coefficients for the global variables are given in Table 1.

While the FDP and the Left Party perform about as well as the CDU no matter if the voter comes from former East or West Germany, both the SPD and the Green Party are much stronger in the west. The coefficients of age are negative for all four parties, which means

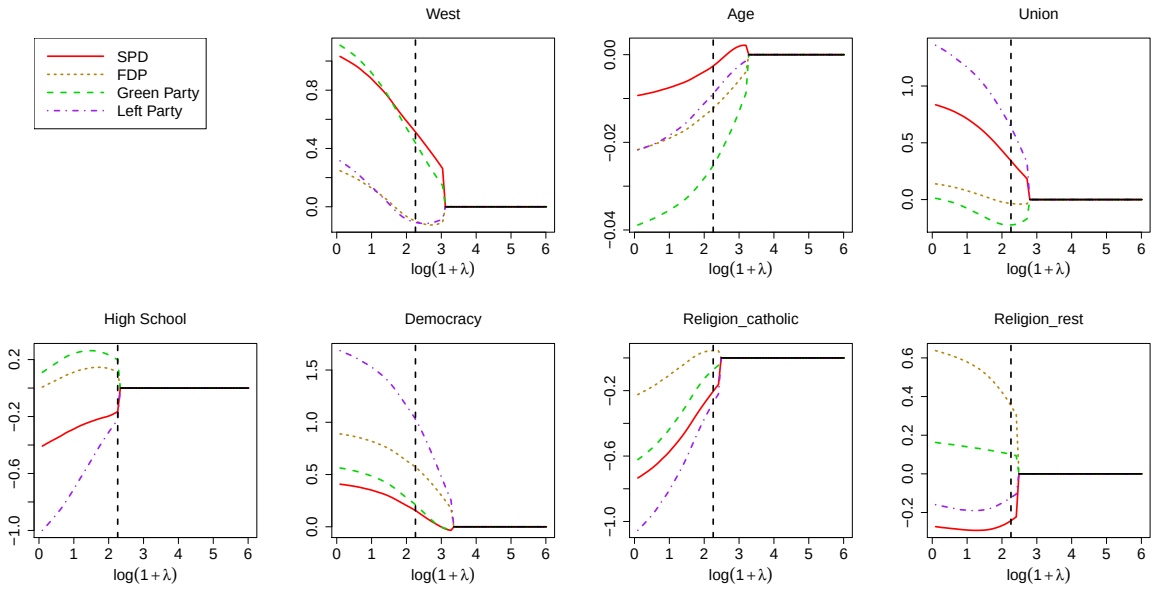

Figure 3: Coefficient buildups for the selected global variables of party choice data.

that the reference party, the CDU, is strongest among elderly people. Growing age is most detrimental to the Green Party, which matches expectations as this party was founded in 1980, so that many older people were already adults with a developed political preference before this party even existed. Voters that are members in a labor union are distinctly more likely to vote for the SPD or the Left Party, which again is expected as these parties are traditionally strong among blue-collar workers. Voters with a high-school degree are more likely to vote in favor of the FDP or the Green Party, which focus on liberal and environmental topics, respectively, whereas the SPD and the Left Party have more success among voters without a high-school degree. Not being satisfied with the functioning of democracy decreases the chances of voting for the CDU, which is not surprising as the CDU is the most conservative party. The Left Party, on the other hand, benefits strongly if the voter is discontent with democracy. Therefore, the Left Party and, to a lesser degree, the FDP can be considered the parties of choice for "protest voters". Catholic people prefer the FDP or the CDU and are noticeably less likely to vote for the SPD, Green Party or Left Party. The success of the CDU with Catholic voters was expected as it is the most conservative among the major parties in Germany and defending Christian values is a core part of the party agenda. It is slightly surprising, however, that the FDP seems to be even more successful among Catholic voters. Being neither Protestant nor Catholic increases the likelihood of voting for the FDP or the Green Party.

To interpret the effect of the category-specific predictors, it is helpful to recall the structure of the linear predictor for party r, which has the form $\eta_{ir} = \beta_{r0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_r + (\mathbf{w}_{ir} - \mathbf{w}_{ik})^T \boldsymbol{\alpha}$, where the vector $\boldsymbol{x}_i$ collects the nine global predictors that characterize the ith voter. The vector $\mathbf{w}_{ir}$ denotes the four-dimensional vector of distances between the positioning of the voter and the

Table 1: Estimated regression coefficients for the global predictors of the party choice data.

