Regularization Approaches for Generalized Linear Models and Single Index Models

Sebastian Petry

München 2011

Regularization Approaches for Generalized Linear Models and Single Index Models

Sebastian Petry

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig–Maximilians–Universität
München

vorgelegt von
Sebastian Petry
aus München

München, den 20.09.2011

Erstgutachter: Prof. Dr. Gerhard Tutz, LMU München
Zweitgutachter: Prof. Dr. Korbinian Strimmer, Universität Leipzig
Tag der mündlichen Prüfung: 28.11.2011

Zusammenfassung

Die Grundlage dieser Arbeit sind generalisierte lineare Modelle (GLMs). Im Gegensatz zum gewöhnlichen linearen Modell erlauben sie zum einen die Verteilung der Responsevariable und zum anderen den Einfluss des Prädiktors auf die Responsevariable nicht linear durch die Linkfunktion zu modellieren. Hierbei wird vorausgesetzt, dass die Verteilung der Responsevariable aus der Klasse der Exponentialfamilien stammt und die Linkfunktion bekannt sowie zweimal stetig diffenrenzierbar und monoton steigend ist. Die Schätzung der über das Modell festgelegten Parameter, stellt hinsichtlich einer möglichst guten Prognose in vielen Datensituationen eine Herausforderung dar. Als Strategien zur Bewältigung dieser Herausforderung haben sich Variablen-Selektion (lediglich eine Teilmenge von Parametern wird ungleich Null geschätzt) und Variablen-Grouping (die Parameter verschiedener Kovariablen werden gleich geschätzt) etabliert.

Bei GLMs haben sich in den letzten Jahren neben anderen Methoden Shrinkage-Verfahren für Variablen-Selektion und -Grouping bewährt. Skrinkage-Verfahren zeichnen sich durch ihre jeweiligen Penalisierungsregionen aus. Eine Klasse von Penalisierungsregionen, die diese Eigenschaften auf den Schätzer induziert, sind spezielle Polytope. Es werden theoretische Ergebnisse zu Polytopen, die die beiden Effekte ermöglichen, präsentiert und hierauf basierend neue Penalisierungsregionen entwickelt. Hierbei wird auf die Lösungsalgorithmen eingegangen. In Simulationsstudien und realen Datensituationen zeigt sich, dass die vorgestellten Methoden die etablierten Konkurrenzverfahren in vielen Situationen dominieren.

Eine Verallgemeinerung der GLMs sind (generalisierte) Single-Index Modelle (SIMs). Hierbei handelt es sich um GLMs mit unbekannter Linkfunktion. Neben dem Parametervektor des linearen Prädiktors ist zusätzlich die Linkfunktion innerhalb des Lösungsalgorithmus zu schätzen. Es werden Algorithmen zur Schätzung von SIMs mit linearem und additiven Prädiktor entwickelt. Hierbei werden vor dem Hingrund der Variablen-Selektion sowohl Boosting als auch Penalisierungsansätze verfolgt. Neben den Algorithmen steht insbesondere die Auswirkung von fehlspezifierten Linkfunktionen auf Variablen-Selektion im Fokus dieser Arbeit. Es zeigt sich in Simulationstudien und Echtdatenbeispielen, dass das simultane Schätzen von Linkfunktion und linearen beziehungsweise nicht parametrischen Einflusstermen sowohl die Vorhersage als auch die Schätzung der Einflussterme und die Variablenselektion verbessert.

Summary

This thesis is based on Generalized Linear Models (GLMs). In contrast to (ordinary) linear models GLMs are able to model the distribution of the response variable and a non linear influence of the linear predictor on the response variable. Hereby it is assumed that the distribution of the response variable is from a simple exponential family and the link function is known, twice differentiable, and monotonically increasing. The estimation of the model parameter in consideration of best prediction is a challenge in many data situations. Established strategies for performing good prediction in complex data situations are variable selection (only a subset of parameters are estimated non zero) and variable clustering (parameter of different covariates are estimated equal).

In the last year for variable selection and variable clustering in GLMs shrinkage procedures become very popular. Shrinkage procedures are characterized by its penalty region or term. A wide class of penalty regions which induce variable selection and clustering are special types of polytopes. Theoretical results about these polytopes are presented and new shrinkage procedures are developed. Beyond the evaluation of the new procedures the focus is on corresponding the algorithms. It is shown by simulation studies and real data problems that the new methods outperform established methods in many situations.

A generalization of GLMs are (generalized) single index models (SIMs). We consider SIMs as GLMs with unknown link function. For the solution of SIMs the parameter vector of the linear predictor and the link function must be estimated inner the algorithm. Algorithms for the estimation of SIMs with linear and additive predictor are presented. Against the background of variable selection we use boosting as well as penalization approaches for the estimation of SIMs. Apart the algorithms the impact of misspecified link functions on the variable selection is a central point of this thesis. In simulation studies and real data problems it is illustrated that simultaneous estimation of the link function and linear respectively non parametric terms improves prediction, estimation of influential terms, and variable selection.

Vorwort

Diese Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Angestellter am Institut für Statistik der Ludwig-Maximilians-Universität München.

Meinen ersten Dank richte ich an meinen Doktorvater Herrn Prof. Gerhard Tutz, der mir nach dem Studium die Möglichkeit zur Promotion eröffnete und dessen konstruktive Denkanstösse diese Arbeit prägten. Das Resultat intensiver Gespräche ist diese Arbeit. Ich danke Ihm für sein immer währendes Verständnis in vielen Bereichen und die sehr gute Zusammenarbeit. Des weiteren möchte ich Herrn Prof. Korbinian Strimmer dafür danken, dass er die Aufgabe als Zweitgutachter für diese Arbeit übernommen hat.

Ein herzliches *"Vergelt's Gott"* meinerseits gilt allen (ehemaligen) Mitarbeitern des Seminars für Stochastik — den Tutzlern — für den intensiven Austausch über die Themenbereiche dieser Arbeit. Namentlich nennen möchte ich Stephanie Rubenbauer, mit der ich lange Zeit das Zimmer teilte. Neben dem fachlichen Input trug Deine motivierenden und liebenswerten Art einen wesentlichen Anteil zur Fertigstellung dieser Arbeit bei. Außerdem danke ich den "beiden Jans" für all den fachlichen Austausch und die Impulse, die meine Arbeit am Institut prägten. Desweiteren möchte ich allen Mitarbeitern des Institus für Statistik danken mit denen ich in den letzten Jahren auch über diese Arbeit hinaus eng zusammenarbeiten durfte. Namentlich erwähnen möchte ich an dieser Stelle Herrn Prof. Thomas Augustin und Herrn Dr. Christian Heumann.

Meine Arbeit am Institut wurde in wesentlichen Teilen aus DFG-Projektmitteln (DFG Projekt TU/4-1) finanziert. Somit gebührt mein Dank auch der Deutschen Forschungsgemeinschaft.

Einen besonderen Dank möchte ich meinen Eltern, Karl-Heinz und Dorothea, sowie meiner Großmutter Ruth aussprechen, die mir das Studium ermöglichten auf dem diese Promotion fusst. Danke, dass Ihr nie aufgehört habt an mich zu glauben und mich immer weiter motiviert habt. Auch die stete Motivation meiner Schwestern und aus meinem Freundeskreis war mir über die Jahre hinweg ein wichtiger Antrieb.

Abschließend danke ich meiner Frau Judith, die mit viel Geduld und all Ihrer Liebe meine Arbeit begleitet hat, auch wenn ich Ihr die Frage *"Was machst Du da eigentlich?"* wohl nie wirklich schlüssig beantworten konnte. Ohne Dich wären das Alles nicht möglich gewesen. Viele Teile dieser Arbeit begleitete — wenn auch großenteils im Verborgenen — unsere Tochter, Theresa. Danke für Dein Strampeln, Schreien und Lächeln. Du hast Alles gegeben um Deinen Beitrag zu dieser Arbeit zu leisten. Euch beiden möchte ich diese Arbeit widmen.

München, im September 2011 *Sebastian Petry*

Contents

Contents **ix**

1 Introduction and Remarks **1**

I Regularization Approaches for Generalized Linear Models **5**

2 Shrinkage by Polytopes **7**
2.1 Introduction . . . 7
2.2 Polytopes as Constraint Region . . . 9
2.2.1 Some Concepts in Polytope Theory . . . 10
2.2.2 LASSO . . . 12
2.2.3 OSCAR . . . 14
2.3 The V8 procedure . . . 17
2.3.1 The V8-polytope . . . 17
2.3.2 Solving Polytopal Constrained Regression Problems . . . 19
2.4 Simulation study . . . 22
2.5 Data Example . . . 26
2.6 Concluding Remarks . . . 27

3 Pairwise Fused Lasso **29**
3.1 Introduction . . . 29
3.2 Pairwise Fused Lasso (PFL) . . . 31
3.2.1 Solving the Penalized ML Problem . . . 35
3.3 Simulation Study . . . 39
3.4 Data Example . . . 46
3.5 Concluding Remarks . . . 49

4 OSCAR for GLMs **51**
4.1 Introduction . . . 51
4.2 Generalized Linear models . . . 52
4.3 The OSCAR Penalty Region . . . 53
4.4 The glmOSCAR Algorithm . . . 55

4.5 Simulation Study . . . 57
4.6 Application . . . 65
4.7 Conclusion and Remarks . . . 66

II Regularization Approaches for Single Index Models 69

5 The FlexLink Boost 71
5.1 Introduction . . . 71
5.2 Estimation . . . 73
5.2.1 Data Generating and Approximating Model . . . 73
5.2.2 Estimation of Parameters Including Variable Selection . . . 75
5.3 Simulation Studies . . . 80
5.4 Modified Estimator and Selection of Predictors . . . 84
5.5 Applications . . . 87
5.5.1 Medical Care Data . . . 87
5.5.2 Noisy Miner Data . . . 92
5.6 Concluding Remarks . . . 93

6 GAMs with Flexible Link Function 95
6.1 Introduction . . . 95
6.2 Flexible Link with Additive Predictor (FLAP) . . . 98
6.2.1 Estimation Procedure . . . 99
6.2.2 Algorithm . . . 103
6.3 Simulation Study . . . 105
6.4 Data Example . . . 110
6.5 Conclusion and Perspectives . . . 114

7 L1-Penalized Single Index Models 117
7.1 Introduction . . . 117
7.2 Data Generating and Approximating Model . . . 118
7.3 Likelihood and Identification Problem . . . 119
7.3.1 Estimation Procedure . . . 121
7.3.2 Solution Path . . . 126
7.4 Simulation Studies . . . 128
7.5 Data Examples . . . 133
7.5.1 Sao Paulo Air Pollution Data Set . . . 133
7.5.2 Bodyfat Data Set . . . 134
7.6 Conclusion and Remarks . . . 138

Appendix 139

Bibliography 143

Chapter 1

Introduction and Remarks

To meet the different contents of this thesis, it is divided into two parts. The first part is about generalized linear models (GLMs) which is proposed by Nelder and Wedderburn (1972). The second part is based on single index models (SIMs). Härdle et al. (1993) or Cui et al. (2009) define SIMs as GLMs where the response function is twice differentiable. In contrast to this we consider a SIM as a GLM with twice differentiable and monotonically increasing response function. We prefer the monotonicity constraint because of better interpretability. Additionally, by monotonicity SIMs are quite close to classical GLMs. However, this thesis is about regularized estimators for SIMs and GLMs. The focus is on two regularization techniques, namely penalization and boosting. In this thesis we consider regularization procedures which induce variable selection. Basis of all regression problems in thesis is the log-likelihood function $l(\boldsymbol{\theta})$ which has to be maximized, where $\boldsymbol{\theta}$ is a parameter vector the log-likelihood depends on. Alternatively $-l(\boldsymbol{\theta})$ has to be minimized.

Penalized log-likelihood problems have the form

$$\widehat{\boldsymbol{\theta}} = \text{argmin}_{\boldsymbol{\theta}} \left\{ -l(\boldsymbol{\theta}) + \left[\sum_{k=1}^{l} \lambda_k P_k(\boldsymbol{\theta}) \right] \right\}$$

where $\lambda_k > 0$ is a weight for the kth penalty term $P_k(\boldsymbol{\theta}) > 0$, $k = 1, ..., l$, each depending on $\boldsymbol{\theta}$. So there is a tradeoff between the sum penalty term $\left[\sum_{k=1}^{l} \lambda_k P_k(\boldsymbol{\theta})\right]$ and the negative log-likelihood function. On the one hand the log-likelihood must be maximized, on the other hand the sum of penalty terms must become small. In the case of penalized log-likelihood problems the regularization is part of the optimization problem. There are many advantages and motivations for the penalization of likelihood functions which are discussed in the introductions of the first three chapters.

Boosting techniques are based on weak learning proposed by Shapire (1990) in context of classification. Freund and Shapire (1997) establish the first boosting algorithm. In the following years boosting techniques are widely developed for solving wide classes of regression models (see Bühlmann and Yu, 2003; Tutz and Binder, 2006; Bühlmann and Hothorn, 2007; Hothorn et al., 2010). Especially componentwise boosting became very popular. In this thesis boosting techniques are used for maximizing the log-likelihood

function. In componentwise boosting the log-likelihood is maximized by small steps. In each step only one covariate is updated. This means that not all covariates are estimated but only one is estimated "a little bit in the right direction". By early stopping the model becomes sparse, if not all covariates are updated when the algorithm is stopped. In contrast to penalization the regularization of componentwise boosting techniques is part of the algorithm. Componentwise boosting regularizes by weak learning only one component in each boosting iteration.

Guideline Through the Thesis

The first part of this thesis is about penalization of GLMs. It focuses on penalty terms that induce

- variable clustering: highly correlated covariates are estimated equal apart from sign
- variable selection: low influential covariates are shrunken to zero.

All presented penalty terms are polytopes. Two of them are correlation driven. In chapter 2 we present a general framework for polytopes as penalty regions in regression and present the V8 procedures which performs variable selection and variable clustering. Hereby clustering is controlled by correlation. In chapter 3 we present a further correlation driven penalty region with variable selection and clustering property. We call it the pairwise fused lasso and it is inspired by fused lasso from Tibshirani et al. (2005). In contrast to the fused lasso the new procedure does not depend on ordered covariates. The last chapter of the first part is the generalization of the OSCAR penalty from Bondell and Reich (2008) to GLMs.

The second part of this thesis is about SIMs (with monotonically increasing response function). The first chapter of this part (chapter 5) presents a boosting technique for estimating sparse SIMs. It is a generalization of the procedures proposed by Tutz and Leitenstorfer (2011). In contrast to Tutz and Leitenstorfer (2011) the new procedure (FlexLink) is not only for normal distributed response and includes variable selection. The following chapter 6 generalizes the FlexLink to additive predictors. In chapter 7 a regularization of SIMs by penalty terms is given. The response function and the parameter vector are penalized by an appropriate term and the corresponding penalized log-likelihood problem is solved.

Software and Publications

Apart from chapter 2 all computations are carried out using the statistical software `R` (see R Development Core Team, 2010) and the related packages which are indicated in the respective chapters and sections. In chapter 2 we use `MATLAB` from Mathworks (see

`http://www.mathworks.de/products/matlab/index.html`) for the computation and illustration of the simulations study and the data example. The graphical illustration of polytopes in chapter 2 and 4 are carried out by `polymake` (see `http://polymake.org/doku.php` and Gawrilow and Joswig (2000)).

Parts of this thesis are published as technical reports or as articles in journals. These articles and technical reports are done in cooperation with coauthors. Part I of this thesis is based on

- **Petry S.** and G. Tutz (2011). Shrinkage and variable selection by polytopes. *Journal of Statistical Planning and Inference (to appear)*. (chapter 2)
- **Petry S.**, C. Flexeder and G. Tutz (2010). Pairwise fused lasso. Technical Report 102, Department of Statistics LMU Munich. (chapter 3)
- **Petry S.** and G. Tutz (2011). The Oscar for generalized linear models. Technical Report 112, Department of Statistics LMU Munich. (chapter 4)

and part II is based on

- Tutz G. and **S. Petry** (2011). Nonparametric estimation of the link function including variable selection. *Statistics and Computing (to appear)*, 21. (chapter 5)

In general each chapter can be seen as an autonomous unit.

Part I

Regularization Approaches for Generalized Linear Models

Chapter 2

Shrinkage and Variable Selection by Polytopes

Constrained estimators that enforce variable selection and grouping of highly correlated data have been shown to be successful in finding sparse representations and obtaining good performance in prediction. We consider polytopes as a general class of compact and convex constraint regions. Well established procedures like LASSO (Tibshirani, 1996) or OSCAR (Bondell and Reich, 2008) are shown to be based on specific subclasses of polytopes. The general framework of polytopes can be used to investigate the geometric structure that underlies these procedures. Moreover, we propose a specifically designed class of polytopes that enforces variable selection and grouping. Simulation studies and an application illustrate the usefulness of the proposed method.

2.1 Introduction

We consider the linear normal regression model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},\ \boldsymbol{\epsilon} \sim N_n(\boldsymbol{0},\, \sigma^2 \boldsymbol{I}),$$

where the response $\boldsymbol{y} = (y_1,\, ...,\, y_n)^T$ and the design $\boldsymbol{X} = (\boldsymbol{x}_1|...|\boldsymbol{x}_p)$ are based on n iid observations. Since the methods considered are not equivariant we will use standardized data.Therefore, $\boldsymbol{y} = (y_1,\, ...,\, y_n)^T$ is the centered response and $\boldsymbol{x}_j = (x_{1j},\, ...,\, x_{nj})^T$ the j-th standardized predictor, $j \in \{1,\, ...,\, p\}$, so that

$$\sum_{i=1}^{n} y_i = 0,\ \sum_{i=1}^{n} x_{ij} = 0,\ \sum_{i=1}^{n} x_{ij}^2 = 1,\ \forall\, j \in \{1,\, ...,\, p\},$$

holds.

In normal distribution regression problems one typically uses the *ordinary least squares estimator* $\widehat{\boldsymbol{\beta}}_{OLS}$. The underlying loss function is the *quadratic loss* or *sum of squares*

$$Q(\boldsymbol{\beta}|\boldsymbol{y},\, \boldsymbol{X}) := \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2$$

and $\widehat{\boldsymbol{\beta}}_{OLS}$ minimizes the unconstrained regression problem

$$\widehat{\boldsymbol{\beta}}_{OLS} = \text{argmin}_{\boldsymbol{\beta}} \left\{ Q(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X}) \right\}.$$

When c is appropriately chosen the contours of the quadratic loss

$$S_c(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X}) = \{\boldsymbol{\beta} \in \mathbb{R}^p \, : \, Q(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X}) \leq c\}$$

form hyperellipsoids centered at $\widehat{\boldsymbol{\beta}}_{OLS}$. Moreover, $Q(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X})$ is upper semicontinuous and strictly convex, which are properties that guarantee an unique solution of constrained estimates.

Constraining the domain of $\boldsymbol{\beta}$ can be motivated by non-sample information given by some scientific theory. For example in economical input-output-systems it is assumed that the inputs have a positive influence on the output. Then the domain of the estimate is restricted by $\beta_{input} > 0$. More general, there is a mathematical motivation to constrain the parameter domain of a regression problem. James and Stein (1961) proposed the first *shrinkage estimator* which became known in the literature as James-Stein-estimator. The expression "shrinkage" is due to the geometrical interpretation of Hoerl and Kennard (1970). Hoerl and Kennard (1970) described that the length of the OLS-vector $|\widehat{\boldsymbol{\beta}}_{OLS}|$ tends to be longer than the length of the true parameter vector $|\boldsymbol{\beta}_{true}|$. This effect can be overcome by restricting the parameter domain to a centrosymmetric region around the origin of the parameter space.

Hoerl and Kennard (1970) used centered p-dimensional spheres with radius t which yields *ridge regression*. Centrosymmetric regions around the origin are a general concept to compensate for the "$|\boldsymbol{\beta}_{true}| < |\widehat{\boldsymbol{\beta}}_{OLS}|$-effect" since the properties of the loss function $Q(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X})$ together with compactness and convexity of the domain guarantee existence and uniqueness of the solution. In the following we will call regions with the three properties convexity, compactness, and centrosymmetry *penalty regions*.

The term penalty region is commonly used when the problem is represented in its penalized form. For some constrained regression problems there exist alternative formulations which have equivalent solutions. For example, the *constrained version* of the ridge estimator is

$$\widehat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \left\{ \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 \, , \, s.t. \, \sum_{j=1}^{p} \beta_j^2 \leq t, \, t \geq 0 \right\}. \tag{2.1}$$

For fixed t the corresponding *penalized regression problem* has the form

$$\widehat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \left\{ \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \lambda \sum_{j=1}^{p} \beta_j^2, \, \lambda \geq 0 \right\}. \tag{2.2}$$

The proof of the equivalence is based on the theory of Lagrangian multipliers and can be found in Luenberger (1969) where the equivalence for a set of constraints is shown by using a vector $\boldsymbol{\lambda}^T \in \mathbb{R}^p$. It should be noted, that not every constrained regression problem can be given as a penalized regression problem.

It is intuitively clear that a penalty region determines the properties of the estimate beyond of tackling the "$|\boldsymbol{\beta}_{true}| < |\widehat{\boldsymbol{\beta}}_{OLS}|$-problem". Therefore the penalty regions should be carefully designed. We will focus on two properties of estimates:

Variable selection: Coefficients whose corresponding predictors have vanishing or low influence on the response should be shrunk to zero.

Grouping : For a group of highly correlated variables it can be advantageous that estimated coefficients differ not too strongly.

A well-established shrinkage procedure that includes variable selection is the LASSO (Tibshirani, 1996). One criticism of the LASSO, which has been pointed out by Zou and Hastie (2005), is the behavior when predictors are highly correlated. In that case the LASSO tends to select only one or two from the group of the correlated influential predictors. Therefore, Zou and Hastie (2005) proposed the *Elastic Net (EN)* which tends to include the whole group of highly correlated predictors. The EN enforces the grouping effect as stated in Theorem 1 of Zou and Hastie (2005) where a relation between sample correlation and grouping was given. The EN does not use the sample correlation explicitly, the grouping effect is achieved by a second penalty term together with a second tuning parameter which do not depend on the sample correlation. In a similar way Bondell and Reich (2008) introduced the OSCAR by including an alternative penalty term that enforces grouping. OSCAR also selects variables and shows the grouping effect. Also a relation between sample correlation and grouping may be derived. An alternative penalty that explicitly uses the correlation and enforces the grouping property was proposed by Tutz and Ulbricht (2009) under the name correlation-based penalty. Variable selection was obtained by combining boosting techniques with the correlation based penalty.

We will consider established procedures within the general framework of constraint regions based on polytopes and introduce a correlation-based penalty region called V8, which groups and selects variables. In Section 2.2 we give some basic concepts of polytope theory. Based on these concepts the LASSO is discussed in Section 2.2.2 and OSCAR in Section 2.2.3. The embedding into the framework of polytopes allows to derive some new results for these procedures. In Section 2.3 we introduce the V8 procedure and give algorithms that solve the constrained least squares problem. In Section 2.4 the V8 procedure is compared to established procedures on the basis of simulations.

2.2 Polytopes as Constraint Region

Polytopes provide a simple class of compact and convex regions that are useful as constraint regions. They were implicitly used in established regression procedures like LASSO (Tibshirani, 1996) or OSCAR (Bondell and Reich, 2008). In general, polytopal constrained regression problems can be reformulated as linear constrained regression problems (cf. Theorem 1). But in practice it can be hard to reformulate the polytopal constrained regression

problem as a linear constrained problem. One objective of this article is to use geometrical arguments for analyzing and designing polytopal penalty regions. In the following the geometric background and the mathematical foundation of polytopes is shortly sketched.

2.2.1 Some Concepts in Polytope Theory

Let in general $\boldsymbol{a} \leq \boldsymbol{b}$ denote that $a_r \leq b_r$ for all components of $\boldsymbol{a}$, $\boldsymbol{b} \in \mathbb{R}^m$. In the following *hyperplanes* and corresponding *halfspaces* play an important role. Definitions are given in the Appendix (see Definition A 1).

Polytopes are a class of fundamental geometric objects defined in $\mathbb{R}^p$. The dimension of a polytope is the dimension of its affine hull and a p-dimensional polytope is called *p-polytope*. There are two ways to describe polytopes: *V-polytopes* and *H-polytopes*.

Definition 1 (V-Polytope) *A* V-Polytope *is the convex hull of a finite point set $\mathcal{V} \subset \mathbb{R}^p$:*

$$P(\mathcal{V}) := conv(\mathcal{V}).$$

Definition 2 (H-Polytope) *A subset $P \subset \mathbb{R}^p$ is called an* H-Polytope *if it is the bounded intersection of a finite number of closed lower linear halfspaces. For $\boldsymbol{A} \in \mathbb{R}^{m \times p}$, $\boldsymbol{t} \in \mathbb{R}^m$*

$$P(\boldsymbol{A}, \boldsymbol{t}) := \{\boldsymbol{x} \in \mathbb{R}^p : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{t}\}$$

describes an H-Polytope if $P(\boldsymbol{A}, \boldsymbol{t})$ is bounded.

The intuitive question is whether there exists a relation between H-polytopes and V-polytopes. The answer is given in Ziegler (1994) where the following theorem is shown to hold.

Theorem 1 (Main Theorem) *A subset $P \subseteq \mathbb{R}^p$ is the convex hull of a finite point set (a V-Polytope)*

$$P = conv(\mathcal{V}), \qquad \text{for some } \mathcal{V} \subset \mathbb{R}^{p \times n}$$

if and only if it is a bounded intersection of closed (lower linear) halfspaces (an H-Polytope)

$$P = P(\boldsymbol{A}, \boldsymbol{t}) = \{\boldsymbol{x} \in \mathbb{R}^p : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{t}\}, \quad \text{for some } \boldsymbol{A} \in \mathbb{R}^{m \times p}, \boldsymbol{t} \in \mathbb{R}^m.$$

However, the transformation from H- to V-representation and vice versa can be computationally expensive. The number of producing halfspaces and of vertices is an indicator for the computational costs.

Each row of the system of inequalities $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{t}$ describes a linear lower closed halfspace. It represents the normal vector of a hyperplane generating a corresponding halfspace. A *vertex* of a p-polytope P is an element $\boldsymbol{v} \in P$ which can not be given as a convex combination of the remaining elements $P \setminus \{\boldsymbol{v}\}$ (see Figure 2.1 where the five vertices are easily identified). Although in Definition 1 a general finite set $\mathcal{V}$ is used to describe P it is sufficient to use only the vertices of P to define the same polytope P. In other words,

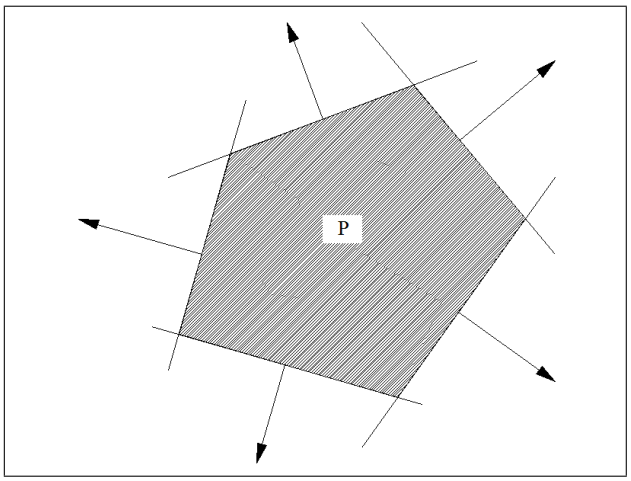

FIGURE 2.1: *Illustration of the H- and V-representation of the Polytope P. The closed lower halfspaces are on the opposite side of the illustrated normal vectors. The intersection of these halfspaces is P which is shown by the shaded area. The graphed intersection of the hyperplanes are the five vertices of P. The convex hull of the five vertices produces the same polytope P.*

let $P = conv(\mathcal{V})$ be a V-polytope and $E(\mathcal{V}) \subseteq \mathcal{V}$ be the set of all vertices of P then $P = conv(\mathcal{V}) = conv(E(\mathcal{V}))$ holds. We assume $\mathcal{V} = E(\mathcal{V})$ in the following. It is obvious that every point $\boldsymbol{x}$ of a polytope $P = conv(\mathcal{V})$ can be presented as the convex combination of all vertices,

$$\boldsymbol{x} = \sum_{i \in I} \lambda_i \boldsymbol{v}_i,\ \lambda_i \geq 0, \sum_{i \in I} \lambda_i = 1,\ \boldsymbol{v}_i \in \mathcal{V}, \tag{2.3}$$

where I is the index set of all vertices. In addition, we only consider H-polytopes whose description is not redundant. This means the leaving out of any row of $\boldsymbol{Ax} \leq \boldsymbol{t}$ will change the polytope.

Alternatively, a polytope can be described by its faces. The definition of faces of a polytope are based on *supporting hyperplanes* or shortly *supports* (for a definition see Def.2 in the Appendix).

Definition 3 (Faces of a Polytope) *Let $P \subset \mathbb{R}^p$ be a p-polytope and $H \subset \mathbb{R}^p$ be a support of P. Then the intersection $P \cap H$ is called* face of P. *A k-dimensional face is called* k-face. *A* 0*-face is a vertex, an* 1*-face is an edge, and a $(p-1)$-face is a facet.*

An important feature is that every face is a convex hull of vertices but not every convex hull of vertices is a face. Hence, not every convex combination of vertices lies on the surface of the polytope, but every facet is the convex hull of $q \geq p$ vertices (see Ziegler, 1994). The linear hull of these q vertices is the intersecting support which produces this facet. The intersecting support is given by one row of $\boldsymbol{Ax} \leq \boldsymbol{t}$ in Definition 2. A p-polytope P is called *simplicial* iff every facet of P contains the minimal number of p vertices.

The special class of polytopes which is of interest here is the following.

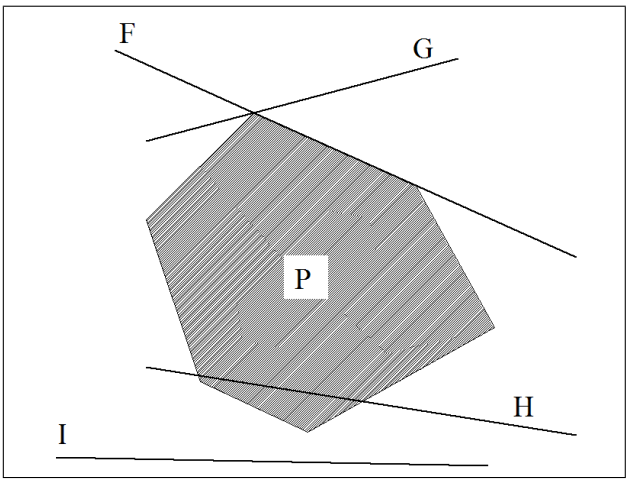

FIGURE 2.2: *Illustration of Definition 3. Four hyperplanes F, G, H, I and their relationship to P: I is not a support because $I \cap P = \emptyset$. H is not a support because P is not entirely contained in one of the both closed halfspaces H^+ or H^-. F and G are supports. $P \cap G$ is a vertex of P. $P \cap F$ is a facet of P. (In $\mathbb{R}^2$ an edge is also a hyperplane.)*

Definition 4 *A p-polytope P is called* centrosymmetric*, if*

1. *the origin is an inner point of P: $\mathbf{0} \in P$.*
2. *If $\boldsymbol{v} \in P$ then $-1 \cdot \boldsymbol{v} \in P$.*

It is intuitively clear that a centrosymmetric p-polytope can be scaled up or down in two ways

1. Multiplying the right hand side of $\boldsymbol{Ax} \leq \boldsymbol{t}$ with $s > 0$.
2. Multiplying all vertices with $s > 0$.

2.2.2 LASSO

The famous LASSO, proposed by Tibshirani (1996), is very popular because of its variable selection property and has been used in many fields of statistics. The LASSO constraint region is given by

$$\sum_{j=1}^{p} |\beta_j| \leq t,\ t > 0, \tag{2.4}$$

which corresponds to a p-polytope. The H-representation of the constraint region is obtained by rewriting the absolute value function $|.|$ in (2.4). The result is a system of inequations

$$\boldsymbol{L\beta} \leq \boldsymbol{t}, \tag{2.5}$$

where $\boldsymbol{L}$ is a $(2^p \times p)$-matrix. Each row of $\boldsymbol{L}$ is one of the 2^p variations of entries -1 or $+1$ and $\boldsymbol{t}$ is a 2^p-dimensional vector whose entries are equal to $t > 0$. An example for the case $p = 3$ can be found in the Appendix (Example A 1). More concise, the LASSO constraint region is a p-crosspolytope, which is scaled up or down by the tuning parameter $t > 0$. (For the definition of a p-crosspolytope see Ziegler (1994), p. 8.). The underlying polytope is simplicial and this property is maintained by scaling up or down. An illustration in $\mathbb{R}^3$ is given in Figure 2.3.

The vertices of the LASSO penalty region are

$$\mathcal{L} = \{t \cdot \boldsymbol{e}_1, -t \cdot \boldsymbol{e}_1, ..., t \cdot \boldsymbol{e}_p, -t \cdot \boldsymbol{e}_p, t > 0\}, \tag{2.6}$$

where $\boldsymbol{e}_j$, $j = 1, ..., p$, denotes the j-th unit vector of $\mathbb{R}^p$. Therefore the V-representation of the LASSO penalty region is $P = conv(\mathcal{L})$.

Since the constraint (2.4) is determined by the 2^p constraints specified in the rows of (2.5), it is easy to transform the LASSO problem in constrained form,

$$\widehat{\boldsymbol{\beta}}_L = \text{argmin}_{\boldsymbol{\beta}} \left\{ \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2, \, s.t. \, \sum_{j=1}^{p} |\beta_j| \leq t, \, t \geq 0 \right\},$$

into a penalized regression problem,

$$\widehat{\boldsymbol{\beta}}_L = \text{argmin}_{\boldsymbol{\beta}} \left\{ \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \lambda \sum_{j=1}^{p} |\beta_j|, \, \lambda \geq 0 \right\}.$$

If the OLS estimate exists and $\sum_{j=1}^{p} |\beta_{OLS_j}| = t_0$ then $\widehat{\boldsymbol{\beta}}_L$ is the contact point of the contour of the loss function $S_c(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X})$ and the penalty region $\sum_{j=1}^{p} |\beta_j| \leq t$, $0 < t < t_0$. The variable selection property of the LASSO can be illustrated by using the V-representation. Although not all convex combinations of vertices are on the surface the solution of a polytopal constrained regression problem lies on the surface. So with respect to the simpliciality of the LASSO penalty region variable selection is performed if the solution is a convex combination of less than p vertices of its penalty region, i.e. at least one of the λ_is in (2.3) is zero. Thus, in $\mathbb{R}^3$ one can distinguish three cases of LASSO solutions:

1. If the LASSO solution lies on a vertex only one coefficient is nonzero, i.e. only one λ_i in (2.3) is 1.

2. If the LASSO solution lies on an edge that connects two axes, two λ_is in (2.3) are non-zero.

3. If the LASSO solution lies on a facet, three λ_is in (2.3) are non-zero.

In the first two cases variables are selected.

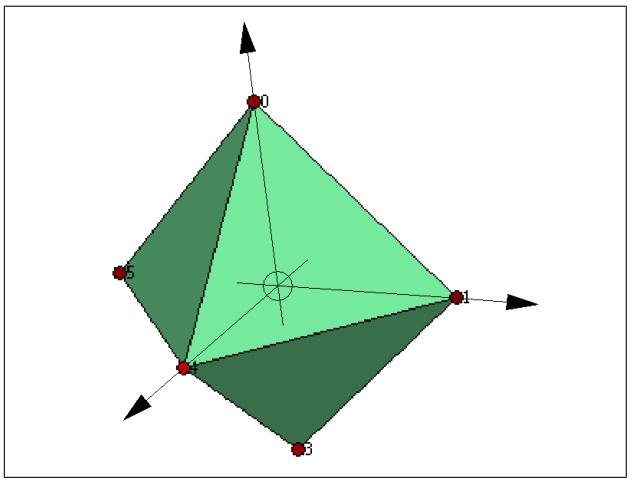

FIGURE 2.3: *LASSO constraint region in* $\mathbb{R}^3$.

2.2.3 OSCAR

Bondell and Reich (2008) proposed a shrinkage methods called OSCAR, which stands for **O**ctagonal **S**hrinkage and **C**lustering **A**lgorithm for **R**egression. Its constraint region is

$$\sum_{j=1}^{p} \left[|\beta_j| + c \cdot \sum_{j<k} \max \{|\beta_j|, |\beta_k|\} \right] \leq t. \tag{2.7}$$

Bondell and Reich (2008) also give an alternative representation of their penalty region. Let $|\beta|_{(k)}$ denote the absolute value of the component of $\boldsymbol{\beta} \in \mathbb{R}^p$ whose rank is k so that $|\beta|_{(1)} \leq |\beta|_{(2)} \leq \ldots \leq |\beta|_{(p)}$ holds. With $|\beta|_{(.)}$ the OSCAR penalty region (2.7) is equivalent to

$$\sum_{j=1}^{p} [c(j-1)+1] \cdot |\beta|_{(j)} \leq t. \tag{2.8}$$

First we discuss the penalty region in the implicitly given H-representation. Then we derive the vertices as a new result. That is helpful because the V-representation allows an alternative perspective on the grouping property of OSCAR.

The analysis of the OSCAR penalty region in H-representation is based on segmentation of the p-dimensional parameter space $\mathbb{R}^p$. First we partition $\mathbb{R}^p$ in the 2^p *orthants*, which are regions for which the signs of components are fixed. Second we segment every orthant in $p!$ regions which are defined by a fixed order of ranks of $|\beta_j|$, $j = 1, ..., p$. Figure 2.4 illustrates the segmentation for one orthant in $\mathbb{R}^3$.

The absolute value function $|.|$ in the OSCAR penalty term corresponds to the orthants and the segmentation of each orthant is given by the sum of pairwise maximum norms. It is seen from (2.8) that the OSCAR penalty region is an H-polytope which depends on the

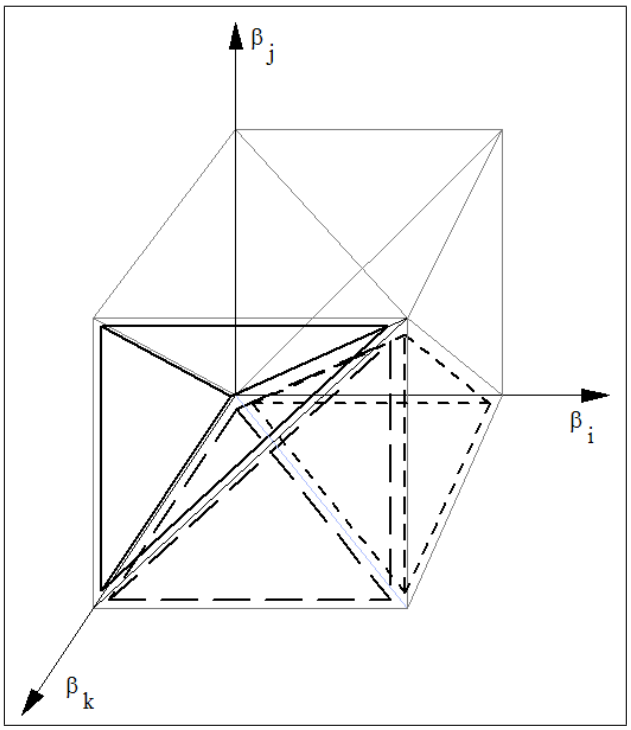

FIGURE 2.4: *The region described by the shortly dashed lines corresponds to the ordering* $|\beta_j| \leq |\beta_k| \leq |\beta_i|$, *long dashed lines correspond to the ordering* $|\beta_j| \leq |\beta_i| \leq |\beta_k|$ *holds and solid lines correspond to the ordering* $|\beta_i| \leq |\beta_j| \leq |\beta_k|$.

order of ranks of $|\beta_j|$ and on the sign constellation with the order of ranks being linked to the weights $[c(j-1)+1]$.

For the derivation of the penalty region, $P(\boldsymbol{A}(c), \boldsymbol{t})$, we consider first the orthant with only positive signs. For this orthant we create a $(p! \times p)$-matrix $\widetilde{\boldsymbol{A}}(c)$ where every row represents one of the $p!$ permutation of the p weights $[c(j-1)+1]$, $j = 1, ..., p$. In a second step we form $(2^p - 1)$ matrices $\widetilde{\boldsymbol{A}}(c)$, which are constructed by changing the sign in one column of $\widetilde{\boldsymbol{A}}(c)$. Finally we combine these matrices obtaining the $(2^p \cdot p!) \times p$-matrix $\boldsymbol{A}(c)$. The matrices built in the second step correspond to the orthants. Example A 2 in the Appendix shows the H-representation of an OSCAR penalty region.

Therefore the OSCAR penalty region with the tuning parameters $t > 0$ and $c > 0$ is represented by the intersection of $2^p \cdot p!$ hyperplanes, which shows the high complexity of the OSCAR penalty region. It is remarkable that the $2^p \cdot p!$ constraints sum up to one constraint given in (2.7).

On OSCAR's Vertices

Hitherto the OSCAR penalty region is considered only as a H-polytope. The Main Theorem (Theorem 1) suggests to consider the OSCAR penalty region as a V-polytope. The vertices of the OSCAR penalty region have a simple structure which is given in the following

proposition.

Proposition 1 *Let an p-dimensional OSCAR penalty region with the tuning parameters $t > 0$ and $c > 0$ be given. Then the set of vertices of the OSCAR penalty region is the set of points with the following properties:*

1. *From the p components $1 \leq m \leq p$ components are nonzero and the absolute value of these components is equal. The remaining $p - m$ components are zero.*

2. *The $1 \leq m \leq p$ nonzero components of a vertex have the absolute value*

$$v(m) := \frac{t}{\sum_{j=p+1-m}^{p}[c(j-1)+1]}. \tag{2.9}$$

For the *proof* see Appendix (Proof A 1).

Corollary 1 *Under the conditions of Proposition 1 one obtains:*

1. *The OSCAR penalty region is the convex hull of $3^p - 1$ vertices,*

2. *The OSCAR penalty region is simplicial.*

For the *proof* see Appendix (Proof A 2). It is remarkable that (2.9) depends not only on the penalty level t and the tuning parameter c but also on the dimension of the problem p.

Figure 2.5 shows the OSCAR penalty region for different tuning parameters. For fixed tuning parameter t and p (2.9) becomes smaller by increasing c. So for graphical illustration we adjust t so that the axis intercepts are equal. The first row of Figure 2.5 explains the naming of OSCAR. It illustrates that orthogonal projections of an OSCAR penalty region on any β_i-β_j-plane form an octagon, which may be shown by using orthogonal projections of the vertices on any β_i-β_j-plane. Because of symmetry, in Figure 2.5 only one projection is shown. For further illustration the set of all vertices of an OSCAR penalty region in the case $p = 3$ are given in the Appendix (Example A 3).

In general, the parameter c controls the form of the OSCAR penalty region. For $c \to 0$ it converges to the LASSO penalty region. This can be shown by considering the limit $c \to 0$ within the system of inequations. It is noteworthy that for $c \to \infty$ and $p > 2$ the OSCAR penalty region does not converge to a p-dimensional cube (p-cube), which would enforce extreme grouping but no variable selection. A p-cube would make sense only if all predictors were very highly correlated. Rather for $c \to \infty$ the OSCAR converges to a specific polytope. This can be derived by considering the following limit: $\lim_{c\to\infty} v(m_1)/v(m_2) = (\sum_{j=p+1-m_2}^{p}(j-1))/(\sum_{j=p+1-m_1}^{p}(j-1))$, where $v(.)$ is given by (2.9). In the limit the ratio $v(m_1)/v(m_2)$ depends only on m_1 and m_2, the different numbers of nonzero components of vertices. Hence for $c \to \infty$ the form of the OSCAR polytope is fixed but does not converge to a p-cube.

Bondell and Reich (2008) describe the grouping (or clustering) property of OSCAR by giving a relation between correlation and grouping. Another perspective on the properties

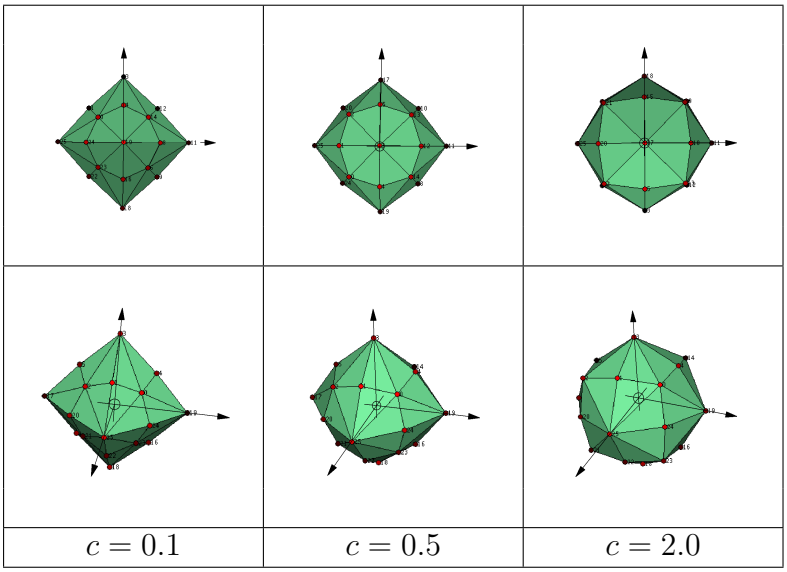

FIGURE 2.5: *The OSCAR penalty region with three different tuning parameter c. In the first row the projections in to a β_i-β_j-plane is shown. In the second row a oblique view is shown.*

variable selection and grouping is obtained by considering vertices. From Figure 2.5 it is seen that grouping of three variables is forced by the vertices in the middle of the orthants. In general, for grouping of more than two predictors vertices with more than two nonzero components seem to be necessary. Grouping or variable selection is performed if less than p vertices take part in the convex combination of the OSCAR solution. Bondell and Reich (2008) give an upper bound criterion for the relationship between the tuning parameter c and the correlation of predictors but they do not use correlation directly for generating the penalty.

2.3 The V8 procedure

In the following a correlation driven polytope is proposed, which uses the correlation within data to define the penalty region.

2.3.1 The V8-polytope

The V8-polytope is called V8 because it is a **V**-polytope for which projections on any β_i-β_j-plane are octagons. The construction focuses on the grouping property, which was advocated in particular by Zou and Hastie (2005) and is behind OSCAR (Bondell and Reich, 2008) and correlation-based penalties (Tutz and Ulbricht, 2009). The grouping of highly correlated variables is enforced by shrinking the corresponding coefficients toward a common coefficient. From a geometrical point of view this means: if two variables $\boldsymbol{x}_i$

and $\boldsymbol{x}_j$ are highly correlated the estimated coefficients should lie on the face of a polytopal penalty region where $|\beta_i| = |\beta_j|$ holds. This suggests to design correlation driven polytopes where the correlation between predictors determines the form of the polytope. Although highly correlated covariates do not necessarily have the same influence on the response, in the case of high correlation, selection procedures like the LASSO tend to pick out only one of the highly correlated covariates. When introducing the elastic net Zou and Hastie had explicitly considered an example where the explanatory variables were just noisy versions of the same variable. In cases like this the same parameter seems warranted. Moreover, also in correlation based procedures the data still determine the coefficient, the penalty just enforces the effect of identical parameters. Only for perfect correlation the correlations are strictly set equal.

The V8-polytope should feature the following properties:

(P1) The orthogonal projection of the polytope on every β_i-β_j-plane, $1 \leq i \leq j \leq p$, is a (convex) octagon.

(P2) The octagons are centrosymmetric.

(P3) Four vertices of each octagon lie on the axis at the values $\pm t$, two on the β_i-axis and two on the β_j-axis.

(P4) The four remaining vertices are on the bisecting line of the β_i-β_j-plane where $|\beta_i| = |\beta_j|$.

The OSCAR penalty region shares all of these properties, which may be shown by projecting the vertices of the OSCAR penalty region on any β_i-β_j-plane. For the V8-polytope in addition the penalty region is supposed to depend on the estimated correlation between two predictors, $\rho_{ij} := corr(\boldsymbol{x}_i, \boldsymbol{x}_j)$ by use of a function $c : [-1, 1] \mapsto [0, 1]$. In general, every function $c(\rho_{ij})$ with the following properties is appropriate:

(1) $c(0) = 1$.

(2) $c(1) = c(-1) = 0$.

(3) $c(\rho_{ij}) = c(-\rho_{ij})$.

(4) $c(.)$ is increasing in $[-1, 0]$ and decreasing in $(0, 1]$.

In the following we use

$$c(\rho_{ij}) := 1 - |\rho_{ij}|^k, k \geq 1. \tag{2.10}$$

The vertices described by (P3) are defined as the same vertices as for the LASSO $\mathcal{L}$ and do not depend on the correlation. The vertices characterized by (P4) for any β_i-β_j-plane, $1 \leq i \leq j \leq p$, are specified by

$$\mathcal{B}_{ij} = \left\{ \boldsymbol{b} \in \mathbb{R}^p : |b_i| = \frac{t}{1 + c(\rho_{ij})}, |b_j| = \frac{t}{1 + c(\rho_{ij})}, b_k = 0, k \neq i, j \right\}.$$

It is obvious that $|\mathcal{B}_{ij}| = 4$. The assumptions (1)–(4) of the function $c(.)$ induce the following properties on $\mathcal{B}_{ij}$. If $\rho_{ij} \to 0$ the elements of $\mathcal{B}_{ij}$ become redundant because they are convex combinations of $\mathcal{L}$. The projection on any β_i-β_j-plane converges to a diamond with side length $\sqrt{2}t$ and so variable selection is enforced. For $|\rho_{ij}| \to 1$ the four elements $\{+t\boldsymbol{e}_i, -t\boldsymbol{e}_i, +t\boldsymbol{e}_j, -t\boldsymbol{e}_j\} \subset \mathcal{L}$ become redundant because they are convex combinations of $\mathcal{B}_{ij}$. In this case the octagon converges to a square with side length $2t$ and grouping of the variables $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is enforced. For $k = 1$ in (2.10) this behavior is illustrated in the first row of Figure 2.6. With $\mathcal{B} = \bigcup_{i<j} \mathcal{B}_{ij}$ the vertices of the V8 penalty region are $\mathcal{V} = \mathcal{L} \cup \mathcal{B}$. There are $\binom{p}{2}$ different sets $\mathcal{B}_{ij}$, and so $|\mathcal{V}| = 2p + 4 \cdot \binom{p}{2} = 2p^2$. An example for the case $p = 4$ is given in the Appendix (Example A 4). It is obvious that $\mathcal{V}$ is convex and that for $\rho_{ij} = 0, \forall i \neq j$, the V8 penalty region is the same as for LASSO. Figure 2.7 illustrates the V8 penalty region for correlation structure given by $\rho_{12} = 0.2, \rho_{13} = 0.5, \rho_{23} = 0.8$. Here we choose $k = 1$ in (2.10) again.

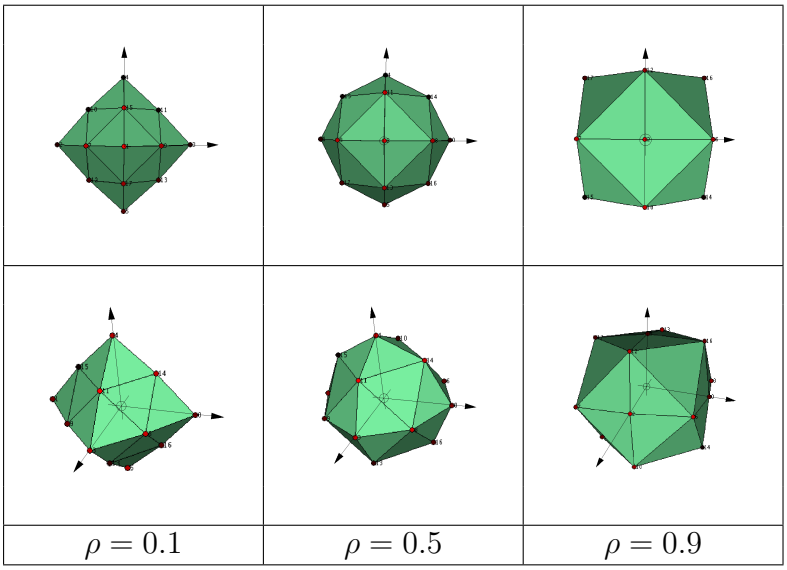

FIGURE 2.6: *V8-polytopes with unique correlation ρ_{ij} between all pairs i-j where $k = 1$ in (2.10).*

In summary, the V8 constraint region enforces variable selection through the LASSO vertices and enforces grouping through the vertices that are added by use of the correlation between two variables.

2.3.2 Solving Polytopal Constrained Regression Problems

In general, a polytopal constrained regression problem can be formulated as follows:

$$\widehat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \left\{ \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 , \text{ s.t. } \boldsymbol{\beta} \in P \right\}, \tag{2.11}$$

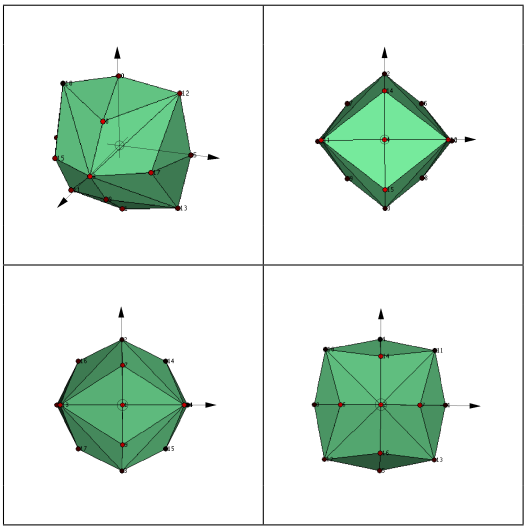

FIGURE 2.7: Top left: *An oblique view of the V8 penalty region.* Top right: *Orthogonal projection on the β_1-β_2-plane where $\rho_{12} = 0.2$.* Bottom left: *Orthogonal projection on the β_1-β_3-plane where $\rho_{13} = 0.5$.* Bottom right: *The orthogonal projection on the β_2-β_3-plane where $\rho_{23} = 0.8$. $k = 1$ is used in (2.10).*

where P is a polytope. Based on the Main Theorem (see Theorem 1) there are two different ways to formulate (2.11). If P is an H-polytopes then (2.11) has the form

$$\widehat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \left\{ \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2, \text{ s.t. } \boldsymbol{A}\boldsymbol{\beta} \leq \boldsymbol{t} \right\}. \tag{2.12}$$

This is a linearly constrained regression problem with the quadratic loss function which can be solved with established tools like `lsqlin` routine in `MATLAB`.