	SPD	**FDP**	**Green Party**	**Left Party**
Intercept	-0.821	-1.701	-1.645	-1.184
Gender	0	0	0	0
West	0.643	-0.034	0.615	-0.056
Age	-0.005	-0.015	-0.030	-0.013
Union	0.493	-0.005	-0.200	0.866
High School	-0.209	0.144	0.250	-0.376
Unemployment	0	0	0	0
Political Interest	0	0	0	0
Democracy	0.237	0.677	0.327	1.247
Religion Catholic	-0.334	0.021	-0.179	-0.454
Religion other	-0.280	0.464	0.116	-0.168

perceived position of party r for the socioeconomic dimension, immigration, nuclear energy and the left-right scale. As a reference point, $\mathbf{w}_{ik}$ contains the distances on these issues between the voter and the CDU. Hence, the variable $w_{isl}, s = 1, \ldots, k$ measures the disagreement between voter i and party s on the lth considered political topic, that is, low values correspond to high agreement and vice versa. The coefficient α_l is the effect of the quantity $w_{irl} - w_{ikl}$, which corresponds to the difference between the disagreement with the rth party and the reference party. If this difference is positive, this means that the voter shows stronger agreement with the CDU than with party r on the topic at hand, so that the odds of voting for this party instead of the CDU should decrease. Therefore, we expect the coefficients for the political issues to be negative.

All four political issues were selected by our method. The estimated parameters for the socioeconomic dimension, immigration, nuclear energy and the left-right scale are given by $\hat{\boldsymbol{\alpha}}^T = (-0.156, -0.110, -0.182, -0.617)$, respectively. The signs are negative as expected, meaning that the odds of voting for a party increase if the agreement between a voter and this party on a particular issue is higher than the agreement with the reference party. The corresponding coefficient paths are shown in Figure 4. The vertical line shows the λ chosen by cross-validation. One can see that the left-right scale has the largest influence on party choice, but it is also the most general of the four considered issues. Comparing Figures 3 and 4, one sees that the voter-specific predictors are removed much earlier than the agreement on political issues. The former are all removed from the model for any $\lambda > 26$, while $\lambda > 63$ is necessary to remove the weakest issue, so that the agreement on political issues can be considered a stronger predictor for party preference.

To conclude the analysis, we computed the AIC for both our regularized estimator and the

ML estimator. The AIC for CATS Lasso is 1670.75, the ML estimator has an AIC of 1692.31. Thus, our regularized method was able to find a model that performs significantly better than the unpenalized one and, at the same time, uses 33% less global predictors.

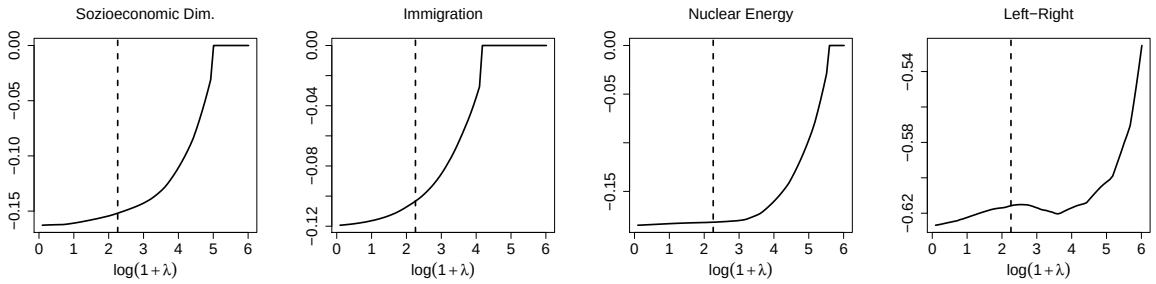

Figure 4: Coefficient buildups for category-specific variables of party choice data.

6. Concluding Remarks

In this paper, Categorically Structured Lasso (CATS Lasso), a new method for regularized estimation in multinomial logit models, is proposed. Multinomial models are designed for multi-category responses and thus are not ordinary but multivariate GLMs. CATS Lasso deals with the specific challenges arising from the multivariate structure in interpretation and model selection by exploiting the natural grouping within the coefficients of a multinomial logit model, with all coefficients that are linked to one predictor forming their own group. We have argued both theoretically and practically that all coefficients within such a parameter group should enter or leave the model simultaneously, so that true variable selection is performed. To achieve this, our procedure uses a structured penalization, in contrast to the unstructured approach of existing methods.

CATS Lasso is also generalized to include the important cases of categorical predictors and predictors that vary over response categories, so that it constitutes a conceptual extension of both the Lasso and the Group Lasso to more general multivariate regression models. Our `R` implementation uses an efficient algorithm and is the first to support the combination of regularized estimation and category-specific predictors.

In a simulation study, we have shown that CATS Lasso outperforms alternative regularization approaches for multinomial models in small and large as well as sparse and data-rich models. The biggest advantage is the vastly improved performance in terms of variable selection, but CATS Lasso also improves the accuracy with which the regression coefficients are estimated and future observations are predicted. An application on the modeling of party choice in Germany demonstrates the success of CATS Lasso on real-world problems. It showed that the gender, employment status and political interest of a voter do not influence his or her party preference given all other predictors.