When P from (2.11) is a V-polytope let $\mathcal{V} = \{\boldsymbol{v}_1, ..., \boldsymbol{v}_{n_V}\}$ denote the set of vertices of P and $I := \{1, ..., n_V\}$ is the index set of $\mathcal{V}$. Every point $\boldsymbol{\beta} \in P$ is a convex combination of elements of $\mathcal{V}$. The convex combination can be written in matrix notation

$$\boldsymbol{\beta} = \boldsymbol{V} \cdot \boldsymbol{\lambda} \text{ with } \boldsymbol{V} = (\boldsymbol{v}_1 | \cdots | \boldsymbol{v}_{n_V}) \text{ and } \boldsymbol{\lambda} = (\lambda_1, ..., \lambda_{n_V})^T \tag{2.13}$$

with $\lambda_i \geq 0, \sum_{i \in I} \lambda_i = 1$, $\boldsymbol{v}_i \in \mathcal{V}$. So (2.11) turns into a quadratic optimization problem in $\boldsymbol{\lambda}$,

$$\widehat{\boldsymbol{\lambda}} = \text{argmin}_{\boldsymbol{\lambda}} \left\{ \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{V} \cdot \boldsymbol{\lambda}\|^2, \text{ s.t. } \lambda_i \geq 0, \sum_{i \in I} \lambda_i = 1, \forall i \in I \right\}. \tag{2.14}$$

For $\widehat{\boldsymbol{\lambda}}$ the estimate $\widehat{\boldsymbol{\beta}}$ is obtained by

$$\widehat{\boldsymbol{\beta}} = \boldsymbol{V} \cdot \widehat{\boldsymbol{\lambda}}. \tag{2.15}$$

Since the transformation from H- to V-representation of a polytope can be computationally very expensive it is advisable to use the representation that is available. Thus we

need an algorithm to find the optimal convex combination of vertices for solving problem (2.14).

The definition of centrosymmetry (cf. Definition 4) states $\boldsymbol{v} \in P \Leftrightarrow -1 \cdot \boldsymbol{v} \in P$. Thus the set $\mathcal{V}$ of all vertices of a centrosymmetric polytope includes two subsets of vertices $\mathcal{V}^+$ and $\mathcal{V}^-$ for which

$$\begin{array}{rclcrcl} \mathcal{V}^- & = & \{-1 \cdot \boldsymbol{v} : \boldsymbol{v} \in \mathcal{V}^+\}, & \quad & \mathcal{V}^+ & = & \{-1 \cdot \boldsymbol{v} : \boldsymbol{v} \in \mathcal{V}^-\}, \\ \mathcal{V}^+ \cap \mathcal{V}^- & = & \emptyset, & & \mathcal{V} & = & \mathcal{V}^+ \cup \mathcal{V}^-, \end{array} \tag{2.16}$$

holds. The structure allows to use only one of these two subsets, because each subset is its complement multiplied by -1. The idea is graphically illustrated for $p = 2$ by Figure 2.8.

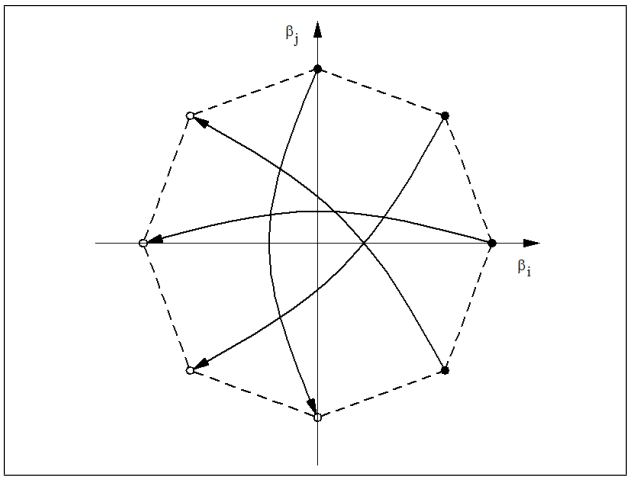

FIGURE 2.8: *The solid vertices are elements of $\mathcal{V}^+$. The remaining vertices of $\mathcal{V}^-$ are produced by multiplying with -1.*

It is obvious that the reduction of the set of vertices changes the constraint in (2.14). We take $\mathcal{V}^+$ and its index set of vertices $I^+ = \{1, ..., n_{V^+}\}$. With $\boldsymbol{v}_i^+$, $i \in I^+$, we denote the elements of $\mathcal{V}^+$. Now we structure $\mathcal{V}$ in the following way. The first n_{V^+} elements of $\mathcal{V}$ are equal to $\mathcal{V}^+$ and the second part of $\mathcal{V}$ is given by $\boldsymbol{v}_{n_{V^+}+i} = -1 \cdot \boldsymbol{v}_i^+$. Then, subject to the convexity constraint of $\boldsymbol{\lambda}$, for every $\boldsymbol{\beta} \in P$ holds

$$\begin{aligned} \boldsymbol{\beta} &= \sum_{i \in I} \lambda_i \boldsymbol{v}_i = \sum_{i \in I^+} \lambda_i \boldsymbol{v}_i^+ + \sum_{i \in I^+} \lambda_{n_{V^+}+i} \boldsymbol{v}_{n_{V^+}+i} \\ &= \sum_{i \in I^+} \lambda_i \boldsymbol{v}_i^+ + \sum_{i \in I^+} \lambda_{n_{V^+}+i} \cdot (-1) \cdot \boldsymbol{v}_i^+ = \sum_{i \in I^+} (\lambda_i - \lambda_{n_{V^+}+i}) \boldsymbol{v}_i^+ \\ &= \sum_{i \in I^+} \lambda_i^+ \boldsymbol{v}_i^+. \end{aligned}$$

Due to the convexity constraint of $\boldsymbol{\lambda}$ it is easy to show that $\sum_{i \in I^+} (\lambda_i - \lambda_{n_{V^+}+i}) = \sum_{i \in I^+} \lambda_i^+ \in [-1, +1]$. Analogously to (2.13) we convey $\mathcal{V}^+$ into a matrix $\boldsymbol{V}^+ =$

$\left(\boldsymbol{v}_1^+ | \cdots | \boldsymbol{v}_{n_{V^+}}^+\right)$. With the reduced set of vertices (2.14) turns into

$$\widehat{\boldsymbol{\lambda}}^+ = \text{argmin}_{\boldsymbol{\lambda}^+} \left\{ \left\| \boldsymbol{y} - \boldsymbol{X}\boldsymbol{V}^+ \cdot \boldsymbol{\lambda}^+ \right\|^2, \text{ s.t. } \sum_{i \in I^+} |\lambda_i^+| \leq 1 \right\} \tag{2.17}$$

where $\boldsymbol{\lambda}^+ = (\lambda_1^+, ..., \lambda_{n_{V^+}}^+)^T$. Analogously to (2.15) the estimate $\widehat{\boldsymbol{\beta}}$ is obtained by

$$\widehat{\boldsymbol{\beta}} = \boldsymbol{V}^+ \cdot \widehat{\boldsymbol{\lambda}}^+. \tag{2.18}$$

The constraint $\sum_{i \in I^+} |\lambda_i^+| \leq 1$ in (2.17) is a LASSO penalty. The equal sign holds if $\widehat{\boldsymbol{\beta}}_{OLS}$ is not a inner point of the constraining polytope. We assume that the tuning parameter t is appropriately chosen. The constrained regression problem (2.17) can be solved with the LARS algorithm from Efron et al. (2004) quite efficiently. So if a centrosymmetric V-polytope constrains the quadratic loss function the estimate is given by $\widehat{\boldsymbol{\beta}} = \boldsymbol{V}^+ \cdot \widehat{\boldsymbol{\lambda}}^+$ with $\widehat{\boldsymbol{\lambda}}^+$ given by (2.17)

2.4 Simulation study

In this section we investigate the performance of several methods. All simulations are based on the linear model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}^{true} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_n).$$

Each setting depends on the true parameter $\boldsymbol{\beta}^{true}$ and the true covariance matrix of the predictors $Cov(\boldsymbol{X}) = \boldsymbol{\Sigma} = \{\sigma_{i,j}\}_{i,j}$. The V8 penalty depends on $\rho_{i,j}$ and so we investigate different covariance matrices for normally distributed variables. Each data set consists of a training and a validation data set. The latter is used to determine the tuning parameters. The denotation n_{train}/n_{vali} is used to describe the number of observation of the corresponding data sets. For each simulation scenario we use 50 replications.

For every method we use the following procedure to measure the performance. We center the response and standardize the predictors of the training data set. $\overline{\boldsymbol{x}}^{train} = (\overline{\boldsymbol{x}}_1^{train}, ..., \overline{\boldsymbol{x}}_p^{train})^T$ denotes the vector of means in the training data set and $\overline{\boldsymbol{y}}^{train}$ is the mean of the response of the training data set. We use the transformed training data set to fit different models specified on a grid of tuning parameters. By retransformation of the coefficients we get a set of models $\mathcal{M}$. The validation data set is used to determine the model $\widehat{\boldsymbol{\beta}}^{opt} \in \mathcal{M}$ which minimizes the prediction error on the validation data set

$$\widehat{\boldsymbol{\beta}}^{opt} = \text{argmin}_{\widehat{\boldsymbol{\beta}} \in \mathcal{M}} \left\{ \sum_{i=1}^{n_{vali}} \left(y_i^{vali} - (\overline{\boldsymbol{y}}^{train} + (\boldsymbol{x}_i^{vali} - \overline{\boldsymbol{x}}^{train})^T \widehat{\boldsymbol{\beta}} \right)^2 \right\}.$$

For the V8 we use the validation data set to determine t and k in (2.10) where $k \in \{1, 2, 3\}$. Finally, we quantify the performance of $\widehat{\boldsymbol{\beta}}^{opt}$ by computing two model assessment measurements. The first is the *mean squared error of prediction* $\text{MSE} = (\boldsymbol{\beta}^{true} - \widehat{\boldsymbol{\beta}}^{opt})^T \boldsymbol{\Sigma} (\boldsymbol{\beta}^{true} -$

$\widehat{\boldsymbol{\beta}}^{opt}$) (where σ^2 is omitted), the second is the mean squared error for the estimation of the parameter vector $\text{MSE}_{\boldsymbol{\beta}}$. The MSE and $\text{MSE}_{\boldsymbol{\beta}}$ of the 50 replications are illustrated by boxplots. The standard deviation of the medians is calculated by bootstrapping with $B = 500$ bootstrap iterations.

Since we focus on shrinkage procedures with variable selection and the grouping property we compare V8, OSCAR, and Elastic Net (EN). It is remarkable that the EN penalty region is not polytopal. We add LASSO in our comparison because it is a special case of these three procedures. For the OSCAR we use the `MATLAB`-code which was available in 2007 on Bondell's homepage. The procedure tuned out to be computational very expensive. Therefore it was not possible to provide OSCAR for all settings. For computing the LASSO and EN we use also a `MATLAB`-routine (see Sjöstrand, 2005).

The settings are described in the following:

Sim 1 Let the underlying parameter vector be $\boldsymbol{\beta}_{true} = (3,\ 0,\ 0,\ 1.5,\ 0,\ 0,\ 0,\ 2)^T$ and standard error $\sigma = 3$. The correlation between the i-th and j-th predictor follows

$$\sigma_{i,\,j} = \rho^{|i-j|}, \forall i,\ j \in \{1,\ ...,\ 8\}. \tag{2.19}$$

The numbers of observations are 20/20. We choose different value for ρ, $\rho \in \{0.5,\ 0.7,\ 0.9\}$. Tibshirani (1996) used a similar setting for $\rho = 0.5$. For $\rho = 0.7$ this setting is equivalent ot the first setting of Bondell and Reich (2008). For $\rho = 0.9$ the range of pairwise correlations is the largest.

Sim 2 This setting is the same as the first setting excepting $\boldsymbol{\beta}_{true} = (3,\ 2,\ 1.5,\ 0,\ 0,\ 0,\ 0,\ 0)^T$. For $\rho = 0.7$ this setting is equivalent to the second setting of Bondell and Reich (2008).

Sim 3 In this setting the correlation is given by (2.19) but the coefficient vector is $\boldsymbol{\beta}_{true} = (0.85,\ 0.85,\ 0.85,\ 0.85,\ 0.85,\ 0.85,\ 0.85,\ 0.85)^T$. For $\rho = 0.5$ we get the third setting of Tibshirani (1996) and for $\rho = 0.7$ we get the third setting of Bondell and Reich (2008).

Sim 4 In this setting there are $p = 100$ predictors. The parameter vector is structured in blocks,

$$\begin{aligned}\boldsymbol{\beta}_{true} \;=\; (&\underbrace{2,\ ...,\ 2}_{10},\ \underbrace{0,\ ...,\ 0}_{10},\ \underbrace{4,\ ...,\ 4}_{10},\ \underbrace{0,\ ...,\ 0}_{10},\ \underbrace{-2,\ ...,\ -2}_{10},\ \underbrace{0,\ ...,\ 0}_{10},\\ &\underbrace{0,\ ...,\ 0}_{15},\ \underbrace{2,\ ...,\ 2}_{5},\ \underbrace{0,\ ...,\ 0}_{20})^T\end{aligned}$$

and $\sigma = 15$. Between the first six blocks of 10 variables there is no correlation. Within these six blocks we use the correlation structure from (2.19). The remaining 40 variables are uncorrelated. The numbers of observations are 200/200. As noted above this setting could not be analyzed by OSCAR.

Sim 5 The last setting is equal to the forth setting but numbers of observations changes to 50/50. So there is the $p \geq n$ case where the OLS does not exist. The OSCAR is not calculable.

The results are summed up in Table 2.1. For illustration we show the simulation scenarios with $\rho = 0.9$ in Figure 2.9.

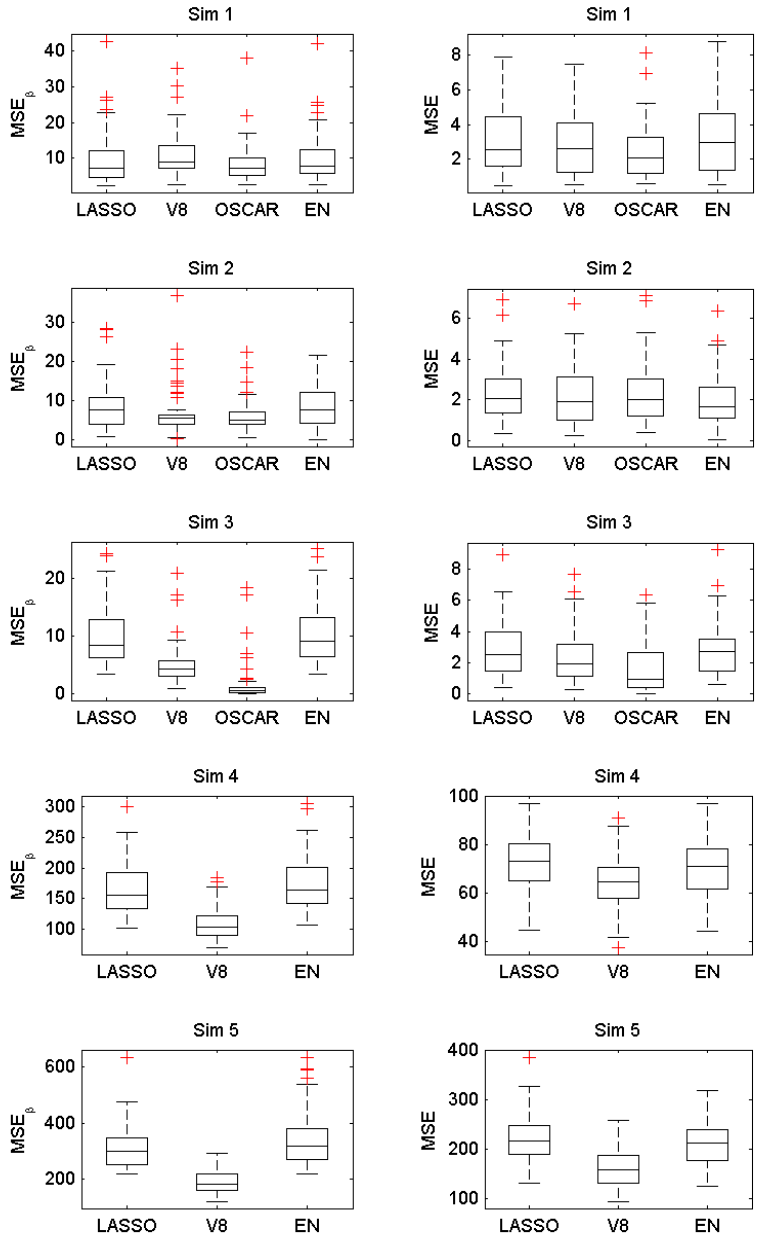

FIGURE 2.9: *Boxplots of the mean squared error of prediction* MSE *and the mean squared error of* $\boldsymbol{\beta}$, $\text{MSE}_{\boldsymbol{\beta}}$, *for the different procedures and the five simulation settings with* $\rho = 0.9$.

		$\text{MSE}_{\boldsymbol{\beta}}$	MSE	$\text{MSE}_{\boldsymbol{\beta}}$	MSE	$\text{MSE}_{\boldsymbol{\beta}}$	MSE
		$\rho = 0.5$		$\rho = 0.7$		$\rho = 0.9$	
Sim 1	LASSO	3.152	3.128	5.197	3.426	7.055	2.558
		(0.455)	(0.392)	(0.665)	(0.413)	(0.880)	(0.429)
	V8	3.334	3.206	6.490	3.845	8.942	2.638
		(0.394)	(0.321)	(0.688)	(0.340)	(0.495)	(0.311)
	OSCAR	2.995	3.376	5.837	3.484	7.167	2.082
		(0.469)	(0.351)	(0.478)	(0.342)	(0.727)	(0.314)
	EN	3.244	3.419	5.676	3.483	7.627	2.949
		(0.546)	(0.458)	(0.576)	(0.480)	(1.000)	(0.577)
Sim 2	LASSO	3.419	3.527	4.661	3.597	7.475	2.046
		(0.620)	(0.306)	(0.603)	(0.406)	(0.659)	(0.236)
	V8	3.088	3.048	4.347	2.842	5.376	1.920
		(0.359)	(0.232)	(0.603)	(0.406)	(0.261)	(0.221)
	OSCAR	2.905	3.536	4.263	3.357	4.835	1.982
		(0.349)	(0.335)	(0.301)	(0.309)	(0.495)	(0.174)
	EN	3.336	3.111	4.870	2.936	7.618	1.667
		(0.555)	(0.421)	(0.547)	(0.273)	(0.684)	(0.186)
Sim 3	LASSO	4.104	3.904	4.882	3.272	8.405	2.527
		(0.201)	(0.373)	(0.467)	(0.281)	(1.094)	(0.239)
	V8	3.163	3.419	3.590	3.081	4.233	1.942
		(0.205)	(0.337)	(0.271)	(0.287)	(0.297)	(0.294)
	OSCAR	1.963	2.644	1.065	2.291	0.455	0.966
		(0.277)	(0.431)	(0.436)	(0.234)	(0.108)	(0.276)
	EN	4.923	4.600	6.647	3.651	9.130	2.692
		(0.255)	(0.454)	(0.420)	(0.408)	(1.083)	(0.249)
Sim 4	LASSO	91.736	90.709	100.463	73.466	156.118	73.466
		(4.695)	(3.249)	(4.657)	(3.232)	(7.800)	(1.712)
	V8	77.429	77.511	74.947	64.996	103.720	64.505
		(3.960)	(3.265)	(2.038)	(3.089)	(3.130)	(1.881)
	OSCAR	—	—	—	—	—	—
		(—)	(—)	(—)	(—)	(—)	(—)
	EN	97.818	89.189	104.211	72.760	164.128	70.958
		(3.778)	(3.589)	(5.130)	(2.200)	(7.583)	(1.944)
Sim 5	LASSO	213.602	260.045	259.152	270.264	298.498	215.500
		(7.369)	(12.753)	(12.542)	(11.104)	(13.134)	(12.688)
	V8	175.526	221.260	175.526	195.957	181.008	159.129
		(6.433)	(14.614)	(5.972)	(6.579)	(10.058)	(7.650)
	OSCAR	—	—	—	—	—	—
		(—)	(—)	(—)	(—)	(—)	(—)
	EN	228.235	262.936	275.566	254.742	317.582	212.603
		(9.050)	(10.609)	(8.712)	(14.413)	(16.670)	(8.267)

TABLE 2.1: *Median of mean squared errors of prediction* MSE *and the median of mean squared errors of* $\boldsymbol{\beta}$, $\text{MSE}_{\boldsymbol{\beta}}$. *The corresponding standard deviations are estimated by bootstrapping with* 500 *bootstrap iterations given in brackets.*

It is obvious that Simulation 1 is a challenge for the V8 procedure. Forced by the underlying correlation structure the V8 tries to group the influential variable with their neighbors that have no influence on the response. As expected, for the second setting the V8 procedure shows better performance. It competes well with OSCAR, the other procedures perform worse in particular in terms of MSE.

Although we are mainly interested in procedures with variable selection property we chose setting 3 because it was often used in the literature (see Bondell and Reich, 2008; Tibshirani, 1996; Zou and Hastie, 2005). In this setting the OSCAR is the best procedure because it can group all the variables. All other procedures are unable to do this. But the setting shows that adding new vertices to the LASSO penalty yields definitely better results. The performance of the LASSO is topped by both polytopes with additional vertices (OSCAR and V8). The V8 is the second best in both criteria.

The two last simulations show that the V8 procedure works quite well especially for the $p \gg n$-case. LASSO as well as EN were outperformed by V8. The computational costs of the OSCAR were so high that it was not possible to include it in the competition.

2.5 Data Example

The body fat data set has been published by Penrose et al. (1985). The aim was to estimate the percentage of body fat of 252 men by use of thirteen regressors. The regressors are age (1), weight (lbs) (2), height (inches) (3), neck circumference (4), chest circumference (5), abdomen 2 circumference (6), hip circumference (7), thigh circumference (8), knee circumference (9), ankle circumference (10), biceps (extended) circumference (11), forearm circumference (12), and wrist circumference (13). All circumferences are measured in cm. Some of the predictors are highly correlated, i.e. $\rho_{ij} \approx 0.9$. The response has been calculated from the equation by Siri (1956) using the body density determined by underwater weighting. In order to compare the performances of the different procedures we split the data at random into 25 training sets with $n_{train} = 151$ and test sets with $n_{test} = 101$. For the second tuning parameter of the V8 we chose $k = 1$ because in the simulation study it turned out to work quite well. The remaining tuning parameters were determined by tenfold cross-validation on the training data set. Afterwards we estimated the model on the whole training data set. The median of prediction errors across 25 random splits were 22.03 (LASSO), 21.32 (V8), 21.99 (OSCAR) and 23.30 (Elastic Net). The corresponding boxplots are shown in Figure 2.10. It is seen that correlation based penalization has the best performance in terms of mean squared errors. The OSCAR and the EN depend on two tuning parameters. So the computational costs are high especially for fine grids. The OSCAR procedure has the highest costs and does not preform better than the V8.

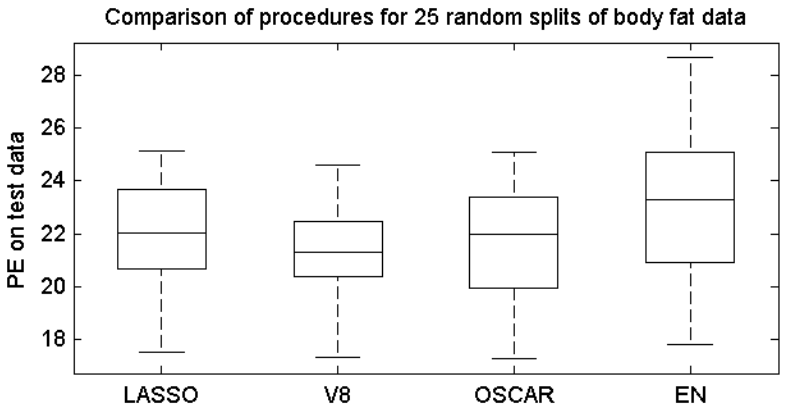

FIGURE 2.10: *Boxplots of different methods for 25 random splits of the body fat data set with* $n_{train} = 151$ *and* $n_{test} = 101$.

2.6 Concluding Remarks

It has been shown that polytopes are very flexible geometric objects which are useful for constraining regression problems. In particular their flexibility can be used to design specific polytopes that incorporate additional information contained in the data. The V8 procedure has been designed in this spirit as a correlation-based V-polytope. Although one can see it as an advantage that OSCAR and the elastic net do not need additional weights their inclusion is an option in the construction of estimators.

For the computation of least squares problems which are constrained by centrosymmetric V-polytopes a modification of the LARS-algorithm has been proposed. V8 works quite well, in particular in the $p \gg n$ case because it uses the efficient LARS-Algorithm. Therefore, it can be applied where OSCAR fails because of its high computational costs. The second tuning parameter of V8, k, has no great influence on the results. So we chose $k = 1$. In this case V8 uses only one tuning parameter, which reduces computational costs for searching optimal tuning parameters.

We restricted attention here to penalty regions which do not assume order information in the predictors. Therefore, we considered only the LASSO and OSCAR as specific polytope based procedures. If order information is available, as for example in signal regression, a successful strategy is to use the Fused Lasso (Tibshirani et al., 2005), which is also a polytopal penalized regression problem with polytopes that reflect the order of predictors.

Of course it is possible to use the presented algorithm for OSCAR by substituting $\boldsymbol{V}^+$ of (2.17) in an appropriate way. The vertices are given in Proposition 1. But the large numbers of vertices of the OSCAR penalty makes the algorithm not more efficient than the original algorithm proposed by (Bondell and Reich, 2008).

Chapter 3

Pairwise Fused Lasso

The Fused Lasso penalty from Tibshirani et al. (2005) combines variable selection with clustering neighbored predictors. In this chapter the Fused Lasso penalty is generalized to unordered predictors. This chapter is based on Petry et al. (2010).

3.1 Introduction

Regularized estimation of regression parameters has been investigated thoroughly within the last decade. With the introduction of LASSO, proposed by Tibshirani (1996), methods for sparse modeling in the high-predictor case became available. In the following years many alternative regularized estimators which include variable selection were proposed, among them the elastic net (Zou and Hastie, 2005), SCAD (Fan and Li, 2001), the Dantzig selector (Candes and Tao, 2007) and boosting approaches (for example Bühlmann and Yu, 2003; Bühlmann and Hothorn, 2007). Meanwhile most procedures are also available for generalized linear models (GLMs). Since we will also work within the GLM framework in the following some notation is introduced.

Let the generalized linear model (GLM) with response function $h(.)$ be given by

$$\boldsymbol{\mu} = E(\boldsymbol{y}|\boldsymbol{X}) = h(\mathbf{1}\beta_0 + \boldsymbol{X}\boldsymbol{\beta}),$$

where $\boldsymbol{y} = (y_1, ..., y_n)^T$ is the response vector and $\boldsymbol{X}$ is the design matrix. It is assumed that the predictors are standardized, $\sum_{i=1}^n x_{ij} = 0$ and $(n-1)^{-1}\sum_{i=1}^n x_{ij}^2 = 1$, $\forall j \in \{1, ..., p\}$. In the linear predictor $\boldsymbol{\eta} = \mathbf{1}\beta_0 + \boldsymbol{X}\boldsymbol{\beta}$ the intercept β_0 is separated because usually it is not penalized. With $\boldsymbol{\beta}_0 = (\beta_0, \boldsymbol{\beta}^T)$ we denote the parameter vector including the intercept β_0. Given the ith observation $\boldsymbol{X}_i$ the y_i are (conditionally) independent observations from a simple exponential family

$$f(y_i|\theta_i, \phi) = \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{\phi} + c(y_i, \phi)\right\}, \tag{3.1}$$

where θ_i is the natural parameter of the family, ϕ is a scale or dispersion parameter and $b(.), c(.)$ are specific functions corresponding to the type of the family.

Penalized likelihood estimates of coefficients have the general form

$$\widehat{\boldsymbol{\beta}}_0 = \text{argmin}_{\boldsymbol{\beta}_0} \left\{ l(\boldsymbol{\beta}_0) + P_{\boldsymbol{\lambda}}(\boldsymbol{\beta}) \right\},$$

where $P_{\boldsymbol{\lambda}}(\boldsymbol{\beta})$ is the penalty term that regularizes the estimates and $l(\boldsymbol{\beta}_0)$ is the negative log-likelihood function which corresponds to (3.1). Hoerl and Kennard (1970) propose the Ridge regression estimator. They use

$$P_\lambda^R(\boldsymbol{\beta}) = \lambda \sum_{j=1}^{p} \beta_j^2$$

as penalty term. The ridge estimator has frequently smaller prediction error than ordinary maximum likelihood (ML) estimates but does not select predictors. The LASSO penalty

$$P_\lambda^L(\boldsymbol{\beta}) = \lambda \sum_{j=1}^{p} |\beta_j|$$

proposed by Tibshirani (1996) has the advantage that coefficients whose corresponding predictors have vanishing or low influence on the response are shrunk to zero.

In the case of highly correlated influential covariates the LASSO procedure tends to select only few of these. Zou and Hastie (2005) present the elastic net (EN) to avert this property of the LASSO. The EN combines variable selection with grouping variables. In terms of Zou and Hastie (2005) an estimator exhibits the grouping property if it tends to estimate the absolute value of coefficients (nearly) equal if the corresponding standardized predictors are highly correlated. As discussed by Zou and Hastie (2005) the LASSO does not group predictors and estimates maximal n predictors unequal to 0. The EN avoids this effects. Its penalty term is the sum of LASSO and ridge penalty

$$P_{\lambda_1}^L(\boldsymbol{\beta}) + P_{\lambda_2}^R(\boldsymbol{\beta})$$

and is a strongly convex penalty which can also perform variable selection. Nowadays `R` packages for solving the LASSO- or the EN-penalized likelihood problems for GLMs are available. For example Goeman (2010a) and Friedman et al. (2010b) proposed algorithms to solve elastic net penalized regression problems. Both algorithms are available as `R`-packages `penalized` and `glmnet`. Lokhorst et al. (2007) and Park and Hastie (2007b) provided the `R`-packages the `lasso2` and `glmpath` for solving LASSO penalized regression problem.

More recently, several alternative methods that also show grouping have been proposed. Bondell and Reich (2008) proposed OSCAR for Octagonal Shrinkage and Clustering Algorithm for Regression. An attractive feature of OSCAR is that it can group very strictly. For specific choice of the tuning parameters the estimates of coefficients are exactly equal. Therefore one obtains clustered predictors where one cluster shares the same coefficient. Typically one big cluster has estimates zero representing the predictors that have not been selected. Tutz and Ulbricht (2009) considered correlation based regularization terms that

explicitly take the correlation of predictors into account. In order to obtain variable selection the correlation-based penalty is used within a boosting algorithm. Alternatively an LASSO term has to be added to the correlation-based terms (see Anbari and Mkhadri, 2008).

In the present chapter an alternative method that enforces the grouping effect is proposed. It uses penalty terms that are similar to the Fused Lasso (FL) proposed by Tibshirani et al. (2005) and shows good performance in terms of variable selection and prediction.

3.2 Pairwise Fused Lasso (PFL)

The original fused lasso (Tibshirani et al., 2005) was developed for ordered predictors or signals as predictors and metrical response. For such kind of predictors it is possible to use the distances between predictors to obtain clustering. For example if the predictors are signals depending on frequencies the predictor is ordered by the frequency. This suggests that adjacent frequencies should have similar influence on the response. Or in others words small chances of the frequency should only have small influence on the response. For inducing variable selection (Tibshirani et al., 2005) add an LASSO term to the penalization of adjacent coefficients. Thus the fused lasso penalty

$$P^{FL}_{\lambda_1,\lambda_2}(\boldsymbol{\beta}) = \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=2} |\beta_j - \beta_{j-1}|,\ \lambda_1,\ \lambda_2 \leq 0, \tag{3.2}$$

penalizes the difference between the coefficients of adjacent predictors β_j and β_{j-1}. With proper selection of tuning parameters adjacent predictors are fused or grouped. The first summand (the LASSO term) of the fused lasso penalty enforces variable selection, the second enforces fusion.

The pairwise fused lasso (PFL), which is proposed here, extends the fused lasso (Tibshirani et al., 2005) to situations where the predictors have no natural ordering. Fusion refers to all possible pairs of predictors and not only to adjacent ones. Thus, the pairwise fused lasso penalty is defined by

$$P^{PFL}_{\lambda,\,\alpha}(\boldsymbol{\beta}) = \lambda \left[\alpha \sum_{j=1}^{p} |\beta_j| + (1-\alpha) \sum_{j=2}^{p} \sum_{k=1}^{j-1} |\beta_j - \beta_k| \right], \tag{3.3}$$

where $\lambda > 0$ and α with $\alpha \in [0,\, 1]$ are the tuning parameters. The first term of the pairwise fused lasso penalty is the LASSO penalty and accounts for variable selection, the second term represents the sum of the absolute values of all pairwise differences of regression coefficients. This part of the penalty induces clustering.

By using all pairwise differences the pairwise fused lasso assumes no ordering of the predictors. For categorical predictors a similar penalty has been used for factor selection in ANOVA by Bondell and Reich (2009), and for categorical variable selection by Gertheiss and Tutz (2010).

Of cause it is possible to give the PFL penalty term (3.3) alternatively by

$$P^{PFL}_{\lambda_1,\lambda_2}(\boldsymbol{\beta}) = \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_1 \sum_{j=2}^{p} \sum_{k=1}^{j-1} |\beta_j - \beta_k|,\ \lambda_1,\ \lambda_2 \leq 0, \tag{3.4}$$

which is more similar to (3.2). We prefer the notation of Equation (3.3) because the range of one tuning parameter, namely α, is known. For special choices of λ and α in (3.3) on the one hand and λ_1 and λ_2 in (3.4) on the other hand the result are equal.

Soil Data - An Illustrating Example

In the soil data, which were used by Bondell and Reich (2008), the response is rich-cove forest diversity (measured by the number of different plants species) in the Appalachian Mountains of North Carolina and the explaining covariates are 15 characteristics. Twenty areas of the same size were surveyed. The number of observations was 20 which is close to the number of predictors which was 15. The data can be partitioned into two blocks. On the one hand there is a group of 7 highly correlated predictors. This group contains cationic covariates, 4 cations (calcium, magnesium, potassium, and sodium) and 3 measurements that are very close to them. The other group of covariates contains 4 other chemical elements and 4 other soil characteristics, for example pH-value. The correlations within this group is not very high. It is remarkable that the design matrix has not full rank.

For illustration we use four different methods, LASSO and three PFL methods. The first segments of the coefficient paths are given in Figure 3.1 and demonstrate the selecting and grouping property. In the left column of Figure 3.1 the paths of the cationic type covariates are shown and in the right column the path of remaining covariates are illustrated. It is seen that there is a strong similarity between the LASSO and the PFL method for $\alpha = 0.98$. For large values of the tuning parameter λ, i.e. small values of $|\boldsymbol{\beta}|$, the LASSO selects only few covariates. This effect is also seen in the group of the highly correlated cationic covariates. As discussed by Zou and Hastie (2005) or Breiman (1996) variable selection inner groups of highly correlated covariates can induces instability to the estimate. For smaller value of α the selection part becomes weaker and the fusion part stronger. It is seen that for $\alpha = 0.9$ and more distinctly for $\alpha = 0.1$ the highly correlated variables are fused, but there is hardly any effect beside selection for the weaker correlated variables in the second column of Figure 3.1.

Extended Versions of Fused Lasso

The pairwise fused lasso penalty (3.3) can be modified by adding different weights to achieve an improvement of the prediction accuracy or of the mean squared error of the estimated parameter vector. Accordingly, a modification of the penalty term is

$$P^{PFL}_{\lambda,\alpha,\boldsymbol{w}}(\boldsymbol{\beta}) = \lambda \left[\alpha \sum_{j=1}^{p} w_j |\beta_j| + (1-\alpha) \sum_{j=2}^{p} \sum_{k=1}^{j-1} w_{jk} |\beta_j - \beta_k| \right], \tag{3.5}$$

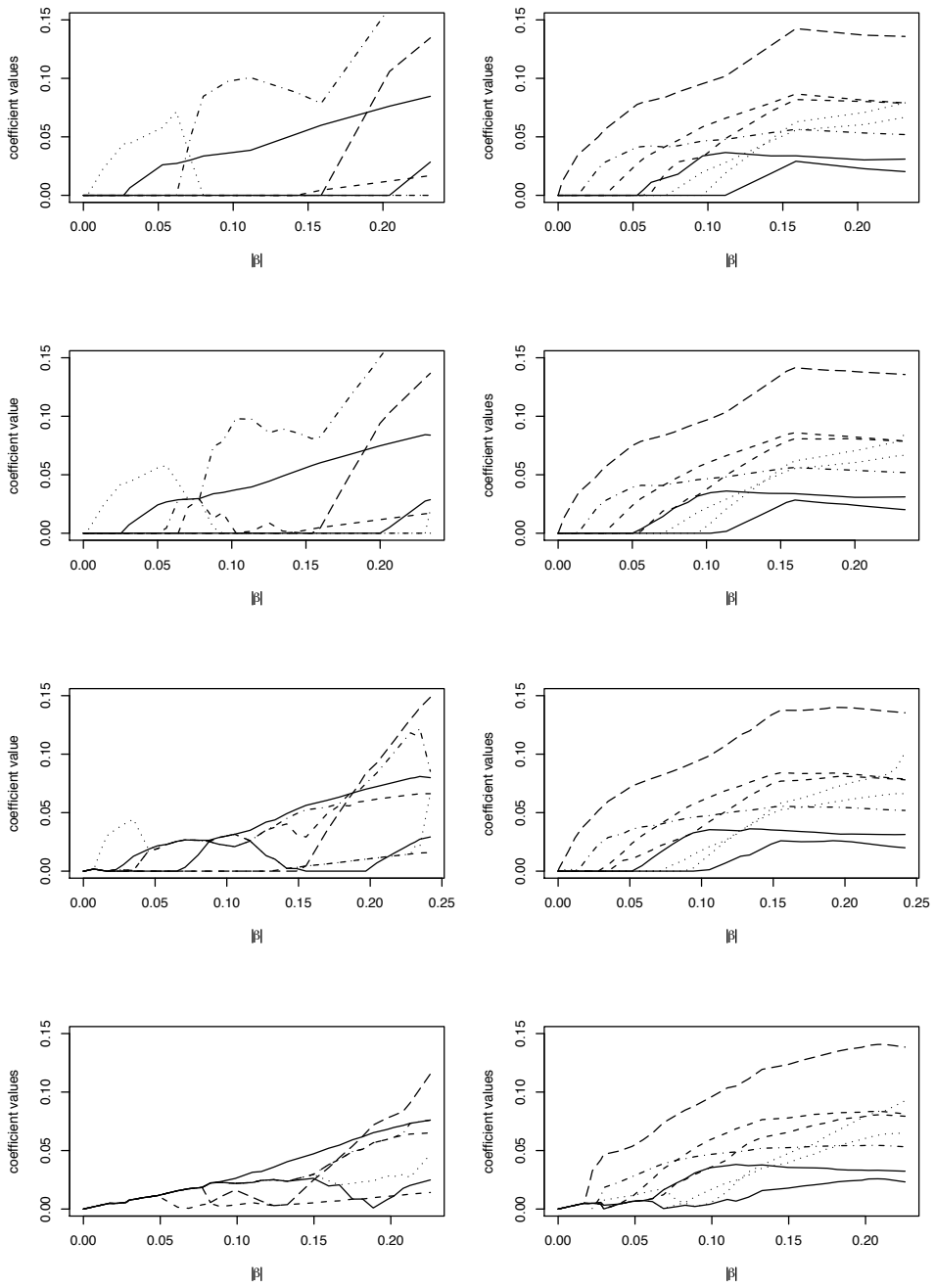

FIGURE 3.1: *First segments of the solution paths for standardized coefficients on the whole soil data set for decreasing tuning parameter λ. Left column: paths of the cationic covariates. Right column: paths of the non cationic covariates. First row: coefficient path of the LASSO. Second row: coefficient path of PFL model with small clustering part ($\alpha = 0.98$). Third row: coefficient path of PFL model with $\alpha = 0.9$. Fourth row: coefficient path of PFL model with dominating fusion part ($\alpha = 0.02$).*

where w_j and w_{jk} are additional weights. One possibility is to choose $w_j = |\beta_j^{ML}|^{-1}$ and $w_{jk} = |\beta_j^{ML} - \beta_k^{ML}|^{-1}$, where β_i^{ML} denotes the ith component of maximum likelihood estimate. This choice is motivated by the adaptive LASSO (Zou, 2006) and its oracle properties. These data-dependent weights can yield better prediction error if the maximum likelihood is well conditioned. In contrast to the simple pairwise fused lasso (3.3) where all parameters have the same amount of shrinkage strength the penalty term (3.5) varies the shrinkage effect across coefficients. Large values of $|\beta_i^{ML}|$ yield small weights w_i and consequently weaker shrinkage of the corresponding parameters. If the maximum likelihood estimates of the jth and the kth predictor have nearly the same value the weight w_{jk} becomes high. So w_{jk} causes a great influence of $|\beta_j - \beta_k|$ on the penalized negative log-likelihood problem. Because of the great weight the difference $|\beta_j - \beta_k|$ must become small by minimizing the penalized negative log-likelihood.

Another possibility is to include the correlation among predictors into the penalty. Zou and Hastie (2005) showed a relationship between correlation and grouping such that strongly correlated covariates tend to be in or out of the model together, but the correlation structure was not used explicitly in the penalty term. A regularization method, which is based on the idea that highly correlated covariates should have (nearly) the same influence on the response except to their sign, is the correlation based penalty considered by Tutz and Ulbricht (2009). Coefficients of two predictors are weighted according to their marginal correlation. As a result, the intensity of penalization depends on the correlation structure. Inspired by this consideration the penalty term of the pairwise fused lasso can be extended to

$$P_{\lambda,\alpha,\widehat{\boldsymbol{\rho}}}^{PFL}(\boldsymbol{\beta}) = \lambda \left[\alpha \sum_{j=1}^{p} |\beta_j| + (1-\alpha) \sum_{j=2}^{p} \sum_{k=1}^{j-1} \frac{1}{1-|\widehat{\rho}_{jk}|} |\beta_j - \operatorname{sign}(\widehat{\rho}_{jk})\beta_k| \right], \tag{3.6}$$

where $\widehat{\rho}_{jk}$ denotes the estimated marginal correlation between the jth and the kth predictor. The factor $\operatorname{sign}(\widehat{\rho}_{jk})$ is caused by the fact that two negatively correlated predictors have the same magnitude of influence but different signs. That is, for $\widehat{\rho}_{jk} \to 1$, the coefficients $\widehat{\beta}_j$ and $\widehat{\beta}_k$ are nearly the same and for $\rho_{jk} \to -1$, $\widehat{\beta}_j$ will be close to $-\widehat{\beta}_k$, respectively. In the case of uncorrelated predictors ($\widehat{\rho}_{jk} = 0$) we obtain the usual, unweighted pairwise fused lasso penalty.

Since the marginal correlation measures the interaction between the predictors $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$ without taking further covariates into account, we also investigate the correlation based penalty in Equation (3.6) with partial correlations instead of the marginal ones. The partial correlation determines to what extent the correlation between two variables depends on the linear effect of the other covariates (Whittaker, 1990). Thereby, the aim is to eliminate this linear effect. We compute the partial correlation matrix with the `R` package `corpcor` (Schäfer et al., 2009). In this package a method for the regularization of (partial) correlation matrix is implemented. Especially in ill conditioned problems the regularization of (partial) correlation matrices makes sense. In general the correlation based weights can be substituted by any dependency measurement which is normed on $[-1, 1]$. A combination of correlation based weights and maximum likelihood weights is possible. But this quite complicate penalty term does not improve the performance.

3.2.1 Solving the Penalized ML Problem

In this section we discuss two procedures for solving the PFL problem

$$\widehat{\boldsymbol{\beta}}_0^{PFL} = \text{argmin}_{\boldsymbol{\beta}_0} \left\{ l(\boldsymbol{\beta}_0) + P_{\lambda,\alpha}^{PFL}(\boldsymbol{\beta}) \right\},$$

where $P_{\lambda,\alpha}^{PFL}(\boldsymbol{\beta})$ can be modified to include weights or correlation terms. The first approach works only for normally distributed response. It is based on the LARS algorithm from Efron et al. (2004). The second procedure is a generic algorithm based on local quadratical approximation (LQA). The basic principles of this algorithm were given by Osborne et al. (2000) and Fan and Li (2001). The general LQA algorithm can solve a very wide class of penalized likelihood problems (see Ulbricht, 2010b) and is available as the `R`-package `lqa` (Ulbricht, 2010a). We will give a short introduction to the algorithm in the second part of this section.

Metric Regression and the LARS approach

We assume that $\boldsymbol{y}$ is centered and the response is normally distributed. Then one has to solve the penalized least square problem

$$\widehat{\boldsymbol{\beta}}^{PFL} = \text{argmin}_{\boldsymbol{\beta}} \left\{ \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + P_{\lambda,\alpha}^{PFL}(\boldsymbol{\beta}) \right\}. \tag{3.7}$$

It is helpful to reparameterize the problem as follows. Let new parameters be defined by

$$\begin{array}{rcl} \theta_{jk} & = & \beta_j - \beta_k,\ 1 \le k < j \le p, \\ \theta_{j0} & = & \beta_j,\ 1 \le j \le p, \end{array} \tag{3.8}$$

with the restriction

$$\theta_{jk} = \theta_{j0} - \theta_{k0},\ 1 \le k < j \le p. \tag{3.9}$$

Let $\mathbf{0}_{p\times\binom{p}{2}}$ be a $p \times \binom{p}{2}$-matrix zero matrix. Then we expand design matrix $\boldsymbol{X}$ with $\mathbf{0}_{p\times\binom{p}{2}}$ to a new design matrix $(\boldsymbol{X}|\mathbf{0}_{p\times\binom{p}{2}})$. The corresponding parameter vector is

$$\boldsymbol{\theta} = (\theta_{10},\ ...,\ \theta_{p0},\ \theta_{21},\ ...,\ \theta_{p(p-1)})^T. \tag{3.10}$$

With the PFL penalty having the form

$$P_{\lambda,\alpha}^{PFL}(\boldsymbol{\theta}) = \lambda \left[\alpha \sum_{j=1}^{p} w_{j0}|\theta_{j0}| + (1-\alpha) \sum_{j=1}^{p-1} \sum_{k=j+1}^{p} w_{jk}|\theta_{jk}| \right],$$

the restriction (3.9) is incorporated by using an additional quadratic penalty term $\sum_{j=1}^{p-1}\sum_{k=j+1}^{p}(\theta_{j0} - \theta_{k0} - \theta_{jk})^2$ weighted by a large tuning parameter γ. This yields

$$\begin{array}{rcl} \widehat{\boldsymbol{\theta}}^{PFL} & = & \text{argmin}_{\boldsymbol{\theta}} \Big\{ \|\boldsymbol{y} - (\boldsymbol{X}|\mathbf{0}_{p\times\binom{p}{2}})\|^2 \\ & & +\gamma \sum_{j=1}^{p-1}\sum_{k=j+1}^{p}(\theta_{j0} - \theta_{k0} - \theta_{jk})^2 \\ & & +\lambda \left[\alpha \sum_{j=1}^{p} w_{j0}|\theta_{j0}| + (1-\alpha) \sum_{j=1}^{p-1}\sum_{k=j+1}^{p} w_{jk}|\theta_{jk}| \right] \Big\}. \end{array} \tag{3.11}$$

For $\gamma \to \infty$ the restriction (3.9) is fulfilled. The reparameterization (3.8) allows to formulate the approximate estimator (3.11) as a LASSO type problem. Similar reparameterizations were used by Zou and Hastie (2005) to represent the elastic net problem as a LASSO type problem. In the present problem one uses

$$\begin{aligned} \widehat{\boldsymbol{\theta}}^{PFL} &= \operatorname{argmin}_{\boldsymbol{\theta}} \Big\{ \|\boldsymbol{y}_0 - \widetilde{\boldsymbol{D}}\boldsymbol{\theta}\|^2 \\ &\qquad + \lambda \left[\alpha \textstyle\sum_{j=1}^{p} w_{j0}|\theta_{j0}| + (1-\alpha) \sum_{j=1}^{p-1} \sum_{k=j+1}^{p} w_{jk}|\theta_{jk}| \right] \Big\}, \end{aligned} \tag{3.12}$$

where $\boldsymbol{y}_0 = (\boldsymbol{y}^T, \mathbf{0}^T_{\binom{p}{2}})^T$ and $\mathbf{0}$ denotes a zero vector of length $\binom{p}{2}$. $\widetilde{\boldsymbol{D}}$ is the design matrix

$$\widetilde{\boldsymbol{D}} = \begin{pmatrix} \boldsymbol{X} | \mathbf{0}_{p \times \binom{p}{2}} \\ \sqrt{\gamma}\boldsymbol{C} \end{pmatrix},$$

where the matrix $\boldsymbol{C}$ is the $p \times \left(\binom{p}{2} + p\right)$-matrix which accounts for the restriction (3.9) which is equivalent to

$$\theta_{j0} - \theta_{k0} - \theta_{jk} = 0,\ 1 \le k < j \le p. \tag{3.13}$$

So the restriction (3.9) is fulfilled if $\boldsymbol{C}\boldsymbol{\theta} = \mathbf{0}_{\binom{p}{2}}$ and $\boldsymbol{C}$ has the following form. Let $\boldsymbol{\delta}_{jk}$, $1 \le k < j \le p$, denote a p-dimensional row vector with -1 at the kth and $+1$ at the jth component and zero otherwise. Let $\boldsymbol{\tau}_m$ denote a $\binom{p}{2}$-dimensional row vector whose mth component is -1 and zero otherwise. Then all constrains given by (3.9) resp. (3.13) can be summarized in matrix notation

$$\boldsymbol{C} = \begin{pmatrix} \boldsymbol{\delta}_{21} & \boldsymbol{\tau}_1 \\ \boldsymbol{\delta}_{31} & \boldsymbol{\tau}_2 \\ \vdots & \vdots \\ \boldsymbol{\delta}_{p1} & \boldsymbol{\tau}_{p-1} \\ \boldsymbol{\delta}_{32} & \boldsymbol{\tau}_p \\ \boldsymbol{\delta}_{42} & \boldsymbol{\tau}_{p+1} \\ \vdots & \vdots \\ \boldsymbol{\delta}_{p(p-1)} & \tau_{\binom{p}{2}} \end{pmatrix}. \tag{3.14}$$

Let $\Theta = \{(i,\, j) | 0 \le j < i < p\}$ denote the index set of the components of $\boldsymbol{\theta}$ given by (3.10) one obtains

$$\begin{aligned} \widehat{\boldsymbol{\theta}}^{PFL} &= \operatorname{argmin}_{\boldsymbol{\theta}} \left\{ \|\boldsymbol{y}_0 - \widetilde{\boldsymbol{D}}\boldsymbol{\theta}\|^2 + \lambda \Big(\sum_{j=1}^{p} |\alpha \cdot \theta_{j0}| + \sum_{j=1}^{p-1} \sum_{k=j+1}^{p} |(1-\alpha) \cdot \theta_{jk}| \Big) \right\} \\ &= \operatorname{argmin}_{\boldsymbol{\theta}} \left\{ \|\boldsymbol{y}_0 - \widetilde{\boldsymbol{D}}\boldsymbol{\theta}\|^2 + \lambda \Big(\sum_{t \in \Theta} |\alpha \cdot \theta_t| + |(1-\alpha) \cdot \theta_t| \Big) \right\}. \end{aligned} \tag{3.15}$$

Equation (3.15) is a LASSO problem on the expanded design matrix $\widetilde{\boldsymbol{D}}$ weighted by α and $(1-\alpha)$. The weights can be included by multiplying $\widetilde{\boldsymbol{D}}$ with the reciprocals of weights

$$\boldsymbol{D} = \widetilde{\boldsymbol{D}}\,\mathrm{diag}(\alpha \cdot w_{10},\, ...,\, \alpha \cdot w_{p0},\, (1-\alpha) \cdot w_{21},\, ...,\, (1-\alpha) \cdot w_{p(p-1)})^{-1}, \tag{3.16}$$

to obtain

$$\widehat{\boldsymbol{\theta}}^{PFL} = \mathrm{argmin}_{\boldsymbol{\theta}} \left\{ \|\boldsymbol{y}_0 - \boldsymbol{D}\boldsymbol{\theta}\|^2 + \lambda(\sum_{t\in\Theta} |\theta_t|) \right\}.$$

Finally to get $\widehat{\boldsymbol{\beta}}^{PFL}$ we have to multiply the first p components of $\widehat{\boldsymbol{\theta}}^{PFL}$ with $\alpha^{-1}\,\mathrm{diag}(\alpha w_{10},\, ...,\, \alpha w_{p0})$. For the correlation based pairwise fused lasso we have to modify the submatrix $\boldsymbol{C}$ of $\widetilde{\boldsymbol{D}}$. If $\mathrm{sign}(\widehat{\rho}_{jk}) = -1$ then δ_{jk}, $1 \le k < j \le p$, is a p-dimensional row vector where the kth and the jth component are $+1$ and all remaining are zero (see equation (3.6)). It is remarkable that for $w_{jk} = 1$, $0 \le 1 < k \le p$, in (3.16) we get the unweighted PFL as given in (3.3).

Generalized Linear Models and the LQA Approach

A general class of penalized generalized linear models can be fitted by using the local quadratic approximation (LQA) approach (Ulbricht, 2010b). The LQA algorithm solves penalized minimization problems

$$\widehat{\boldsymbol{\beta}}_0 = \mathrm{argmin}_{\boldsymbol{\beta}_0} \left\{ l(\boldsymbol{\beta}_0) + P_{\boldsymbol{\lambda}}^{\delta}(\boldsymbol{\beta}) \right\}, \tag{3.17}$$

where $l(\boldsymbol{\beta}_0)$ is the negative log-likelihood of the underlying generalized linear model and the penalty term is a sum of J penalty functions having the form

$$P_{\boldsymbol{\lambda}}^{\delta}(\boldsymbol{\beta}) = \sum_{j=1}^{J} p_{\lambda_j,j}(|\boldsymbol{a}_j^T\boldsymbol{\beta}|), \tag{3.18}$$

where the $\boldsymbol{a}_j$ are known vectors of constants.Let the superscript δ denote the specific penalty family, e.g. $P_{\lambda,\alpha}^{PFL}(\boldsymbol{\beta})$ denotes the pairwise fused lasso penalty. The penalty proposed by Fan and Li (2001) has the special structure $P_{\lambda}^{\delta}(\boldsymbol{\beta}) = \sum_{j=1}^{p} p_{\lambda}(|\beta_j|)$. Since for that structure the vectors $\boldsymbol{a}_j$ have only one non-zero element it cannot be used to include interactions between the predictors. Hence, the approach of Fan and Li (2001) can be applied only to penalty families such as ridge and LASSO, but not to the fused lasso or pairwise fused lasso.