A. Algorithmic details

A.1. The Fast Iterative Shrinkage Thresholding Algorithm (FISTA)

Recall that the iterates of proximal gradient methods for maximizing a penalized log-likelihood $l_p(\boldsymbol{\theta}) = l(\boldsymbol{\theta}) - \lambda J(\boldsymbol{\theta})$ are defined by

$$\hat{\boldsymbol{\theta}}^{(t+1)} = \underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\operatorname{argmin}} \left(Q_\nu(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(t)}) := -l(\hat{\boldsymbol{\theta}}^{(t)}) - \nabla l(\hat{\boldsymbol{\theta}}^{(t)})^T (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}^{(t)}) + \frac{\nu}{2} ||\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}^{(t)}||^2 + \lambda J(\boldsymbol{\theta}) \right). \quad (20)$$

As shown in Section 3.1, the proximal operator can be used to express the solutions to (20) by

$$\hat{\boldsymbol{\theta}}^{(t+1)} = \mathcal{P}_{\frac{\lambda}{\nu}} \left(\hat{\boldsymbol{\theta}}^{(t)} + \frac{1}{\nu} \nabla l(\hat{\boldsymbol{\theta}}^{(t)}) \right).$$

This means that a gradient step toward the ML estimator is performed, starting from the current iterate $\hat{\boldsymbol{\theta}}^{(t)}$, to create a search point, which is then projected on the penalty region by the proximal operator. The analysis in Beck & Teboulle (2009), however, showed that this procedure does not reach the optimal convergence rate and can be accelerated. To achieve this speed-up, the direction given by the current and the previous iteration is extrapolated with the help of deliberately chosen acceleration factors a_t:

$$\hat{\boldsymbol{\vartheta}}^{(t)} = \hat{\boldsymbol{\theta}}^{(t)} + \frac{a_{t-1} - 1}{a_t} (\hat{\boldsymbol{\theta}}^{(t)} - \hat{\boldsymbol{\theta}}^{(t-1)}).$$

Instead of the current iterate $\hat{\boldsymbol{\theta}}^{(t)}$, this extrapolated point $\hat{\boldsymbol{\vartheta}}^{(t)}$ is used as the starting point from which the step toward the ML estimator is taken, yielding an "extrapolated search point" which is then projected on the penalty region. Because in practice one does not know a suitable value for the inverse stepsize parameter ν, it is determined by a standard backtracking line search (see, e.g., Bertsekas et al., 2003). In detail, the FISTA method for minimizing (8) is given by

---

FISTA for computing maximum penalized log-likelihood estimates

---

(0) **Input:** Formulas for $l(\boldsymbol{\theta})$, $\nabla l(\boldsymbol{\theta})$, $J(\boldsymbol{\theta})$, $\mathcal{P}_\lambda(\cdot)$; starting values $\hat{\boldsymbol{\theta}}^{(0)}$, $\nu_0 > 0$; a tolerance ε.

(1) **Initialize:** Set $\hat{\boldsymbol{\theta}}^{(1)} = \hat{\boldsymbol{\theta}}^{(0)}$, $a_0 = 0$, $a_1 = 1$ and $t = 1$.

(2) **Extrapolate:** $\hat{\boldsymbol{\vartheta}}^{(t)} = \hat{\boldsymbol{\theta}}^{(t)} + \frac{a_{t-1}-1}{a_t} (\hat{\boldsymbol{\theta}}^{(t)} - \hat{\boldsymbol{\theta}}^{(t-1)})$.

(3) **Line search:**

(i) **Initialize:** $\nu = \nu_{t-1}$

(ii) **Projection:** $\hat{\boldsymbol{\theta}}^{(t+1)} = \mathcal{P}_{\frac{\lambda}{\nu}}\left(\hat{\boldsymbol{\vartheta}}^{(t)} + \frac{1}{\nu}\nabla l(\hat{\boldsymbol{\vartheta}}^{(t)})\right)$.

(iii) **Armijo rule:** End if

$$-l_p(\hat{\boldsymbol{\theta}}^{(t+1)}) \leq Q_\nu(\hat{\boldsymbol{\theta}}^{(t+1)}, \hat{\boldsymbol{\vartheta}}^{(t)}),$$

else multiply ν by 2 and repeat.

(4) **Update:**

$$\begin{aligned} a_{t+1} &= \frac{1 + \sqrt{1 + 4a_t^2}}{2}, \\ \nu_t &= \nu, \\ \hat{\boldsymbol{\theta}}^{(t+1)} &= \mathcal{P}_{\frac{\lambda}{\nu_t}}\left(\hat{\boldsymbol{\vartheta}}^{(t)} + \frac{1}{\nu_t}\nabla l(\hat{\boldsymbol{\vartheta}}^{(t)})\right). \end{aligned}$$

(5) **Convergence check:** Stop if

$$\frac{||l_p(\hat{\boldsymbol{\theta}}^{(t+1)}) - l_p(\hat{\boldsymbol{\theta}}^{(t)})||}{||l_p(\hat{\boldsymbol{\theta}}^{(t)})||} \leq \varepsilon,$$

else increase t by 1 and repeat steps (2) - (5).