In 3.18 the sum of all J penalty functions $p_{\lambda_j,j}(|\boldsymbol{a}_j^T\boldsymbol{\beta}|)$ determines the penalty region, the number J of penalty functions is in general not equal to the number of regressors p. Furthermore, the type of the penalty function and the tuning parameter λ_j do not have to be the same for all J penalty functions. It is easily seen that the pairwise fused lasso penalty can be described by

$$P_{\lambda,\alpha}^{PFL}(\boldsymbol{\beta}) = \sum_{j=1}^{p+\binom{p}{2}} p_{\lambda,\alpha,j}(|\boldsymbol{a}_j^T\boldsymbol{\beta}|).$$

The first p penalty functions are

$$p_{\lambda,\alpha,j}(\cdot) = \lambda \cdot \alpha |\boldsymbol{a}_j^T \boldsymbol{\beta}|, \quad j = 1, \ldots, p,$$

where $\boldsymbol{a}_j = (0, \ldots, 0, 1, 0, \ldots, 0)^T$ is a p-dimensional vector of zeros apart from the jth position which is 1. This part represents the LASSO term of the PFL penalty. The remaining $\binom{p}{2}$ penalty functions are for the difference penalty term. They are given by

$$p_{\lambda,\alpha,j}(\cdot) = \lambda \left(1 - \alpha\right) |\boldsymbol{a}_j^T \boldsymbol{\beta}|, \quad j = p+1, \ldots, \tilde{p} + p$$

where the p-dimensional vectors $\boldsymbol{a}_j$ have the form $\boldsymbol{a}_j = (0, \ldots, -1, 0, \ldots, 1, 0, \ldots, 0)$. They describe the differences between two parameters.

An often applied principle in solving convex optimization problems is to use a quadratic approximation of the objective function. If the objective function is twice continuously differentiable iterative procedures of the Newton type are applied. Therefore, we need the gradient and the Hessian of the objective function. Since the first term of (3.17) is the negative log-likelihood, we can use the corresponding score function and expected Fisher information matrix. For the second term, one cannot proceed the same way because it includes L_1-norm terms. Therefore, Ulbricht (2010b) developed a quadratic approximation of the penalty term (3.18) which is shortly sketched in the following. Based on this approximation, Newton-type algorithms can be applied.

Let the following properties hold for all J penalty functions:

1. $p_{\lambda,j} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ with $p_{\lambda,j}(0) = 0$,
2. $p_{\lambda,j}$ is continuous and monotone in $|\boldsymbol{a}_j^T \boldsymbol{\beta}|$,
3. $p_{\lambda,j}$ is continuously differentiable for all $\boldsymbol{a}_j^T \boldsymbol{\beta} \neq 0$, i.e. $dp_{\lambda,j}\left(|\boldsymbol{a}_j^T \boldsymbol{\beta}|\right) / d|\boldsymbol{a}_j^T \boldsymbol{\beta}| \geq 0$ for all $\boldsymbol{a}_j^T \boldsymbol{\beta} \geq 0$.

Let $\boldsymbol{\beta}_{(k)}$ denote the approximation of the estimate $\widehat{\boldsymbol{\beta}}$ at the kth iteration of the LQA algorithm. Then the first order Taylor expansion of the jth penalty function in the neighborhood of $\boldsymbol{\beta}_{(k)}$ can be written as

$$p_{\lambda,j}\left(|\boldsymbol{a}_j^T \boldsymbol{\beta}|\right) \approx p_{\lambda,j}\left(|\boldsymbol{a}_j^T \boldsymbol{\beta}_{(k)}|\right) + \frac{1}{2} \frac{p'_{\lambda,j}\left(|\boldsymbol{a}_j^T \boldsymbol{\beta}_{(k)}|\right)}{\sqrt{\left(\boldsymbol{a}_j^T \boldsymbol{\beta}_{(k)}\right)^2 + c}} \left(\boldsymbol{\beta}^T \boldsymbol{a}_j \boldsymbol{a}_j^T \boldsymbol{\beta} - \boldsymbol{\beta}_{(k)}^T \boldsymbol{a}_j \boldsymbol{a}_j^T \boldsymbol{\beta}_{(k)}\right) \tag{3.19}$$

which is a quadratic function of $\boldsymbol{\beta}$. Thereby, $p'_{\lambda,j}\left(|\boldsymbol{a}_j^T \boldsymbol{\beta}_{(k)}|\right) = dp_{\lambda,j}\left(|\boldsymbol{a}_j^T \boldsymbol{\beta}|\right) / d|\boldsymbol{a}_j^T \boldsymbol{\beta}| \geq 0$ denotes the first derivative and c is a small positive integer (for our computations we choose $c = 10^{-8}$). Using matrix notation and summation over all J penalty functions the Taylor expansion is equivalent to

$$\sum_{j=1}^{J} p_{\lambda,j}\left(|\boldsymbol{a}_j^T \boldsymbol{\beta}|\right) \approx \sum_{j=1}^{J} p_{\lambda,j}\left(|\boldsymbol{a}_j^T \boldsymbol{\beta}_{(k)}|\right) + \frac{1}{2}\left(\boldsymbol{\beta}^T \boldsymbol{a}_\lambda \boldsymbol{\beta} - \boldsymbol{\beta}_{(k)}^T \boldsymbol{a}_\lambda \boldsymbol{\beta}_{(k)}\right), \tag{3.20}$$

with

$$\boldsymbol{a}_\lambda = \sum_{j=1}^{J} \frac{p'_{\lambda,j}\left(|\boldsymbol{a}_j^T\boldsymbol{\beta}_{(k)}|\right)}{\sqrt{\left(\boldsymbol{a}_j^T\boldsymbol{\beta}_{(k)}\right)^2 + c}}\boldsymbol{a}_j\boldsymbol{a}_j^T \tag{3.21}$$

which does not depend on the parameter vector $\boldsymbol{\beta}$. Since an intercept is included in the model, the penalty matrix is extended to

$$\boldsymbol{a}_\lambda^* = \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \boldsymbol{a}_\lambda \end{bmatrix}, \tag{3.22}$$

where $\mathbf{0}$ is the p-dimensional zero vector. Then, starting with the initial value $\boldsymbol{b}_{(0)}$, the update step of this Newton-type algorithm based on local quadratic approximations of the penalty term is

$$\boldsymbol{b}_{(k+1)} = \boldsymbol{b}_{(k)} - \left(\boldsymbol{F}(\boldsymbol{b}_{(k)}) + \boldsymbol{a}_\lambda^*\right)^{-1}\left\{-\boldsymbol{s}(\boldsymbol{b}_{(k)}) + \boldsymbol{a}_\lambda^*\boldsymbol{b}_{(k)}\right\}. \tag{3.23}$$

Corresponding to the log-likelihood $l(\boldsymbol{b})$, $\boldsymbol{s}(\boldsymbol{b})$ and $\boldsymbol{F}(\boldsymbol{b})$ denote the score function and Fisher information matrix, respectively. Iterations are carried out until the relative distance moved during the kth step is less or equal to a specified threshold ϵ, i.e. the termination condition is

$$\frac{\|\boldsymbol{b}_{(k+1)} - \boldsymbol{b}_{(k)}\|}{\|\boldsymbol{b}_{(k)}\|} \leq \epsilon, \quad \epsilon > 0. \tag{3.24}$$

3.3 Simulation Study

In this section we investigate the performance of the pairwise fused lasso and compare it to established procedures. All simulations are based on the generalized linear model

$$E(\boldsymbol{y}|\boldsymbol{X}) = h(\boldsymbol{X}\boldsymbol{\beta}_{true}),$$

where $h(.)$ is the canonical response function. 50 replications are performed for every simulation scenario and in each replication we generate a training, a validation and a test data set. The observation numbers of the corresponding data sets are denoted by $n_{train}/n_{vali}/n_{test}$. We use training data set to fit the models defined by the different tuning parameter(s). With $\mathcal{B}$ we denote the corresponding set of parameter vectors. By the minimizing the predictive deviance on the validation data set we determined the optimal tuning parameters and corresponding parameter vector $\widehat{\boldsymbol{\beta}}_0^{opt}$,

$$\widehat{\boldsymbol{\beta}}_0^{opt} = \text{argmin}_{\widehat{\boldsymbol{\beta}}_0 \in \mathcal{B}}\left\{-2(l(\boldsymbol{y}_{vali},\, h(\widehat{\beta}_0 + \boldsymbol{X}_{vali}\widehat{\boldsymbol{\beta}})) - l(\boldsymbol{y}_{vali},\, \boldsymbol{y}_{vali}))\right\},$$

where $\boldsymbol{y}_{vali}$ is the response vector of the validation data set and $h(\widehat{\beta}_0 + \boldsymbol{X}_{vali}\widehat{\boldsymbol{\beta}})$ is the estimated expectation based on $\boldsymbol{\beta}_0 \in \mathcal{B}$ and on the matrix of covariates of the validation data set $\boldsymbol{X}_{vali}$. Finally we use the test data set to evaluate the prediction by the predictive deviance on the test dataset,

$$\text{Dev}_{test} = -2(l(\boldsymbol{y}_{test},\, \widehat{\boldsymbol{y}}_{test}) - l(\boldsymbol{y}_{test},\, \boldsymbol{y}_{test})).$$

Here $\widehat{\boldsymbol{y}}_{test} = h(\widehat{\beta}_0^{opt} + \boldsymbol{X}_{vali}\widehat{\boldsymbol{\beta}}^{opt})$ is the estimated expectation based on the optimal parameter vector $\widehat{\boldsymbol{\beta}}_0^{opt}$ and the matrix of covariates of the test data set $\boldsymbol{X}_{test}$. With $\boldsymbol{y}_{test}$ we denote the observed response values of the test data set. Further we use the mean squared error between the true parameter vector and the estimate $\boldsymbol{\beta}^{opt}$

$$\text{MSE}_{\boldsymbol{\beta}} = \|\boldsymbol{\beta}_{true} - \widehat{\boldsymbol{\beta}}^{opt}\|^2$$

to measure the accuracy of the estimate of $\boldsymbol{\beta}$. The result are illustrated by boxplots. We do not show the outliers because of graphical reasons. As abbreviation for the differently weighted PFLs we will use the following:

- PFL denotes PFL penalty with all weights set to 1.
- PFL.ml denotes PFL penalty with ML-weights.
- PFL.cor denotes PFL penalty with correlation driven weights.
- PFL.pcor denotes PFL penalty with partial correlation driven weights.

We give the LASSO, EN, and the ML estimates for comparison. We calculate the LASSO and the EN estimates by the `LQA` routine. Since we investigate a regularization method with both variable selection and grouping property, we use the following simulation scenarios.

Normal Regression

Setting 1: This setting is specified by the parameter vector $\boldsymbol{\beta}_{true} = (3, 1.5, 0, 0, 0, 2, 0, 0)^T$ and standard error $\sigma = 3$. The correlation between the i-th and the j-th predictor is

$$\text{corr}(i, j) = 0.9^{|i-j|},\ \forall i, j \in \{1, \ldots, 8\}\,. \tag{3.25}$$

We chose 20/20/200 for the numbers of observations.

Setting 2: We consider $p = 20$ predictors and the parameter vector is structured into blocks:

$$\boldsymbol{\beta}_{true} = \big(\underbrace{0, \ldots, 0}_{5}, \underbrace{2, \ldots, 2}_{5}, \underbrace{0, \ldots, 0}_{5}, \underbrace{2, \ldots, 2}_{5}\big)^{\mathrm{T}}.$$

The standard error σ is 15 and the correlation between two predictors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is $\text{corr}(i, j) = 0.5$ for $i \neq j$. The numbers of observations are 50/50/400.

Setting 3: This setting has also $p = 20$ predictors and parameter vector is given by

$$\boldsymbol{\beta}_{true} = \big(5, 5, 5, 2, 2, 2, 10, 10, 10, \underbrace{0, \ldots, 0}_{11}\big)^T,$$

and the standard deviation of the error term is $\sigma = 15$. The design matrix $\boldsymbol{X}$ is specified in the following way. First we generate three auxiliary predictors $Z_j \sim N_n(\boldsymbol{0}, \boldsymbol{I})$, $j \in \{1, 2, 3\}$. With these predictors we generate

$$\begin{aligned} \boldsymbol{X}_i &= Z_1 + \tilde{\boldsymbol{\epsilon}}_i,\ i \in \{1, 2, 3\}, \\ \boldsymbol{X}_i &= Z_2 + \tilde{\boldsymbol{\epsilon}}_i,\ i \in \{4, 5, 6\}, \\ \boldsymbol{X}_i &= Z_3 + \tilde{\boldsymbol{\epsilon}}_i,\ i \in \{7, 8, 9\}, \end{aligned}$$

with $\tilde{\boldsymbol{\epsilon}}_i \sim N_n(\boldsymbol{0}, 0.01\boldsymbol{I})$, $i \in \{1, \ldots, 9\}$. The predictors $\boldsymbol{X}_i$, $i \in \{10, \ldots, 20\}$, are white noise, i.e. $\boldsymbol{X}_i \sim N_n(\boldsymbol{0}, \boldsymbol{I})$. Thus, within the first three blocks of 3 variables there is a quite high correlation, but there is no correlation between these blocks. The observation numbers are 50/50/400.

The results are summed up in Figure 3.2.

Binary Regression

In each simulation scenario the observation numbers $n_{train}/n_{vali}/n_{test}$ correspond to 100/100/400. The way of generating the matrices of covariates is equal to the normal case. But the predictor $\boldsymbol{\eta} = \boldsymbol{X}\boldsymbol{\beta}_{true}$ from the normal case is multiplied by a factor a in order to realize a appropriate domain for the logistic response function. We choose a so that the range of the predictor is approximately the interval $[-4, 4]$. Thus, for each setting we determine a factor a and multiply the true parameter vector from the normal case by this factor. The corresponding factor a and the modified parameter vector for each simulation setting are given by:

Setting 1:

$$a = 0.40 \quad \rightarrow \quad \boldsymbol{\beta}_{true} = (1.2, 0.6, 0, 0, 0, 0.8, 0, 0)^T.$$

Setting 2:

$$a = 0.15 \quad \rightarrow \quad \boldsymbol{\beta}_{true} = \big(\underbrace{0, \ldots, 0}_{5}, \underbrace{0.3, \ldots, 0.3}_{5}, \underbrace{0, \ldots, 0}_{5}, \underbrace{0.3, \ldots, 0.3}_{5}\big)^T.$$

Setting 3:

$$a = 0.10 \quad \rightarrow \quad \boldsymbol{\beta}_{true} = \big(0.75, 0.75, 0.75, 0.3, 0.3, 0.3, 1.5, 1.5, 1.5, \underbrace{0, \ldots, 0}_{11}\big)^T.$$

We use the canonical response function and so the response is modeled by $y_i = Bin(1, (1 + \exp(-\boldsymbol{x}_i^T\boldsymbol{\beta}_{true}))^{-1})$. In Figure 3.3 the results are illustrated by boxplots.

Poisson Regression

Analogously to the simulation study on binary responses, the predictor $\boldsymbol{\eta} = \boldsymbol{X}\boldsymbol{\beta}_{true}$ is multiplied by a factor a. Since the value range of the mean $\boldsymbol{\mu} = \exp(\boldsymbol{\eta})$ should be approximately in the interval $[0, 8]$, we determine for each setting the corresponding factor a. We

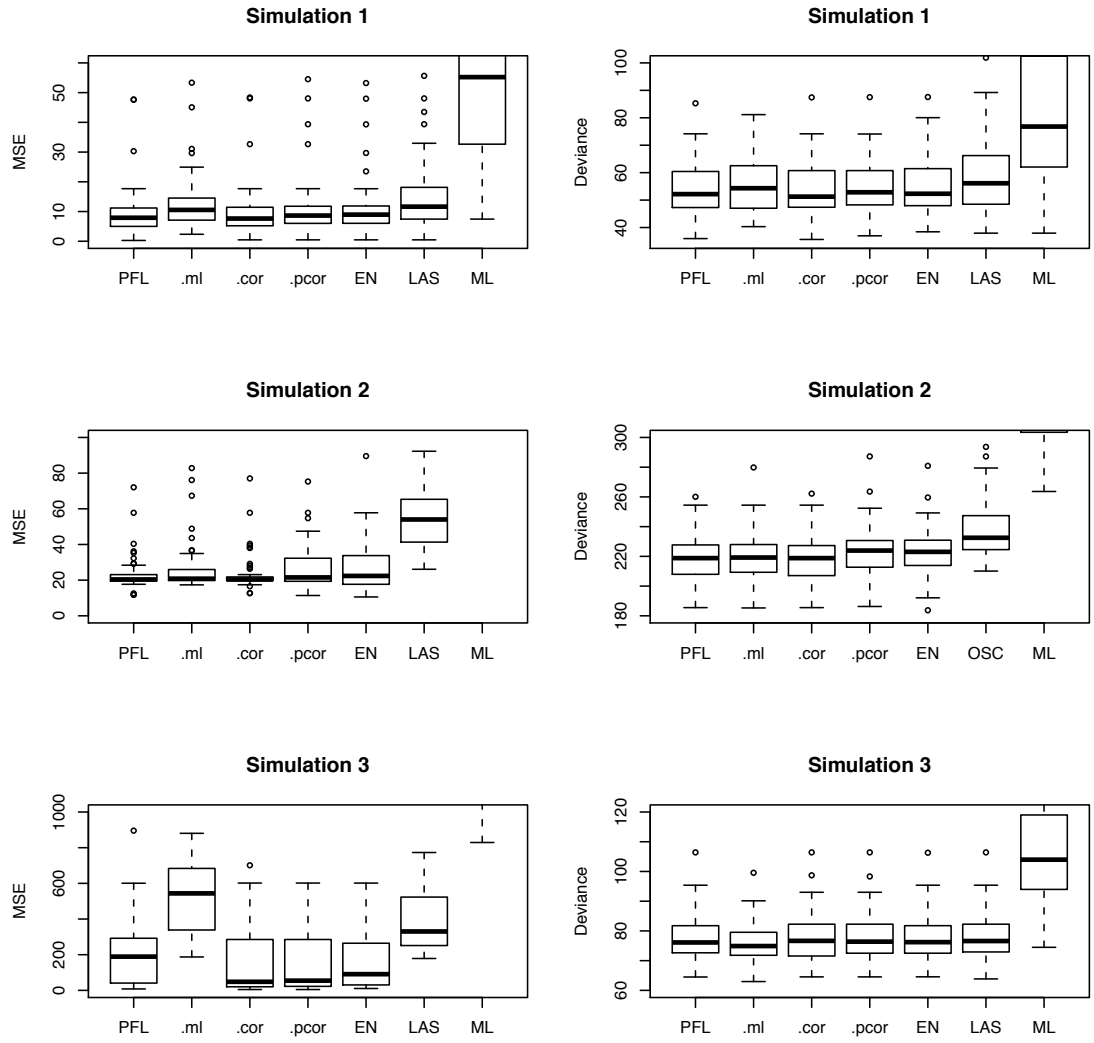

FIGURE 3.2: *Boxplots of the* $\text{MSE}_{\boldsymbol{\beta}}$ *and* Dev_{test} *for simulations with normal distributed response.*

model the response by $y_i = Pois(\exp(\boldsymbol{x}_i^T \boldsymbol{\beta}_{true}))$. The modified parameter vectors and the factor a determine the settings:

Setting 1:

$$a = 0.15 \quad \rightarrow \quad \boldsymbol{\beta}_{true} = (0.45, 0.225, 0, 0, 0, 0.3, 0, 0)^T$$

Setting 2:

$$a = 0.05 \quad \rightarrow \quad \boldsymbol{\beta}_{true} = \big(\underbrace{0, \ldots, 0}_{5}, \underbrace{0.1, \ldots, 0.1}_{5}, \underbrace{0, \ldots, 0}_{5}, \underbrace{0.3, \ldots, 0.3}_{5}\big)^T$$

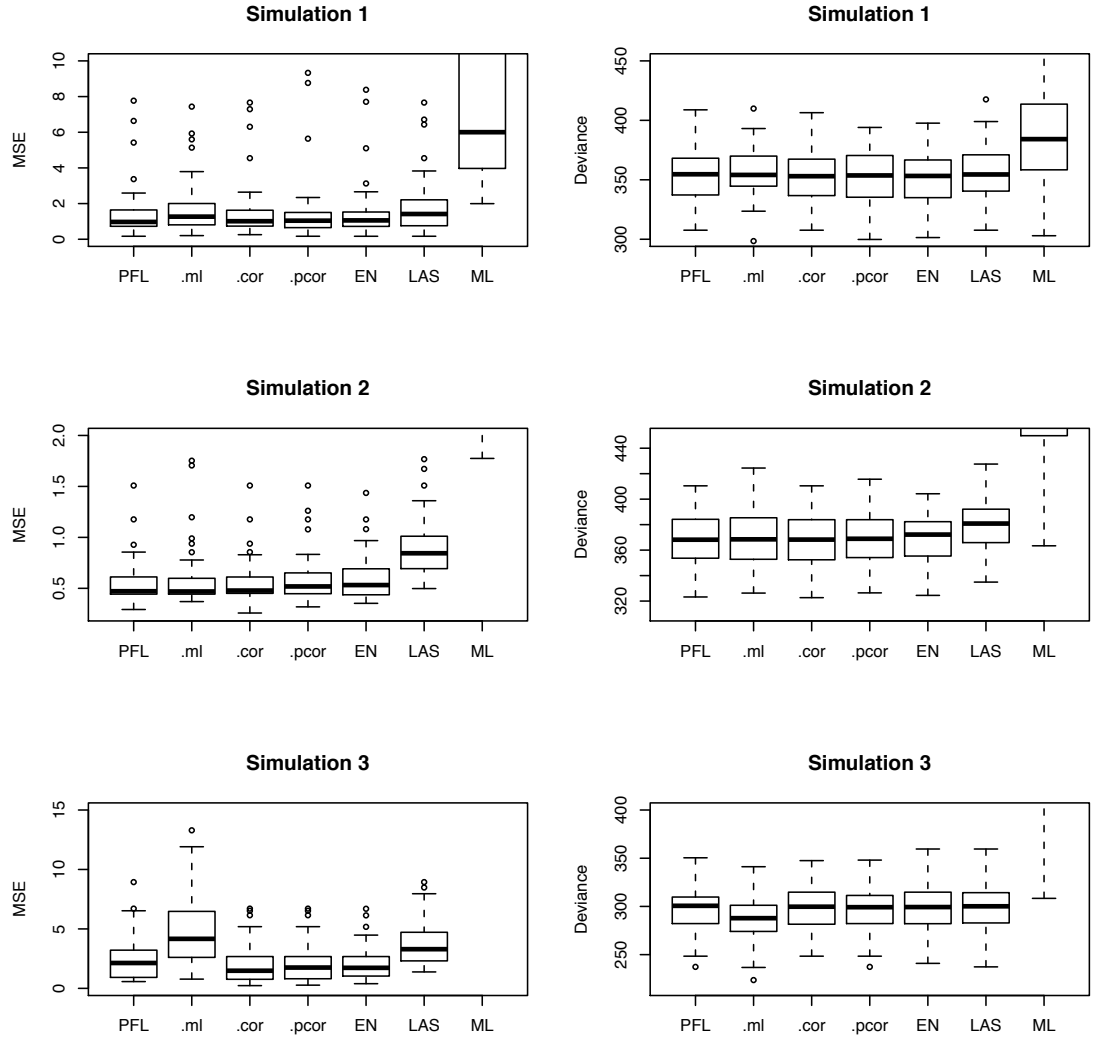

FIGURE 3.3: *Boxplots of the* $\text{MSE}_{\boldsymbol{\beta}}$ *and* Dev_{test} *for simulations with binomial distributed response.*

Setting 3:

$$a = 0.03 \quad \rightarrow \quad \boldsymbol{\beta}_{true} = \big(0.15, 0.15, 0.15, 0.06, 0.06, 0.06, 0.3, 0.3, 0.3, \underbrace{0, \ldots, 0}_{11}\big)^T$$

Figure 3.4 sums up the result by boxplots.

Summing Up the Result

The results of the simulation studies are summarized in Table 3.1. It is seen that the PFL is competitive in terms of the predictive test deviance (Dev_{test}) and the $\text{MSE}_{\boldsymbol{\beta}}$. The simulation study gives no clear indication which weights are best. Both performance measure-

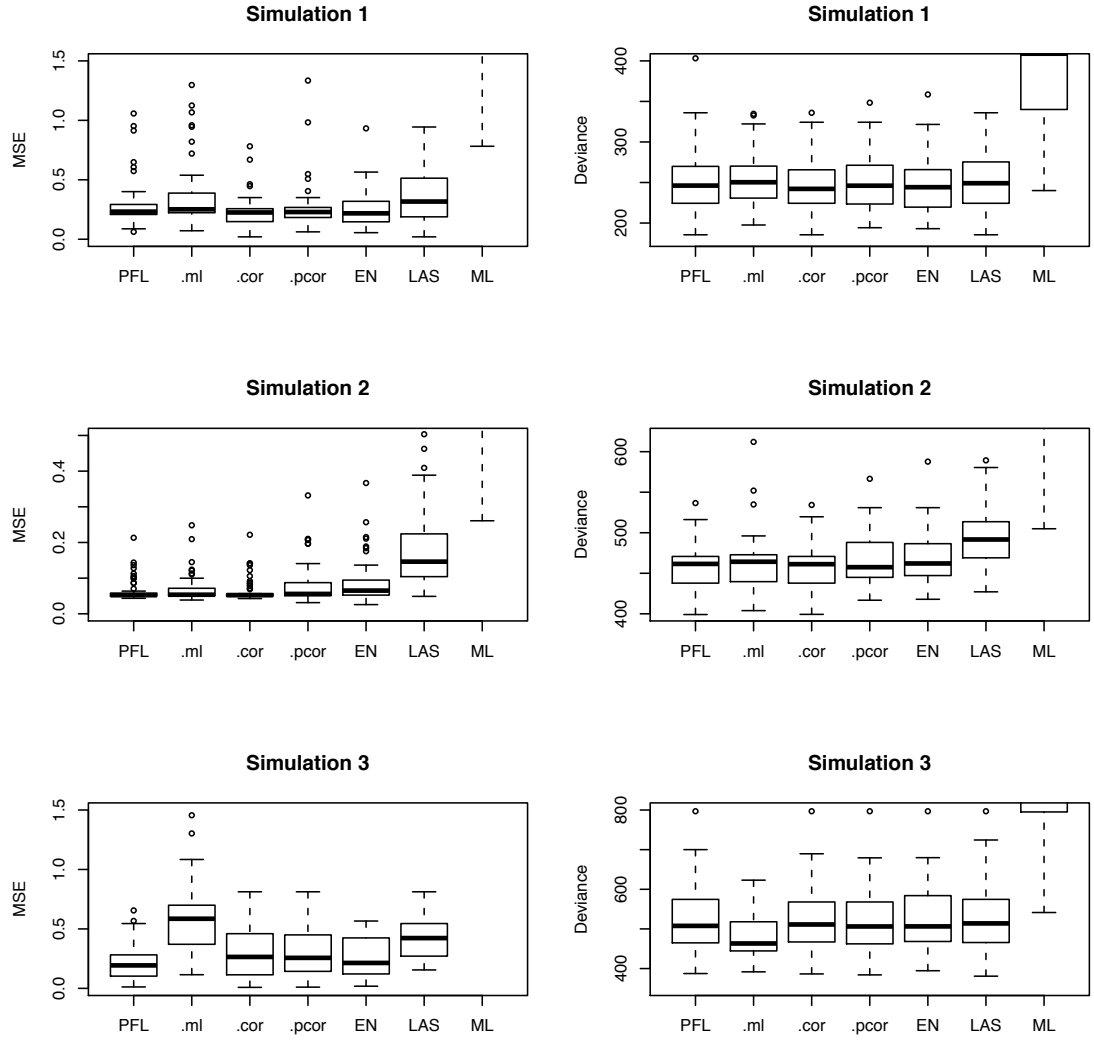

FIGURE 3.4: *Boxplots of the* $\text{MSE}_{\boldsymbol{\beta}}$ *and* Dev_{test} *for simulations with Poisson distributed response.*

ments for the correlation based weights are quite similar. Across all settings the correlation based weights seem to perform quite well. In general, apart from the ML based estimate, the PFL penalties distinctly outperform the LASSO and are strong competitors for the elastic net. The pairwise penalization seems to be an appropriate way for improving the performance of estimates. But the methods based on ML weights are strongly influenced by the instability of the ML estimate. In ill conditioned cases it can be an appropriate way to replace the MLE by a regularized estimate which does not select variables like the ridge estimator. It is remarkable that in contrast to the elastic net the PFL penalty enforces identical coefficients for "similar" variables, where the meaning of "similar" is specified by the chosen weights.

		PFL	PFL.ml	PFL.cor	PFL.pcor	EN	LASSO	ML
Normal distribution								
Setting 1	$\text{MSE}_{\boldsymbol{\beta}}$	7.90	10.54	7.64	8.64	8.95	11.64	55.22
		(0.88)	(0.68)	(0.94)	(0.61)	(0.57)	(1.83)	(8.13)
	Dev_{test}	52.17	54.33	51.25	52.85	52.32	56.13	76.79
		(2.05)	(3.35)	(1.90)	(2.31)	(2.68)	(2.86)	(4.62)
Setting 2	$\text{MSE}_{\boldsymbol{\beta}}$	20.39	20.82	20.35	21.53	22.37	54.01	284.16
		(0.25)	(0.52)	(0.20)	(1.50)	(1.57)	(3.77)	(22.11)
	Dev_{test}	218.82	219.21	218.82	223.90	223.04	232.50	336.00
		(3.01)	(2.66)	(2.82)	(2.52)	(2.05)	(3.05)	(12.71)
Setting 3	$\text{MSE}_{\boldsymbol{\beta}}$	189.15	543.81	48.07	54.40	90.79	330.20	4057.24
		(44.89)	(51.63)	(76.95)	(70.67)	(58.76)	(26.06)	(315.11)
	Dev_{test}	76.12	74.90	76.66	76.39	76.22	76.60	103.97
		(1.37)	(1.00)	(1.24)	(1.00)	(1.34)	(1.30)	(2.77)
Binomial distribution								
Setting 1	$\text{MSE}_{\boldsymbol{\beta}}$	0.97	1.27	1.01	1.0404	1.06	1.42	6.00
		(0.12)	(0.13)	(0.11)	(0.11)	(0.14)	(0.15)	(1.23)
	Dev_{test}	354.66	354.11	353.04	353.66	353.24	354.49	384.20
		(4.50)	(3.31)	(4.53)	(4.54)	(4.55)	(5.38)	(4.34)
Setting 2	$\text{MSE}_{\boldsymbol{\beta}}$	0.47	0.47	0.48	0.52	0.53	0.84	8.46
		(0.01)	(0.01)	(0.01)	(0.02)	(0.03)	(0.05)	(1.39)
	Dev_{test}	368.18	368.46	368.26	368.91	372.20	380.85	528.39
		(2.22)	(3.05)	(0.99)	(2.60)	(3.51)	(2.97)	(40.19)
Setting 3	$\text{MSE}_{\boldsymbol{\beta}}$	2.13	4.17	1.48	1.75	1.73	3.30	399.04
		(0.33)	(0.26)	(0.43)	(0.35)	(0.24)	(0.44)	(100.04)
	Dev_{test}	300.64	287.81	299.71	299.21	299.34	300.14	544.94
		(2.36)	(4.36)	(3.63)	(2.99)	(3.71)	(3.66)	(51.69)
Poisson distribution								
Setting 1	$\text{MSE}_{\boldsymbol{\beta}}$	0.23	0.25	0.23	0.23	0.22	0.32	5.70
		(0.01)	(0.02)	(0.01)	(0.01)	(0.02)	(0.05)	(1.06)
	Dev_{test}	246.19	250.30	242.14	246.06	244.20	249.11	408.90
		(6.44)	(6.50)	(5.40)	(6.66)	(6.88)	(6.05)	(55.29)
Setting 2	$\text{MSE}_{\boldsymbol{\beta}}$	0.05	0.05	0.05	0.06	0.07	0.15	1.54
		(0.00)	(0.00)	(0.00)	(0.00)	(0.01)	(0.02)	(0.14)
	Dev_{test}	461.56	464.22	461.23	457.51	462.09	491.67	929.31
		(4.53)	(4.15)	(3.08)	(7.00)	(6.04)	(5.59)	(61.26)
Setting 3	$\text{MSE}_{\boldsymbol{\beta}}$	0.19	0.59	0.26	0.26	0.21	0.42	20.19
		(0.03)	(0.04)	(0.03)	(0.03)	(0.04)	(0.05)	(2.58)
	Dev_{test}	507.66	463.25	511.19	506.18	506.36	513.92	1061.44
		(12.33)	(8.10)	(18.07)	(15.28)	(18.65)	(19.69)	(63.48)

TABLE 3.1: *Results of the simulation scenarios.*

3.4 Data Example

In this section we give two real data examples. One for the Binomial case and one for the normal case. In both cases we split the data set 50 times in two parts. First the training data set with n_{train} observations and second the test data set with n_{test} observations. We use the training data set to learn the model by a 5-fold cross validation. The model is determined by the parameter vector $\widehat{\boldsymbol{\beta}}_{train}$. The test data set is used for measuring the predictive deviance $\text{Dev}_{test} = -2(l(\boldsymbol{y}_{test}, \widehat{\boldsymbol{y}}_{test}) - l(\boldsymbol{y}_{test}, \boldsymbol{y}_{test})$, where $l(.,.)$ denotes the log likelihood function and $\widehat{\boldsymbol{y}}_{test} = h((\mathbf{1}, \boldsymbol{X}_{test})\boldsymbol{\beta}_{train})$ is the modeled expectation for the test data set.

Biopsy Data Set

The Biopsy Data Set is from the `R`-package `MASS` from Venables and Ripley (2002). It contains 699 observations and 9 covariates. We exclude the 16 observations with missing values. The covariates are whole-number scores between 0 and 10 about cell properties. Their description is given in Table 3.2.

Number	Explanation
1	clump thickness
2	uniformity of cell size
3	uniformity of cell shape
4	marginal adhesion
5	single epithelial cell size
6	bare nuclei
7	bland chromatin
8	normal nucleoli
9	mitoses

TABLE 3.2: *Explanation of the covariates of the Biopsy Data Set*

The response contains two classes of breast cancer "benign" or "malignant" and so we fit a logistic regression model. For n_{train} we choose $n_{train} = 400$. We give the predictive deviance in Figure 3.5 and in Table 3.3. In Figure 3.6 the estimates are shown.

PFL	PFL.ml	PFL.cor	PFL.pcor	EN	LASSO	ML
49.2292	49.6492	49.4307	48.18492	48.6917	49.4356	51.5290
(10.8875)	(10.9686)	(11.3377)	(10.7444)	(8.8604)	(11.0634)	(27.7673)

TABLE 3.3: *The median of predictive deviance on test data for the Biopsy Data Set. We give bootstrap variance of the medians in brackets. The bootstrapped variance is based on $B = 500$ bootstrap samples.*

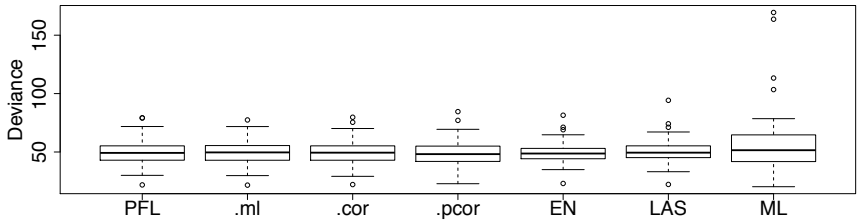

FIGURE 3.5: *Boxplots of the predictive deviance* Dev_{test} *for the Biopsy Data Set*

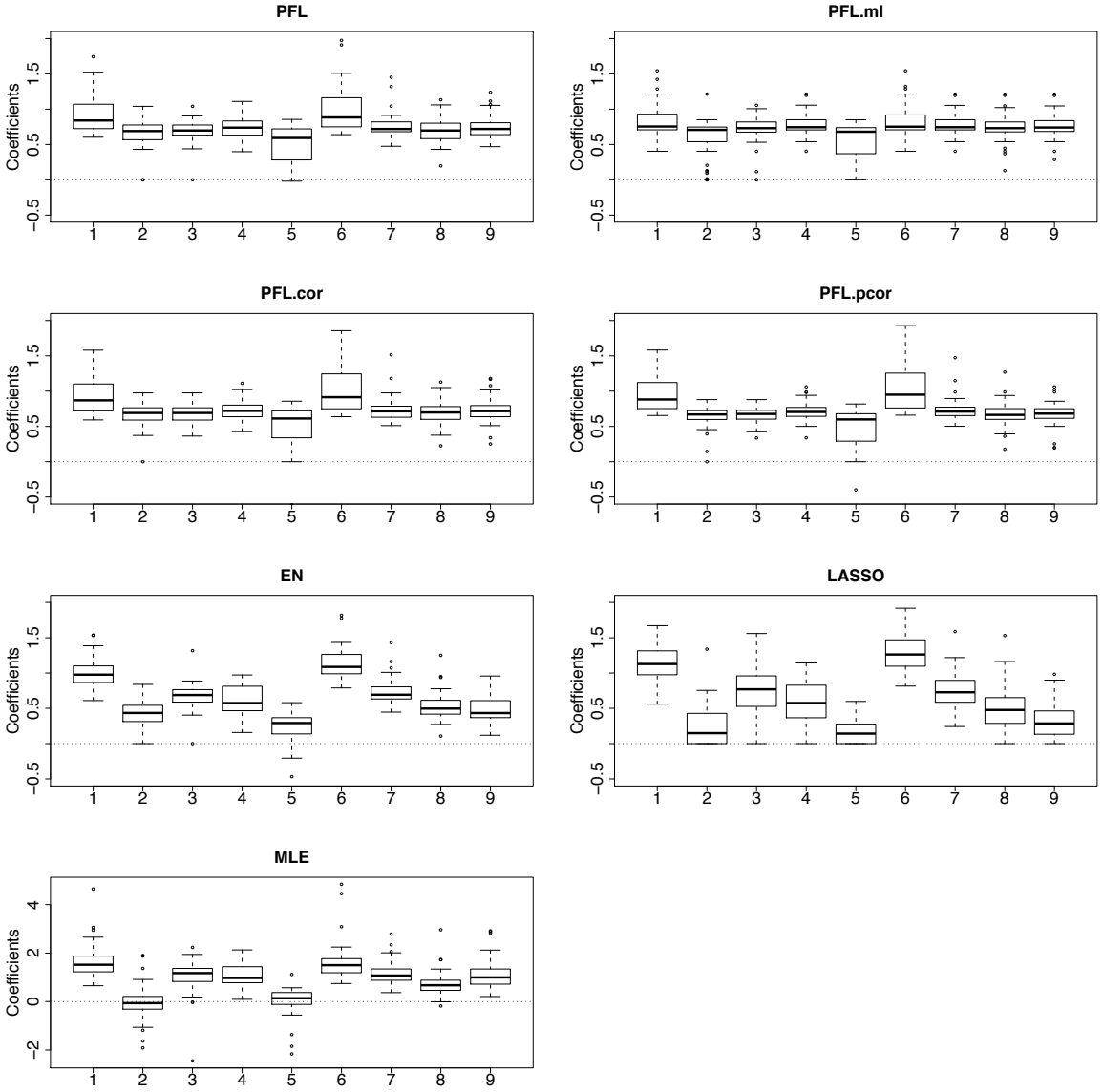

FIGURE 3.6: *Boxplots of the coefficient estimates for the Biopsy Data Set*

In contrast to the Elastic Net estimates the grouping property of the PFL is stronger. Further it is remarkable that different models have similar predictive deviances. In some replications the MLE leads to perfect discrimination of the response groups and the procedures gives warning.

Bones Data Set

This study aims at estimating the age at date of death of 87 persons. Since we choose the normal model. The underlying data set contains 20 covariates. These covariates are bones characteristics and the gender of the deceased person and given in Table 3.4. Some of the

Number	Explanation
1	gender
2	size of an compact bone
3	femur class
4	type I osteon
5	type II osteon
6	osteon fragments
7	osteon population density
8	Haverssche canals
9	non Haverssche canals
10	Volkmannsche canals
11	resorption lacuna
12	percentage of resorption lacuna
13	percentage of general lamellae
14	percentage of osteonal bones
15	percentage of fragmental bones
16	surface of an osteon
17	surface of a resorption lacuna
18	quotient of the surface of a resorption lacuna and the surface of an osteon
19	activation frequency
20	bone formation rate

TABLE 3.4: *Explanation of the covariates of the Bones Data Set*

covariates are highly correlated, i.e. $\rho_{ij} \approx 0.9$. The data based on the Basel-Kollektiv and are provided by Stefanie Doppler from the Department of Anthropology and Human Genetics of the Ludwig-Maximilians-Universität (see Doppler, 2008). It is a collection of excavation of a graveyard. So the age at date of death of the persons is known and the skeletons are in a quite good conditions.

We randomly split the data set 25-times into a test data set with 60 observations and a test data set with 27. The predictive deviance on test data and for each method are given in Table 3.5 and illustrated in Figure 3.7. We give standardized estimates by

PFL	PFL.ml	PFL.cor	PFL.pcor	EN	LASSO	ML
3.1969	3.1085	3.2367	3.1873	3.1432	3.1873	4.4276
(0.9178)	(0.7589)	(0.9112)	(0.8401)	(0.8366)	(0.9212)	(0.8909)

TABLE 3.5: *The median of predictive deviance on test data for the Bones Data Set. We give bootstrap variance of the medians in brackets. The bootstrapped variance is based on* $B = 500$ *bootstrap samples.*

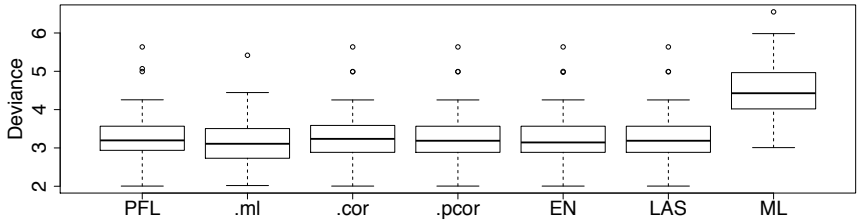

FIGURE 3.7: *Boxplots of the predictive deviance for the Bones Data Set*

boxplots of the coefficient estimates in Figure 3.8. Because for standardized covariates the grouping effect appears. The MLE-weighted PFL tends to group the covariates 12,13, and 14. It has the best predictive deviance. In general it is remarkable that variable selection dominates clustering in the other cases. Apart from the MLE and the MLE-weighted PFL the results are very similar. Although the MLE is quite ill conditioned the MLE-weighted PFL outperforms the remaining procedures.

3.5 Concluding Remarks

We proposed a regularization method that enforces the grouping property by including pairwise differences of coefficients in the penalty term. It works for linear as well as generalized linear models and is strong a competitor for the lasso and the elastic net. Although it uses fusion methodology it does not assume that a metric on predictors is available. Therefore it can used for common regression problems.

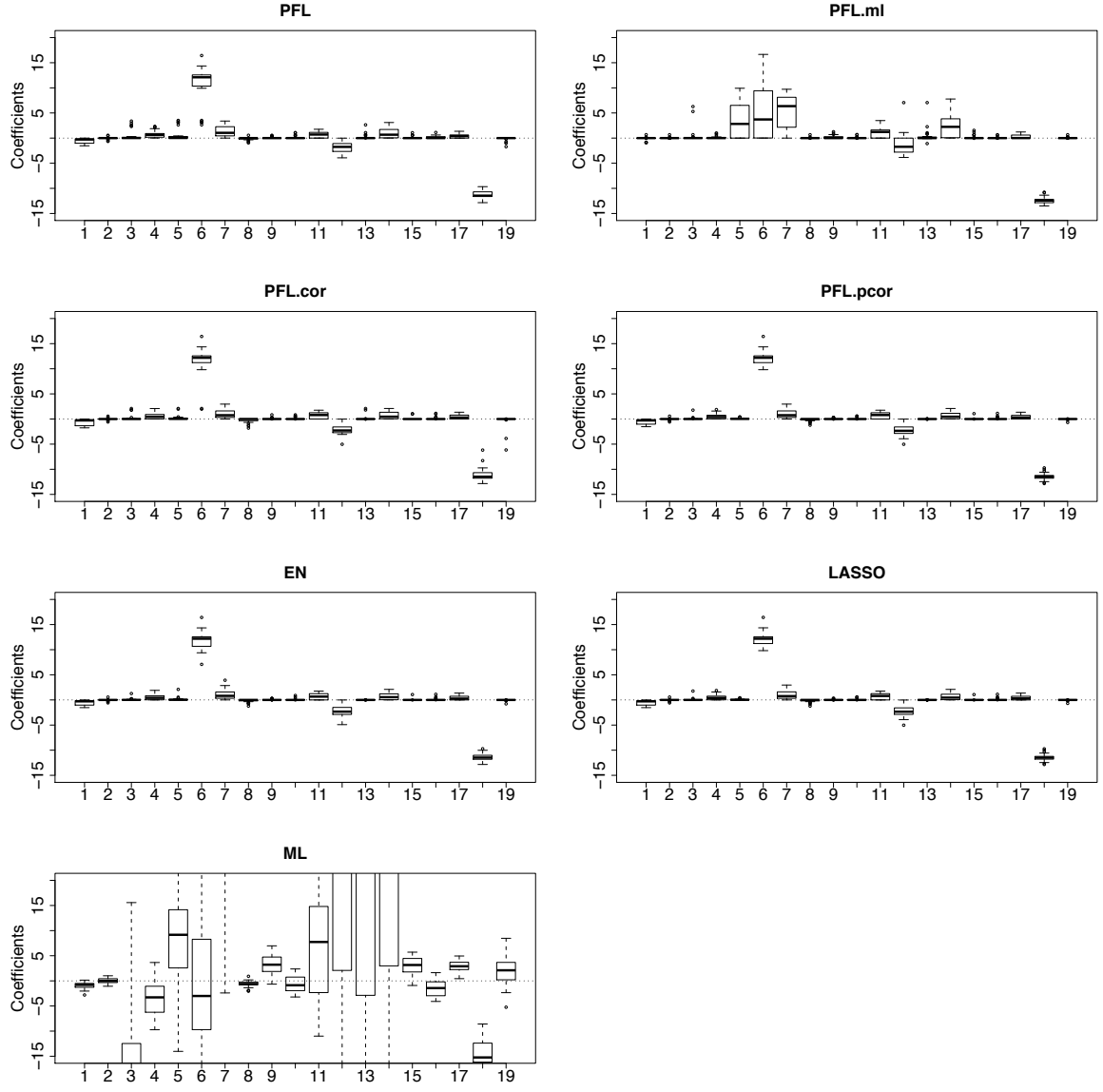

FIGURE 3.8: *Boxplots of the predictors for the Bones Data Set*

Chapter 4

The OSCAR Penalty for Generalized Linear Models

The Octagonal Selection and Clustering Algorithm in Regression (OSCAR) proposed by Bondell and Reich (2008) has the attractive feature that highly correlated predictors can obtain exactly the same coefficient yielding clustering of predictors. Estimation methods are available for linear regression models. It is shown how the OSCAR penalty can be used within the framework of generalized linear models. An algorithm that solves the corresponding maximization problem is given. The estimation method is investigated in a simulation study and the usefulness is demonstrated by an example from water engineering. This chapter is based on Petry and Tutz (2011a).

4.1 Introduction

Within the last decades various regularization techniques for generalized linear models (GLMs) have been developed. Most methods aim at stabilizing estimates and finding simpler models. In particular variable selection has been a major topic. One of the oldest methods is ridge regression, which has been proposed by Hoerl and Kennard (1970). In ridge regression the parameter space is restricted to a p-sphere around the origin $\sum_{j=1}^{p} \beta_j^2 \leq t$, $t \geq 0$. Another popular shrinkage methods is the LASSO for **L**east **A**bsolute **S**hrinkage and **S**election **O**perator (Tibshirani, 1996), where the parameter space is restricted to a p-crosspolytope $\sum_{j=1}^{p} |\beta_j| \leq t$, $t \geq 0$. The restriction induces shrinkage and variables selection. In general, restricted parameter spaces are called penalty regions. For many penalty regions the problem can be transformed into a penalized likelihood problem by adding a penalty term to the log-likelihood. For ridge regression the penalty term is $\lambda \sum_{j=1}^{p} \beta_j^2$ and for the LASSO it is $\lambda \sum_{j=1}^{p} |\beta_j|$, with $\lambda \geq 0$ in both cases. A combination of the ridge and the LASSO uses $\lambda_1 \sum_{j=1}^{p} \beta_j^2 + \lambda_2 \sum_{j=1}^{p} |\beta_j|$. It is well known as the elastic net (Zou and Hastie, 2005).

Zou and Hastie (2005) showed that variable selection leads to unsatisfying results in the case of multicollinearity, that is, if some of the covariates are highly correlated. Then

procedures like the LASSO tend to include only a few covariates from a group of the highly correlated covariates. They show that for the elastic net a grouping property holds, which means that the estimated parameters of highly correlated covariates are similar up to sign. An alternative penalty region that enforces grouping of variables was proposed by Bondell and Reich (2008) under the name OSCAR for **O**ctagonal **S**election and **C**lustering **A**lgorithm in **R**egression. For LASSO and the elastic net (EN) several methods have been proposed to solve the penalized log-likelihood problem in generalized linear models (GLMs); (see Park and Hastie, 2007b; Goeman, 2010a; Friedman et al., 2010a). For OSCAR it seems that algorithms are available only for the linear model. In the following estimation methods for OSCAR are proposed that work within the more general GLM framework.

In Section 4.2 we give a short overview on GLMs. In Section 4.3 the OSCAR penalty region is discussed. In Section 4.4 we use the results of Section 4.3 and present an algorithm for estimating the corresponding restricted regression problem based on the active set method. A simulation study is presented in Section 4.5, which uses settings that are similar to the settings used by Bondell and Reich (2008). A real data example with water engineering background is given in Section 4.6.

4.2 Generalized Linear models

We consider data $(\boldsymbol{y}, \boldsymbol{X})$ where $\boldsymbol{y} = (y_1, ..., y_n)^T$ is the response and $\boldsymbol{X}$ is the $(n \times p)$ matrix of explanatory variables that contains n observations $\boldsymbol{x}_i^T = (x_{i1}, ..., x_{ip})$, $i = 1, ..., n$. In GLMs (McCullagh and Nelder, 1983) it is assumed that the distribution of $y_i|\boldsymbol{x}_i$ is from a simple exponential family

$$f(y_i|\theta_i, \phi) = \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{\phi} + c(y_i, \phi)\right\}, \tag{4.1}$$

where θ_i is the natural parameter and ϕ is a dispersion parameter; $b(.)$ and $c(.)$ are specific functions corresponding to the type of the family. In addition, it is assumed that the observations are (conditionally) independent. For given data the conditional expectation of $y_i|\boldsymbol{x}_i$, $\mu_i = E(y_i|\boldsymbol{x}_i)$, is modeled by

$$g(\mu_i) = \eta_i \quad \text{or} \quad \mu_i = h(\eta_i),$$

where $\eta_i = \beta_0 + \boldsymbol{x}_i^T\boldsymbol{\beta}$ is the linear predictor, $g(.)$ is the link function and $h(.) = g^{-1}(.)$ is the response function. Let $\boldsymbol{\beta}_0 = (\beta_0, \boldsymbol{\beta}^T)^T$ denote the parameter vector that includes the intercept. Then the corresponding design matrix is $\boldsymbol{Z} = (\boldsymbol{1}_n, \boldsymbol{X})$ and the linear predictor is $\boldsymbol{\eta} = \boldsymbol{Z}\boldsymbol{\beta}_0$. The maximum likelihood estimate (MLE) is given by

$$\widehat{\boldsymbol{\beta}}_0 = \text{argmax}_{\boldsymbol{\beta}_0}\left\{\sum_{i=1}^{n} l_i(\boldsymbol{\beta}_0)\right\}$$

where $l_i(\boldsymbol{\beta}_0)$ is the likelihood function of the ith observation. The maximum likelihood problem can be iteratively solved by

$$\widehat{\boldsymbol{\beta}}_0^{(l+1)} = \text{argmin}_{\boldsymbol{\beta}_0}\left\{\boldsymbol{\beta}_0^T\boldsymbol{Z}^T\widehat{\boldsymbol{W}}^{(l)}\boldsymbol{Z}\boldsymbol{\beta}_0 - 2\boldsymbol{\beta}_0^T\boldsymbol{Z}^T\widehat{\boldsymbol{W}}^{(l)}\widetilde{\boldsymbol{y}}^{(l)}\right\}, \tag{4.2}$$

where

$$\widetilde{\boldsymbol{y}}^{(l)} = \boldsymbol{Z}\widehat{\boldsymbol{\beta}}_0^{(l)} + (\widehat{\boldsymbol{D}}^{(l)})^{-1}(\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^{(l)})$$

is the working response vector,

$$\widehat{\boldsymbol{W}}^{(l)} = (\widehat{\boldsymbol{D}}^{(l)})^T(\widehat{\boldsymbol{\Sigma}}^{(l)})^{-1}\widehat{\boldsymbol{D}}^{(l)}$$

is the weight matrix with the derivative matrix of the response function,

$$\widehat{\boldsymbol{D}}^{(l)} = \operatorname{diag}\left\{\frac{\partial h(\widehat{\eta}_i^{(l)})}{\partial \eta}\right\}_{i=1}^{n},$$

and the matrix of variances

$$\widehat{\boldsymbol{\Sigma}}^{(l)} = \operatorname{diag}\left\{\phi V(h(\widehat{\eta}_i^{(l)}))\right\}_{i=1}^{n},$$

all of them evaluated at the previous step. $V(.)$ is the variance function, which is determined by the distributional assumption and $\widehat{\boldsymbol{\mu}}^{(l)}$ is the estimated prediction of the previous step. The update is repeated until $\|\widehat{\boldsymbol{\beta}}_0^{(l+1)} - \widehat{\boldsymbol{\beta}}_0^{(l)}\|/\|\widehat{\boldsymbol{\beta}}_0^{(l)}\| < \varepsilon$ for small ε. The re-weighted least square estimates

$$\widehat{\boldsymbol{\beta}}_0^{(l+1)} = \left(\boldsymbol{Z}^T\widehat{\boldsymbol{W}}^{(l)}\boldsymbol{Z}\right)^{-1}\boldsymbol{Z}^T\widehat{\boldsymbol{W}}^{(l)}\widetilde{\boldsymbol{y}}^{(l)}$$

is also known as Fisher scoring. The algorithm we will present uses a constrained Fisher scoring combined with the active set method that uses the specific structure of the OSCAR penalty.