A.2. Implementation

Numerical threshold

When computing CATS Lasso with our self-written implementation, it sometimes happens that the parameter estimates of some predictors are very small but non-zero - in such cases we typically observe $||\hat{\boldsymbol{\beta}}_{\bullet j}|| \in [0.001, 0.1]$ if the j-th predictor is among the variables affected by this phenomenom. To prevent our estimator from being affected by this numerical inefficiency, we use a very small and asymptotically diminishing threshold of order $O(n^{-0.4})$, i.e. the estimates of parameter groups with a norm below the threshold are set to zero.

Stabilizing ML estimates

ML estimates of the full model are required to compare our regularized estimator with the unpenalized model and to compute the adaptive weights from (7). However, we frequently observed a divergence of parameters to $\pm\infty$ even in the $d < n$ case. Roughly speaking, this happens when the data space allows for a perfect separation of the response categories. This is

always the case for $d > n$, but can also happen if the number of observations per parameter to be estimated is above 1, but rather small. The same problems were also observed and discussed by Friedman et al. (2010).

It is obvious that any kind of p-norm penalization of the parameters (with $p > 0$) will prevent this problem. Therefore, we propose to use a ridge penalty (Hoerl & Kennard, 1970) with a very small tuning parameter (say between 0.01 and 0.1) if coefficients are detected to be diverging:

$$J_R(\boldsymbol{\theta}) = \frac{\lambda_R}{2} ||\boldsymbol{\theta}||_2^2.$$

The proximal operator associated with this ridge penalty is given by

$$\mathcal{P}_{\lambda_R}(\tilde{\boldsymbol{\theta}}) = \frac{\tilde{\boldsymbol{\theta}}}{1 + \lambda_R}.$$

References

Beck, A. & Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM Journal on Imaging Sciences* **2**, 183–202.

Bertsekas, D., Nedić, A. & Ozdaglar, A. (2003). *Convex Analysis and Optimization*, Athena Scientific, Belmont.

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

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

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

Efron, B., Hastie, T., Johnstone, I. & Tibshirani, R. (2004). Least angle regression, *The Annals of Statistics*, 407–451.

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

Fan, J. & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties, *Journal of the American Statistical Association* **96**, 1348–1360.

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

Hastie, T., Tibshirani, R. & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Springer, New York.

Hoerl, A. & Kennard, R. (1970). Ridge regression: biased estimation for nonorthogonal problems, *Technometrics* **12**, 55–67.

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

Leng, C., Lin, Y. & Wahba, G. (2006). A note on the lasso and related procedures in model selection, *Statistica Sinica* **16**, 1273–1284.

McFadden, D. (1973). Conditional logit analysis of qualitative choice behaviour, *Frontiers in Econometrics*, ed. by P. Zarembka, Academic Press, New York.

Meier, L., van de Geer, S. & Bühlmann, P. (2008). The group lasso for logistic regression, *Journal of the Royal Statistical Society B* **70**, 53–71.

Nemirovski, A. (1994). Efficient methods in convex programming, *Lecture Notes*.

Nesterov, Y. (2004). *Introductory lectures on convex optimization: A basic course*, Kluwer Academic Publishers.

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.

Thurner, P. & Eymann, A. (2000). Policy-specific alienation and indifference in the calculus of voting: A simultaneous model of party choice and abstention, *Public Choice* **102**, 49–75.

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

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. & Knight, K. (2005). Sparsity and smoothness via the fused lasso, *Journal of the Royal Statistical Society B* **67**, 91–108.

Turlach, B., Venables, W. & Wright, S. (2005). Simultaneous variable selection, *Technometrics* **47**, 349–363.

Tutz, G. (2012). *Regression for Categorical Data*, Cambridge University Press, Cambridge.

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

Wang, H. & Leng, C. (2008). A note on adaptive group lasso, *Computational Statistics and Data Analysis* **52**, 5277–5286.

Yellott, J. (1977). The relationship between Luce's choice axiom, Thurstone's theory of comparative judgement, and the double exponential distribution, *Journal of Mathematical Psychology* **15**, 109–144.

Yuan, M. & Lin, Y. (2006). Model selection and estimation in regression with grouped variables, *Journal of the Royal Statistical Society B* **68**, 49–67.

Zou, H. (2006). The adaptive lasso and its oracle properties, *Journal of the American Statistical Association* **101**, 1418–1429.

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