4.3 The OSCAR Penalty Region

In the following we consider standardized covariates, that is, $\sum_{i=1}^n x_{ij} = 0$ and $(n-1)^{-1}\sum_{i=1}^n x_{ij}^2 = 1$. When Bondell and Reich (2008) introduced the OSCAR for the normal linear regression they also centered the responses by using $\sum_{i=1}^n y_i = 0$. If all covariates and the response are centered no intercept has to be estimated. Then the OSCAR can be given as the constrained least-squares problem

$$\widehat{\boldsymbol{\beta}} = \operatorname{argmax}_{\boldsymbol{\beta}}\left\{\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2, \text{ s.t. } \boldsymbol{\beta} \in \mathcal{O}_{c,t}(\boldsymbol{\beta})\right\}, \tag{4.3}$$

with OSCAR penalty region given by

$$\mathcal{O}_{c,t}(\boldsymbol{\beta}) = \left\{\boldsymbol{\beta} : \sum_{j=1}^p |\beta_j| + c\sum_{1\le j<k\le p}\max\left\{|\beta_j|, |\beta_k|\right\} \le t\right\}. \tag{4.4}$$

The first sum $\sum_{j=1}^p |\beta_j|$ is the LASSO penalty which induces variable selection. The second sum $c\sum_{1\le j<k\le p}\max\left\{|\beta_j|, |\beta_k|\right\}$ accounts for clustering of similar variables. With $c \ge 0$ and $t > 0$ an equivalent form of the OSCAR penalty (4.4) is

$$\mathcal{O}_{c,t}(\boldsymbol{\beta}) = \left\{\boldsymbol{\beta} : \sum_{j=1}^p \left\{c(j-1)+1\right\}|\beta_{(j)}| \le t\right\}, \tag{4.5}$$

where $|\beta_{(1)}| \leq |\beta_{(2)}| \leq ... \leq |\beta_{(p)}|$ and $|\beta_{(j)}|$ denotes the jth largest component of $|\boldsymbol{\beta}| = (|\beta_1|, ..., |\beta_p|)^T$. The parameter c controls the clustering and t the amount of shrinkage. Bondell and Reich (2008) gave a `MatLab`-code at `http://www4.stat.ncsu.edu/~bondell/software.html` which solves the least square problem under constraints

$$\mathcal{O}_{\alpha,t}(\boldsymbol{\beta}) = \left\{ \boldsymbol{\beta} : (1-\alpha) \sum_{j=1}^{p} |\beta_j| + \alpha \sum_{1 \leq j < k \leq p} \max\{|\beta_j|, |\beta_k|\} \leq t \right\} \tag{4.6}$$

$$= \left\{ \boldsymbol{\beta} : \sum_{j=1}^{p} \{\alpha(j-1) + (1-\alpha)\} |\beta_{(j)}| \leq t \right\} \tag{4.7}$$

where $\alpha \in [0, 1]$ and $t > 0$. If $\alpha = 0$, respectively $c = 0$, the OSCAR is equivalent to the LASSO. For appropriate values of c, α and t the penalty regions (4.4) and (4.6) are equivalent. In the following we use $\mathcal{O}_{\alpha,t}(\boldsymbol{\beta})$ from (4.6) and (4.7).

In contrast to the Elastic Net penalty the OSCAR enforces that parameters obtain the same value. Bondell and Reich (2008) derived a relationship between the clustering of covariates (which obtain the same value) and their correlation. The word octagonal in OSCAR is motivated by the geometry of the penalty region. The projection of the penalty region into each β_i-β_j-plane is an octagon. The octagonal shape accounts for the estimation of identical parameters as well as variable selection because the coordinates of the vertices have a very specific structure. In particular the absolute values of the coordinates of a vertex on the surface are equal or zero. So each convex combination of vertices on the surface describes an area with specific properties. If less than p vertices are convexly combined one obtains variable selection and/or clustering. For illustration, Figure 4.1 shows an OSCAR penalty region in $\mathbb{R}^3$.

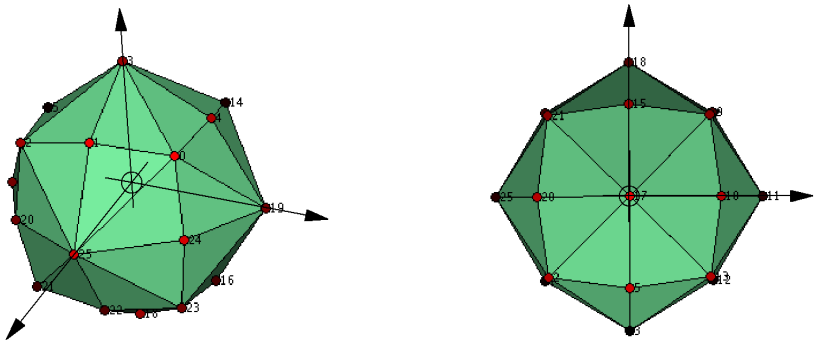

FIGURE 4.1: *OSCAR penalty region from two different perspectives. On the right it is the projection on a β_i-β_j-plane.*

In Petry and Tutz (2011b) it is shown that the OSCAR penalty is the intersection of $2^p \cdot p!$ halfspaces. So $\mathcal{O}_{\alpha,t}(\boldsymbol{\beta})$ can be rewritten into a system of inequations $\boldsymbol{A}\boldsymbol{\beta} \leq \boldsymbol{t}$ where $\boldsymbol{A}$ is the $(2^p \cdot p!) \times p$-dimensional matrix $\boldsymbol{A} = (\boldsymbol{a}_1, ..., \boldsymbol{a}_{2^p \cdot p!})^T$ that contains the normal

vectors $\boldsymbol{a}_q$ of each generating hyperplane. Each normal vector $\boldsymbol{a}_q$ is characterized by two attributes:

1. The vector of signs of the components of the normal vector,

$$\text{sign}(\boldsymbol{a}_q) = (\text{sign}(a_{q1}),\ ...,\ \text{sign}(a_{qp}))^T$$

   This attribute is induced by the absolute value of the components of $\boldsymbol{\beta}$ (see (4.6) or (4.7)).

2. The vector of ranks of the absolute value of the components of the normal vector

$$\text{rank}(|\boldsymbol{a}_q|) = (\text{rank}(|a_{q1}|),\ ..,\ \text{rank}(|a_{qp}|))^T,$$

   which is a p-dimensional vector. Its jth entry is the position of a_{qj} in the order $|a_{q(1)}| \leq |a_{q(2)}| \leq ... \leq |a_{q(p)}|$ where $|a_{q(j)}|$ denotes the absolute value of the jth largest component of $|(|\boldsymbol{a}_q) = (|a_{q1}|,\ ...,\ |a_{qp}|)^T$. This attribute is induced by using the pairwise maximum norm in (4.6) or the ordered components like in (4.7) respectively.

Each row of $\boldsymbol{A}$ is given by signs and a permutation of the weights $\boldsymbol{w} = \{\alpha(j-1) + (1-\alpha) : j = 1,\ ...,\ p\}$ given in (4.7). Each half space refers to one constraint of the restricted optimization problem that can be written as

$$\boldsymbol{a}_q = (\alpha \cdot (\text{rank}(|\boldsymbol{a}_q|) - 1) + (1-\alpha))^T \,\text{diag}(\text{sign}(\boldsymbol{a}_q)) \leq t. \tag{4.8}$$

Already for small dimensional cases the dimension of $\boldsymbol{A}$ becomes very large, for example, if $p = 5$ the matrix $\boldsymbol{A}$ is 3840×5-dimensional.

4.4 The glmOSCAR Algorithm

For GLMs the least-squares problem (4.3) turns into the restricted maximum likelihood problem

$$\widehat{\boldsymbol{\beta}}_0 = \text{argmax}_{\boldsymbol{\beta}_0} \left\{ \sum_{i=1}^{n} l_i(\boldsymbol{\beta}_0),\ \text{s.t. } \boldsymbol{\beta}_0 \in \mathbb{R} \times \mathcal{O}_{\alpha,t}(\boldsymbol{\beta}) \right\},$$

where $l_i(.)$ is the log-likelihood of a GLM. In contrast to the linear normal regression, where responses are easily centered, now an unrestricted intercept has to be included. The new penalty region is $\mathbb{R} \times \mathcal{O}_{\alpha,t}$, which can be rewritten as an system of inequations

$$(\boldsymbol{0},\ \boldsymbol{A})\boldsymbol{\beta}_0 \leq \boldsymbol{t}, \tag{4.9}$$

where $\boldsymbol{\beta}_0 = (\beta_0,\ \boldsymbol{\beta}^T)^T$. The region (4.9) is an unbounded intersection of subspaces called polyhedron. Each row of (4.9) refers to one constraint. In general a constraint of a system of inequations is called active if the equal sign holds in the corresponding row of the system of inequations. If the equal sign holds the solution lies on the corresponding face

of the polyhedron. Only the active constraints have an influence on the solution. The remaining constraints are fulfilled but have no influence on the solution, and are called inactive constraints. Removing inactive constraints has no influence on the solution of the constrained log-likelihood problem. The solution is unique and each point in $\mathbb{R}^p$ can be represented by the intersection of p hyperplanes of dimension $p-1$. Therefore, the number of constraints can be reduced from $2^p \cdot p!$ to p. Only by numerical reasons sometimes in the algorithm more than p constraints are set active. Because of (4.8) and (4.9) for an given parameter vector $\boldsymbol{\beta}_0$ an active constraint from (4.9) has the following form

$$\boldsymbol{a}(\boldsymbol{\beta}_0)\boldsymbol{\beta}_0 = (0,\, ((1-\alpha)\cdot(\text{rank}(|\boldsymbol{\beta}|)-1)+\alpha)^T \,\text{diag}(\text{sign}(\boldsymbol{\beta})))\boldsymbol{\beta}_0 = t. \tag{4.10}$$

It is important that rank($|\boldsymbol{\beta}|$) is a p-dimensional vector where all elements of $\{1, 2, ..., p\}$ are used as entries. If some elements of $|\boldsymbol{\beta}|$ are equal the assembly of their ranks is arbitrary.

The following algorithm is an active set method combined with Fisher scoring. There are two parts.

AS (Active Set): This step accounts for the creation of the active set and is indexed by $^{(k)}$.

FS (Fisher Scoring): This step solves the restricted ML problem. It is indexed by $^{(l)}$ in analogy to (4.2). The constraints are given by the active set that is determined by the AS-step.

First we initialize $k = 0$ and choose an initial value $\widehat{\boldsymbol{\beta}}_0^{(0)}$, for instance, the MLE.

AS-step

We set k to $k+1$. With $\widehat{\boldsymbol{\beta}}_0^{(k-1)}$ we determine $\boldsymbol{a}(\widehat{\boldsymbol{\beta}}_0^{(k-1)}) = \boldsymbol{a}^{(k)}$ as given in (4.10). The new active constraint $\boldsymbol{a}^{(k)}$ is added as a new row to $(\boldsymbol{0},\, \boldsymbol{A})^{(k-1)}\boldsymbol{\beta}_0 \leq \boldsymbol{t}$ if $\boldsymbol{a}^{(k)}$ is not a row of $(\boldsymbol{0},\, \boldsymbol{A})^{(k-1)}$

$$\begin{pmatrix} (\boldsymbol{0},\, \boldsymbol{A})^{(k-1)} \\ \boldsymbol{a}^{(k)} \end{pmatrix} \boldsymbol{\beta}_0 = (\boldsymbol{0},\, \boldsymbol{A})^{(k)}\boldsymbol{\beta}_0 \leq \boldsymbol{t}. \tag{4.11}$$

Finally we remove all inactive constraints from $(\boldsymbol{0},\, \boldsymbol{A})^{(k)}\boldsymbol{\beta}_0 \leq \boldsymbol{t}$.

FS-step

We have to solve the constrained ML problem

$$\widehat{\boldsymbol{\beta}}_0^{(k)} = \text{argmin}_{\boldsymbol{\beta}_0} \left\{ -\sum_{i=1}^{n} l_i(\boldsymbol{\beta}_0),\ \text{s.t. } (\boldsymbol{0},\, \boldsymbol{A})^{(k)}\boldsymbol{\beta}_0 \leq \boldsymbol{t} \right\}, \tag{4.12}$$

which is a combination of the unconstrained least square problem (4.2) and the penalty region $(\boldsymbol{0},\, \boldsymbol{A})^{(k)}\boldsymbol{\beta}_0 \leq \boldsymbol{t}$ from the AS. For clarity we do not use double indexing. For solving

(4.12) we use the following constrained Fisher scoring

$$\begin{aligned}\widehat{\boldsymbol{\beta}}_0^{(l+1)} &= \operatorname{argmin}_{\boldsymbol{\beta}_0}\Big\{\boldsymbol{\beta}_0^T\boldsymbol{Z}^T\widehat{\boldsymbol{W}}^{(l)}\boldsymbol{Z}\boldsymbol{\beta}_0 - 2\boldsymbol{\beta}_0^T\boldsymbol{Z}^T\widehat{\boldsymbol{W}}^{(l)}\widetilde{\boldsymbol{y}}^{(l)}, \\ &\qquad \text{s.t. } (\boldsymbol{0},\, \boldsymbol{A})^{(k)}\boldsymbol{\beta}_0 \leq \boldsymbol{t}\Big\}.\end{aligned} \tag{4.13}$$

It is solved iteratively with the `quadprog` package from `R` (see Turlach, 2009). The constrained update (4.13) is repeated up to convergence $\delta_2 = \|\hat{\boldsymbol{\beta}}_0^{(l)} - \hat{\boldsymbol{\beta}}_0^{(l+1)}\|/\|\hat{\boldsymbol{\beta}}_0^{(l)}\| \leq \varepsilon$, for small ε. After convergence $\widehat{\boldsymbol{\beta}}_0^{(l+1)}$ is the solution of (4.12). With $\widehat{\boldsymbol{\beta}}_0^{(k)}$ we start the AS-step again.

The AS-step envelops the FS-step. Both loops are repeated until $\delta_1 = \|\hat{\boldsymbol{\beta}}_0^{(k)} - \hat{\boldsymbol{\beta}}_0^{(k+1)}\|/\|\hat{\boldsymbol{\beta}}_0^{(k)}\| \leq \varepsilon$, for small ε.

Algorithm: glmOSCAR

Step 1 (Initialization) Choose $\widehat{\boldsymbol{\beta}}_0^{(0)}$ and set $\delta_1 = \infty$.

Step 2 (Iteration)

AS: While $\delta_1 > \varepsilon$.

- Determine $\boldsymbol{a}^{(k)}$ as described in (4.10).
- Determine $(\boldsymbol{0},\, \boldsymbol{A})^{(k)}$ as described in (4.11) and remove the inactive constraints.

FS: Set $\delta_2 = \infty$.

- Solve $\widehat{\boldsymbol{\beta}}_0^{(k+1)} = \operatorname{argmin}\big\{-\sum_{i=1}^n l_i(\boldsymbol{\beta}_0),\ \text{s.t. } (\boldsymbol{0},\, \boldsymbol{A})^{(k)}\boldsymbol{\beta}_0 \leq \boldsymbol{t}\big\}$ using a constrained Fisher scoring from (4.13) up to convergence $\delta_2 < \varepsilon$.
- After converging the constrained Fisher scoring (4.13) compute $\delta_1 = \frac{\|\hat{\boldsymbol{\beta}}_0^{(k)} - \hat{\boldsymbol{\beta}}_0^{(k+1)}\|}{\|\hat{\boldsymbol{\beta}}_0^{(k)}\|}$ and go to AS.

This algorithm can be generalized to a wide class of linearly restricted GLMs if the restricting halfspaces are defined by sign and rank.

4.5 Simulation Study

The settings of the simulation study are similar to the settings of Bondell and Reich (2008). However, we adapt the true parameter vectors to GLMs with canonical link function by scaling and changed the number of observations for some settings. We compare the OSCAR penalty with the MLE and two established methods:

LASSO: The LASSO penalty, which uses the penalty $\lambda \sum_{j=1}^{p} |\beta_j|$,

Elastic Net (EN): The EN, which uses a combination of the LASSO penalty term and the ridge term $\lambda \left[\alpha \sum_{i=1}^{p} |\beta_j| + (1-\alpha) \sum_{i=1}^{p} \beta_j^2\right]$.

Several program packages in `R` that fit the EN and the LASSO for GLMs are available (for example Lokhorst et al., 2007; Park and Hastie, 2007a; Friedman et al., 2008; Goeman, 2010b). We use the `R`-package `glmnet` (see Friedman et al., 2008, 2010a; Simon et al., 2011).

The predictive performance is measured by the predictive deviance

$$\text{Dev}(\boldsymbol{y}, \widehat{\boldsymbol{\mu}}, \phi) = -2\phi \sum_i (l(y_i, \mu_i) - l(y_i, y_i)),$$

where $\widehat{\boldsymbol{\mu}}$ is the estimated prediction based on data $(\boldsymbol{y}, \boldsymbol{X})$. First we fit the models for different tuning parameters on a training data set with n_{train} observations to get a set of parameter vector $\mathcal{B} = \{\widehat{\boldsymbol{\beta}}_0^{[1]}, ..., \widehat{\boldsymbol{\beta}}_0^{[q]}\}$ where the superscript $^{[q]}$ indicates the tuning parameter constellation. Then a validation data set with n_{vali} observations is used to determine the optimal tuning parameter constellation that minimizes the predictive deviance on the validation data set

$$\widehat{\boldsymbol{\beta}}_0^{[opt]} = \text{argmin}_{\widehat{\boldsymbol{\beta}}_0 \in \mathcal{B}} \left\{ \text{Dev}(\boldsymbol{y}_{vali}, h(\boldsymbol{Z}_{vali} \widehat{\boldsymbol{\beta}}_0), \phi) \right\}.$$

The test data is used to measure the predictive deviance

$$\text{Dev}_{\text{test}} = \text{Dev}(\boldsymbol{y}_{test}, h(\boldsymbol{Z}_{test} \widehat{\boldsymbol{\beta}}_0^{[opt]}), \phi).$$

In addition we give the mean square error of $\boldsymbol{\beta}$ $\text{MSE}_{\boldsymbol{\beta}} = p^{-1} \|\boldsymbol{\beta}_{true} - \widehat{\boldsymbol{\beta}}^{[opt]}\|^2$. We will consider the following settings.

Normal Case

For completeness we repeat the simulation study from Bondell and Reich (2008) with small modifications as described above. The generating model for all data sets is $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}_{true} + \boldsymbol{\varepsilon}$ where $\boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \sigma \boldsymbol{I})$.

Norm1 The true parameter vector is $\boldsymbol{\beta}_1 = (3, 2, 1.5, 0, 0, 0, 0, 0)^T$ and the covariates are from $N(\boldsymbol{0}, \boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma} = \{\sigma_{ij}\}_{i,j}$ with $\sigma_{ij} = 0.7^{|i-j|}$, $i, j = 1, ..., 8$. The number of observations are $n_{train} = 20$, $n_{vali} = 20$, and $n_{test} = 100$. As Bondell and Reich (2008) we choose $\sigma = 3$ for the standard deviation of the error term.

Norm2 This setting is the same as Norm1 but the true parameter vector is $\boldsymbol{\beta}_2 = (3, 0, 0, 1.5, 0, 0, 0, 2)^T$.

Norm3 This setting is the same as Norm1 and Norm2 but the true parameter vector is $\boldsymbol{\beta}_3 = 0.85 \cdot \boldsymbol{1}_8$.

Norm4 The true parameter vector is

$$\boldsymbol{\beta}_4 = (\underbrace{0, ..., 0}_{10}, \underbrace{2, ..., 2}_{10}, \underbrace{0, ..., 0}_{10}, \underbrace{2, ..., 2}_{10})^T.$$

In each block of ten the covariates are from a $N(\mathbf{0}, \boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma} = \{\sigma_{ij}\}_{i,j}$ with $\sigma_{ij} = 0.5$ if $i \neq j$ and $\sigma_{ii} = 1$, $i, j = 1, ..., 10$. Between the four blocks there is no correlation. The number of observations are $n_{train} = 100$, $n_{vali} = 100$, and $n_{test} = 500$. The standard deviation of the error term is $\sigma = 15$ (compare Bondell and Reich, 2008).

Norm5 The true parameter vector is

$$\boldsymbol{\beta}_5 = (\underbrace{3, ..., 3}_{15}, \underbrace{0, ..., 0}_{25})^T.$$

and the number of observations are $n_{train} = 50$, $n_{vali} = 50$, and $n_{test} = 250$. The covariates are generated as follows. V_1, V_2, and V_3 are iid from a univariate $N(0, 1)$ with $X_i = V_1 + \varepsilon_i$, $i = 1, ..., 5$, $X_i = V_2 + \varepsilon_i$, $i = 6, ..., 10$, $X_i = V_3 + \varepsilon_i$, $i = 11, ..., 15$, $X_i \sim N(0, 1)$, $i = 16, ..., 40$. where $\varepsilon_i \sim N(0, 0.16)$. So only the influential covariates are parted in three blocks of five. Inner each block the covariates are correlated. Between these blocks there is no correlation. The non influential covariates are uncorrelated and the standard deviation of the error term is $\sigma = 15$ (compare Bondell and Reich, 2008).

The results of this part of the simulation study is shown in Figure 4.2.

Poisson case

In the first three settings we divide the true parameter vector of the first three setting from Bondell and Reich (2008) by 4. The generating model of the Poisson setting has the form $y_i \sim Pois(\boldsymbol{x}_i^T \boldsymbol{\beta}_{true})$. The covariates are generated in the same way as in the normal case NormX.

Pois1 The true parameter vector is $\boldsymbol{\beta}_1 = (0.75, 0.5, 0.375, 0, 0, 0, 0, 0)^T$. The number of observations are $n_{train} = 20$, $n_{vali} = 20$, and $n_{test} = 100$.

Pois2 This setting is the same as Pois1 apart from the true parameter vector which is $\boldsymbol{\beta}_2 = (0.75, 0, 0, 0.375, 0, 0, 0, 0.5)^T$.

Pois3 This setting is the same as Pois1 and Pois2 apart from the true parameter vector $\boldsymbol{\beta}_3 = 0.2125 \cdot \mathbf{1}_8$.

Pois4 For this setting we divide the true parameter vector from Bondell and Reich (2008) by 20

$$\boldsymbol{\beta}_4 = (\underbrace{0, ..., 0}_{10}, \underbrace{0.1, ..., 0.1}_{10}, \underbrace{0, ..., 0}_{10}, \underbrace{0.1, ..., 0.1}_{10})^T.$$

The number of observations are $n_{train} = 100$, $n_{vali} = 100$, and $n_{test} = 500$.

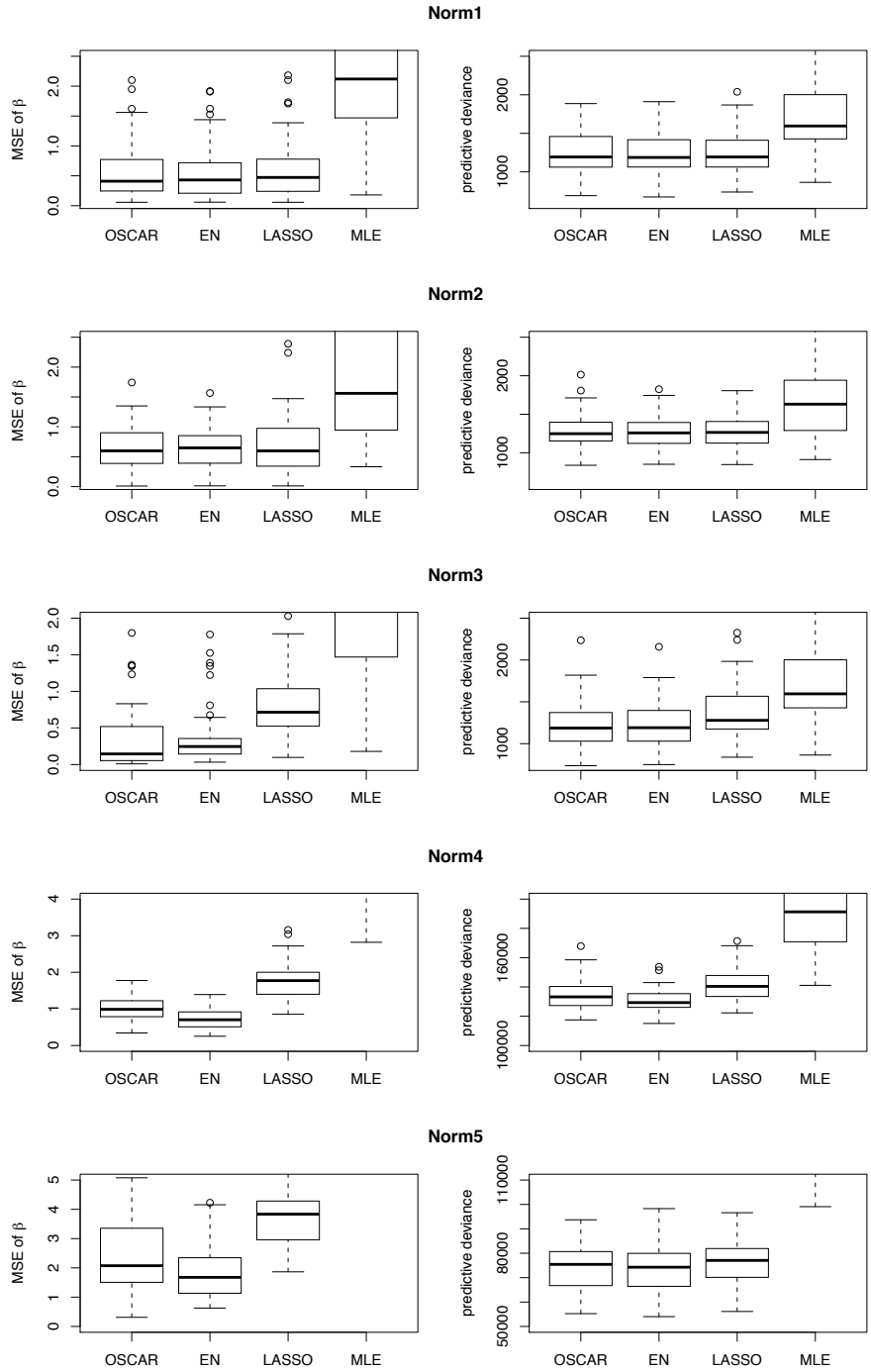

FIGURE 4.2: *Boxplots of* $\text{MSE}_{\boldsymbol{\beta}}$ *and* Dev_{test} *of the 5 settings in the normal case (see Bondell and Reich, 2008).*

Pois5 The true parameter vector Bondell and Reich (2008) is divided by 30

$$\boldsymbol{\beta}_5 = (\underbrace{0.1, ..., 0.1}_{15}, \underbrace{0, ..., 0}_{25})^T.$$

The number of observations are $n_{train} = 100$, $n_{vali} = 100$, and $n_{test} = 500$.

The result of this part of the simulation study is shown in Figure 4.3.

Binomial case

In all setting covariates are generated in the same way as in the corresponding Poisson setting PoisX. The generating model is $y_i \sim Bin(\boldsymbol{x}_i^T \boldsymbol{\beta}_{true})$. For the first three settings we divide the true parameter vector of the first three setting from Bondell and Reich (2008) by 2.

Bin1 The true parameter vector is $\boldsymbol{\beta}_1 = (1.5, 1, 0.75, 0, 0, 0, 0, 0)^T$. The number of observations are $n_{train} = 100$, $n_{vali} = 100$, and $n_{test} = 500$.

Bin2 This setting is the same as Bin1 but the true parameter vector is $\boldsymbol{\beta}_2 = (1.5, 0, 0, 0.75, 0, 0, 0, 1)^T$.

Bin3 This setting is the same as Bin1 and Bin2 but the true parameter vector $\boldsymbol{\beta}_3 = 0.425 \cdot \mathbf{1}_8$.

Bin4 We divide the true parameter vector from Bondell and Reich (2008) by 10

$$\boldsymbol{\beta}_4 = (\underbrace{0, ..., 0}_{10}, \underbrace{0.2, ..., 0.2}_{10}, \underbrace{0, ..., 0}_{10}, \underbrace{0.2, ..., 0.2}_{10})^T.$$

and increase the number of observations to $n_{train} = 200$, $n_{vali} = 200$, and $n_{test} = 1000$

Bin5 The true parameter vector Bondell and Reich (2008) is divided by 15

$$\boldsymbol{\beta}_5 = (\underbrace{0.2, ..., 0.2}_{15}, \underbrace{0, ..., 0}_{25})^T.$$

The number of observations is equal to Bin4.

In Figure 4.4 the results are illustrated by boxplots.

The results are summarized in Table 4.1. As a general tendency, it is seen that the procedures with clustering or grouping property outperform the LASSO, with the exception of settings Norm2 and Bin2. In the third settings the exact clustering of OSCAR seems to have an advantage over the non-exact grouping of the Elastic Net. Here the OSCAR dominates the other estimates. In the fourth setting OSCAR and EN outperform the LASSO, but the EN is the best for both criteria for all distributions. In the fifth setting the differences of the predictive deviance are quite small. With the exception of setting Bin2 the OSCAR is the best or second best for both criteria. In summary, the OSCAR for GLMs is a strong competitor to the Elastic Net, which outperforms the LASSO.

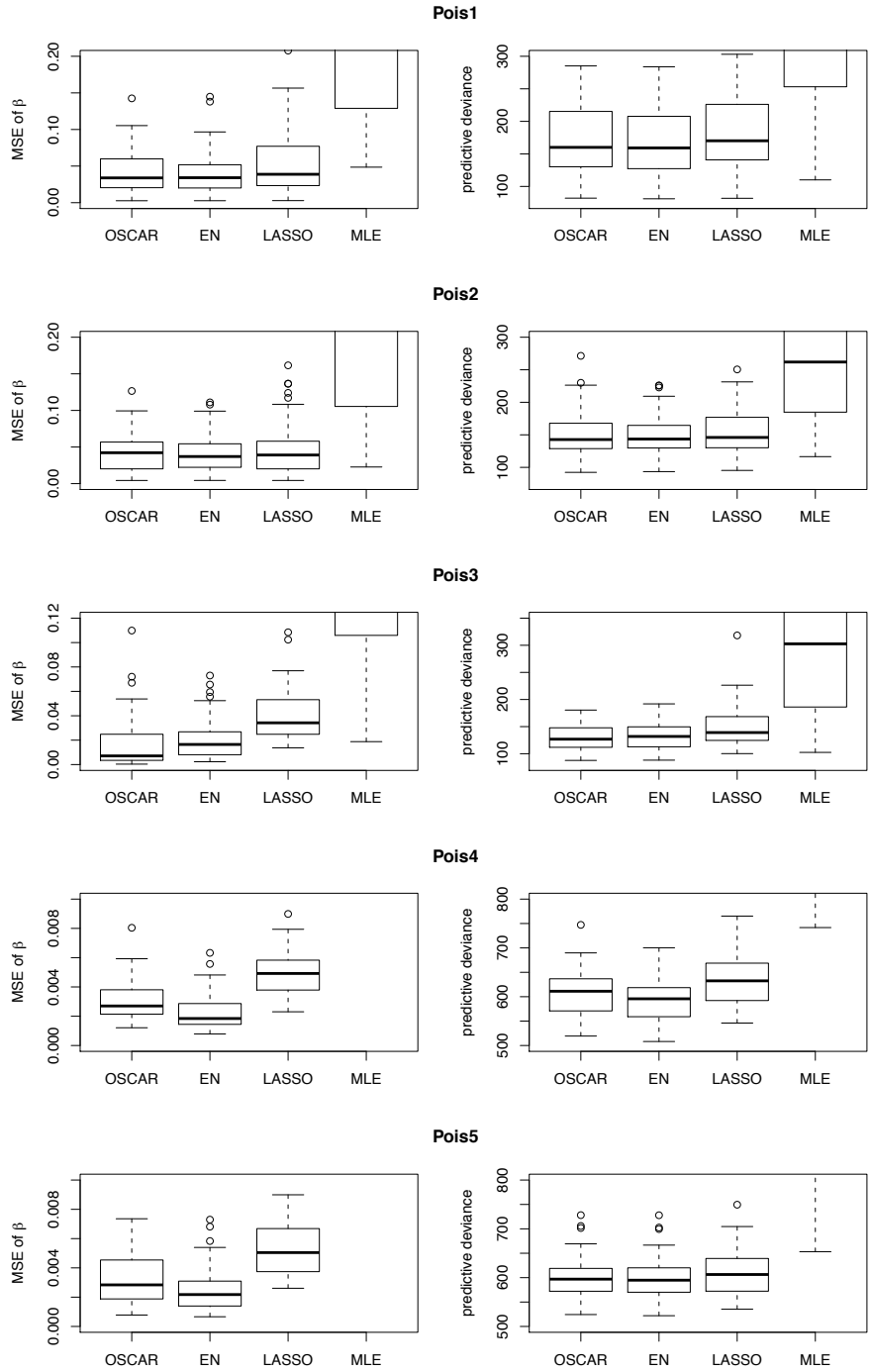

FIGURE 4.3: *Boxplots of* $\text{MSE}_{\boldsymbol{\beta}}$ *and* Dev_{test} *of the 5 Poisson settings.*

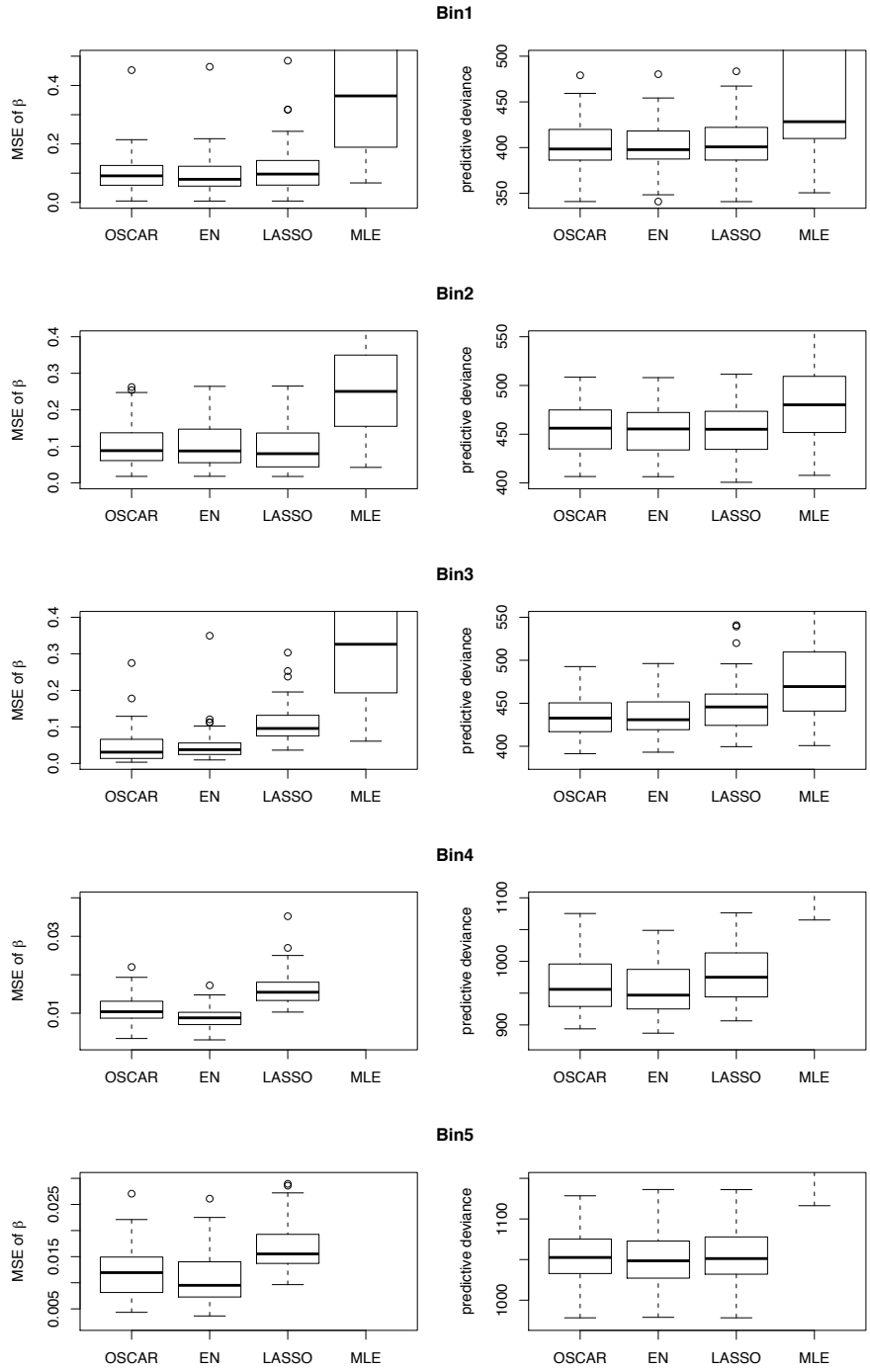

FIGURE 4.4: *Boxplots of* $\mathrm{MSE}_{\boldsymbol{\beta}}$ *and* Dev_{test} *of the 5 binomial settings.*

		OSCAR	Elastic Net	LASSO	MLE
Normal Case (Results of predictive deviances are divided by 100)					
β_1	$\text{MSE}_{\boldsymbol{\beta}}$	0.4095 (0.0754)	0.4303 (0.0667)	0.4724 (0.0781)	2.1195 (0.1283)
	Dev_{test}	11.93 (0.270)	11.87 (0.321)	11.93 (0.328)	15.95 (0.531)
β_2	$\text{MSE}_{\boldsymbol{\beta}}$	0.5985 (0.0587)	0.6484 (0.0555)	0.5981 (0.0750)	1.5609 (0.3545)
	Dev_{test}	12.48 (0.265)	12.58 (0.226)	12.64 (0.400)	15.44 (0.950)
β_3	$\text{MSE}_{\boldsymbol{\beta}}$	0.1212 (0.0610)	0.2272 (0.0333)	0.6321 (0.0733)	1.9654 (0.1888)
	Dev_{test}	11.27 (0.371)	11.72 (0.495)	12.54 (0.268)	15.45 (0.936)
β_4	$\text{MSE}_{\boldsymbol{\beta}}$	0.9893 (0.0449)	0.7034 (0.0609)	1.7730 (0.0753)	6.7606 (0.4305)
	Dev_{test}	1332.13 (15.919)	1293.09 (12.569)	1403.87 (23.670)	1912.99 (63.918)
β_5	$\text{MSE}_{\boldsymbol{\beta}}$	2.0738 (0.2089)	1.6770 (0.1335)	3.8346 (0.2317)	64.4542 (3.9849)
	Dev_{test}	754.37 (26.295)	742.73 (21.561)	770.61 (18.180)	3053.32 (192.65)
Poisson Case					
β_1	$\text{MSE}_{\boldsymbol{\beta}}$	0.0339 (0.0053)	0.0341 (0.0045)	0.0388 (0.0071)	0.2710 (0.0459)
	Dev_{test}	160.01 (12.845)	159.17 (8.859)	170.15 (13.045)	354.85 (58.980)
β_2	$\text{MSE}_{\boldsymbol{\beta}}$	0.0422 (0.0055)	0.0370 (0.0050)	0.0391 (0.0049)	0.2116 (0.0534)
	Dev_{test}	142.75 (4.031)	143.50 (4.108)	145.98 (4.727)	261.91 (42.965)
β_3	$\text{MSE}_{\boldsymbol{\beta}}$	0.0071 (0.0027)	0.0165 (0.0031)	0.0342 (0.0046)	0.3171 (0.0590)
	Dev_{test}	126.84 (5.399)	131.85 (5.928)	139.03 (6.500)	302.63 (51.125)
β_4	$\text{MSE}_{\boldsymbol{\beta}}$	0.0027 (0.0003)	0.0018 (0.0002)	0.0049 (0.0002)	0.0333 (0.0032)
	Dev_{test}	611.19 (8.774)	595.71 (9.000)	632.49 (7.975)	1295.12 (43.155)
β_5	$\text{MSE}_{\boldsymbol{\beta}}$	0.0028 (0.0004)	0.0022 (0.0002)	0.0050 (0.0002)	0.0515 (0.0035)
	Dev_{test}	596.93 (7.479)	594.78 (6.857)	606.44 (6.359)	1192.28 (119.52)
Binomial Case					
β_1	$\text{MSE}_{\boldsymbol{\beta}}$	0.0908 (0.0140)	0.0790 (0.0115)	0.0968 (0.0188)	0.3642 (0.0809)
	Dev_{test}	398.41 (3.693)	397.71 (4.283)	400.76 (4.194)	428.32 (10.937)
β_2	$\text{MSE}_{\boldsymbol{\beta}}$	0.0883 (0.0134)	0.0875 (0.0152)	0.0800 (0.0096)	0.2504 (0.0194)
	Dev_{test}	456.17 (5.394)	455.39 (6.554)	455.07 (6.657)	480.13 (7.296)
β_3	$\text{MSE}_{\boldsymbol{\beta}}$	0.0309 (0.0053)	0.0376 (0.0040)	0.0958 (0.0089)	0.3262 (0.0475)
	Dev_{test}	432.73 (4.484)	430.86 (5.779)	445.72 (5.523)	469.44 (7.657)
β_4	$\text{MSE}_{\boldsymbol{\beta}}$	0.0104 (0.0005)	0.0095 (0.0005)	0.0155 (0.0009)	0.1921 (0.0136)
	Dev_{test}	956.02 (7.059)	946.90 (7.069)	975.07 (6.156)	1354.75 (33.367)
β_5	$\text{MSE}_{\boldsymbol{\beta}}$	0.0119 (0.0009)	0.0093 (0.0008)	0.0155 (0.0010)	0.1806 (0.0221)
	Dev_{test}	1052.56 (6.903)	1048.43 (6.005)	1051.21 (7.999)	1366.61 (25.889)

TABLE 4.1: *Summary of the results of the Simulation study*

4.6 Application

The data were collected in water engineering in Southern California and contain 43 years worth of precipitation measurements. They are available from the R-package `alr3` (see Weisberg, 2011, 2005). The response variable is the stream runoff near Bishop (CA) in acre-feet. There are six covariates which are the snowfall in inches at different measurement stations labeled by `APMAM`, `APSAB`, `APSLAKE`, `OPBPC`, `OPRC`, and `OPSLAKE`. The covariates are grouped by its position. The covariates with labels that start with the same letter are quite close to each other and are highly correlated. The correlation structure is shown in Figure 4.5. We consider two cases: First we fit a linear normal model to predict the stream

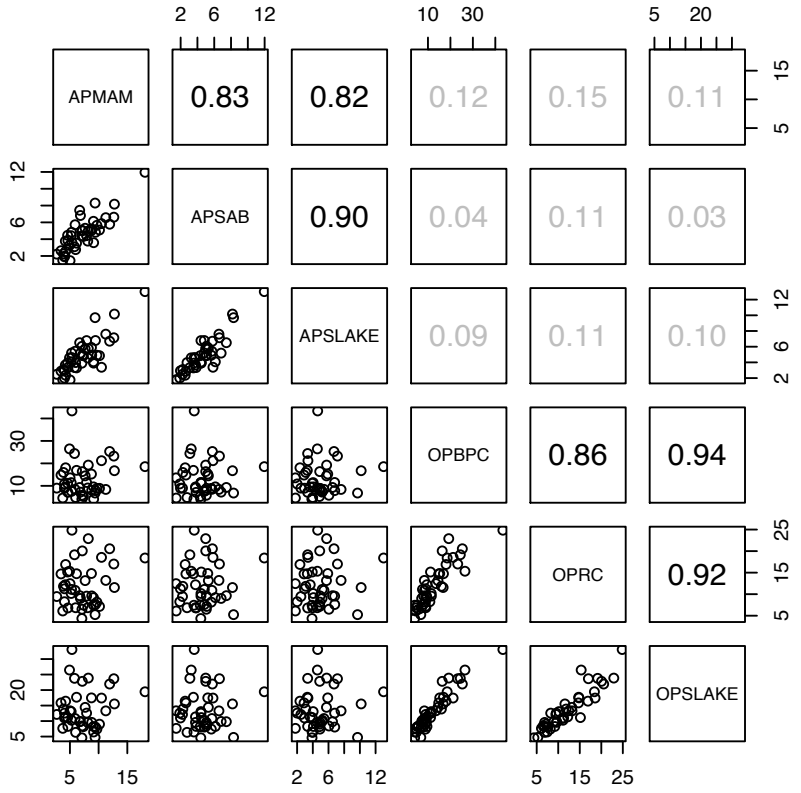

FIGURE 4.5: *Correlation structure of the covariates of the water data.*

runoff. Then we split the response variable in two parts by setting the response $y_i = 0$ if $y_i < \text{median}(\boldsymbol{y})$ and $y_i = 1$ if $y_i \geq \text{median}(\boldsymbol{y})$. With this binary response we fit a GLM with binomial distribution and logit link. The tuning parameter are determined by

$$AIC = 2\sum_{i=1}^{n} l_i(\boldsymbol{\beta}_0) + 2(df + 1).$$

Bondell and Reich (2008) proposed for *df* the number of coefficients that are absolute unique but non zero, or, in other words, the number of distinct non zero entries of $|\boldsymbol{\beta}|$. We use the *AIC* to determine the tuning parameters because the MLE exists, which is necessary for using the `quadprog` procedure (see Turlach, 2009). Cross-validation does not work, especially in the binomial case, because the MLE does not exist for all sub-samples. For the binomial case $c = 0.9$ and for the normal case $c = 0.2$ was determined. The EN and the LASSO paths were calculated with the `glmnet` (see Friedman et al., 2008, 2010a; Simon et al., 2011). For the EN we determine $\alpha = 0.9$ in the normal case and $\alpha = 0.5$ in the binomial case. The coefficient buildups of standardized coefficients for the different procedures are shown in Figure 4.6, OSCAR is in the first row, LASSO in the second row, and the Elastic Net is in the third row of Figure 4.6. On the left the solution paths of the normal distribution case and on the right of the binomial distribution case are given. The dotted horizontal line show the optimal tuning parameter t. The coefficient buildups of the OSCAR show a strong influence of the measurement stations that have names starting with "O". Especially in the normal case the clustering and the variable selection of OSCAR is quite impressive. All variables of the group starting with "O" are estimated equal and the second group is shrunken to zero for AIC optimal t. In the binary case clustering and variable selection is somewhat weaker, but still impressive, in particular when compared to to the elastic net. For optimal t `OPBPC` and `OPSLAKE` are clustered as well as two weaker correlated covariates (`APMAM` and `OPRC`). Only the variable `APSAB` is shrunken to zero. In the normal case the clustering coefficient buildups of EN and LASSO are quite similar. In the binomial case the EN has at least a tendency to cluster the covariates starting with "O". The exact clustering of covariates is easy to interpret, especially in the normal case. The snowfall at adjacent measurement stations has the same influence on the stream runoff. But only the influence of the snowfall at the measurement stations that have names starting with "O" have non-zero influence. The remaining (starting with an "A") are shrunk to zero.

4.7 Conclusion and Remarks

We adapt the OSCAR penalty to GLMs. For solving the constrained log-likelihood problem we present an algorithm which combines the active set method and Fisher scoring. It turns out that the OSCAR is quite competitive. In the simulation study it is the best or second best (with the exception of one setting) in terms of the $\text{MSE}_{\boldsymbol{\beta}}$ and the predictive deviance. Especially in the normal case the result of the data example is good to interpret. The snowfall at closed measurement stations is quite similar and so it can be assumed that their influence on the stream runoff is nearly equal. The data example also illustrates that the LASSO picks only two highly correlated covariates out of the group of three.

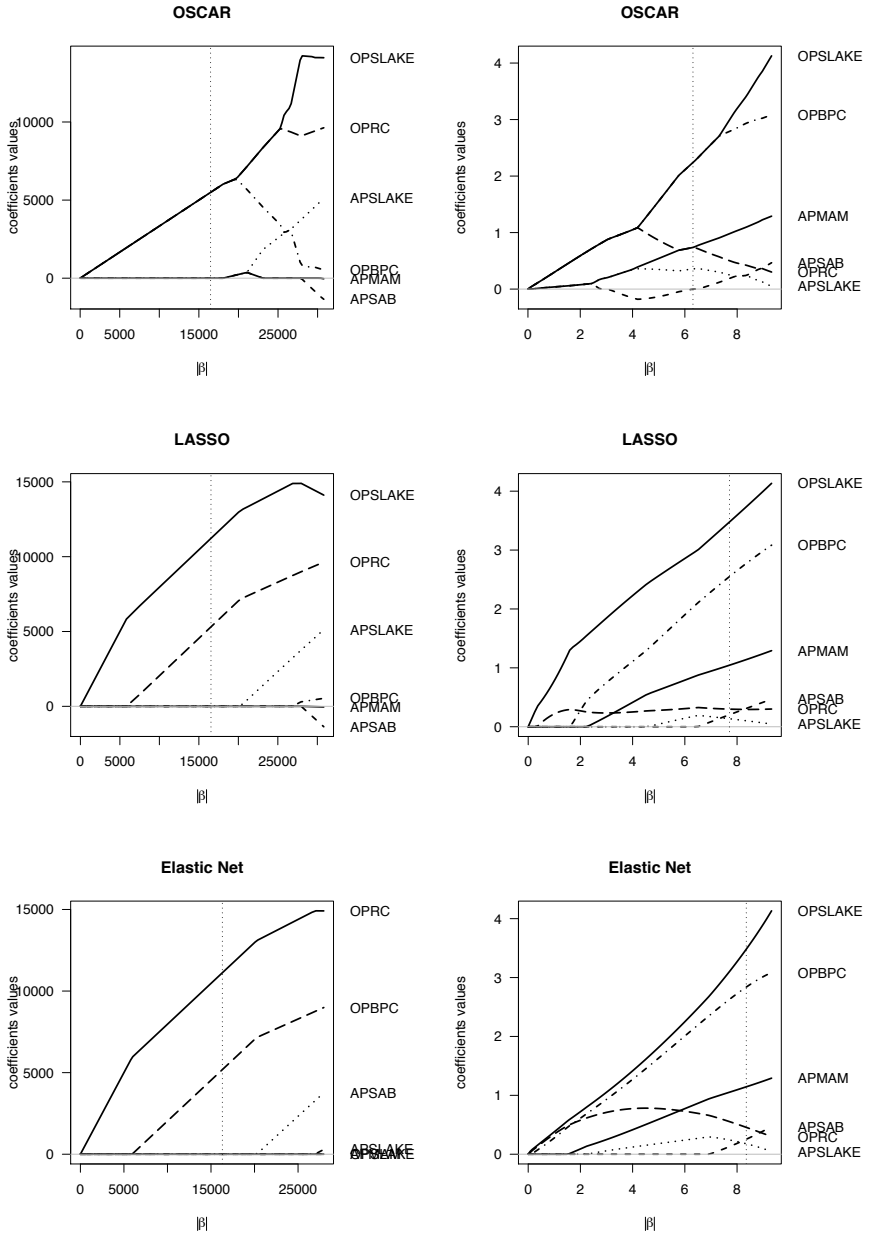

FIGURE 4.6: *Coefficient buildups for the water data. The left column shows the normal case and the right column shows the binary distribution case. In the first row the solution paths of the OSCAR are given, the second row shows the LASSO- and the third row the EN-paths.*

Part II

Regularization Approaches for Single Index Models

Chapter 5

Nonparametric Estimation of the Link Function Including Variable Selection

Nonparametric methods for the estimation of the link function in generalized linear models are able to avoid bias in the regression parameters. But for the estimation of the link typically the full model, which includes all predictors, has been used. When the number of predictors is large these methods sometimes fail since the full model can not be estimated. This chapter is based on Tutz and Petry (2011). It presents a boosting type method is proposed that simultaneously selects predictors and estimates the link function. The method performs quite well in simulations and real data examples.

5.1 Introduction

In generalized linear models (GLMs), for given data $(y_i, \boldsymbol{x}_i)$, $i = 1, \ldots, n$, the conditional expectation of $y_i|\boldsymbol{x}_i$, $\mu_i = E(y_i|\boldsymbol{x}_i)$, is modeled by

$$g(\mu_i) = \eta_i \quad \text{or} \quad \mu_i = h(\eta_i),$$

where $\eta_i = \beta_0 + \boldsymbol{x}_i^T\boldsymbol{\beta}$ is the linear predictor, $g(.)$ is the link function and $h(.) = g^{-1}(.)$ is the response function.

Usually it is assumed that the response function $h(.)$ is fixed and known, for example $h(.) = \exp(.)$ yields the loglinear model which represents the canonical link model if responses follow a Poisson distribution. In applications, typically the link function is unknown and frequently the canonical link function is used. But it is well known that misspecification of the link function can lead to substantial bias in the regression parameters (see Czado and Santner (1992) for binomial responses). That may be avoided by flexible modeling of the link.

When responses are metrically scaled, a flexible generalization of classical approaches is the so-called single-index model. It assumes that $h(.)$ is unknown and has to be estimated

by nonparametric techniques. The model may be seen as a special case of projection pursuit regression, which assumes that μ_i has additive form $h_1(\boldsymbol{x}_i^T\boldsymbol{\beta}_1) + \cdots + h_m(\boldsymbol{x}_i^T\boldsymbol{\beta}_m)$ with unknown functions $h_1, \ldots, h_m$, which transform the indices $\boldsymbol{x}_i^T\boldsymbol{\beta}_j$ (see Friedman and Stützle, 1981). In single index models only one index, $\boldsymbol{x}_i^T\boldsymbol{\beta}$, is assumed. The difference between a single index model and a GLM is that in the former the transformation function $h(.)$ is not restricted and unknown whereas in GLMs it is assumed that $h(.)$ is known and strictly monotone, hence invertible. Although single index models are useful in dimension reduction, strict monotonicity, as assumed in GLMs, is very helpful when parameters are to be interpreted. Therefore we will focus on monotonic response functions. Then nonparametric estimation of the function $h(.)$ may be seen as estimation of the unknown link function in a GLM.

Estimation of the unknown link function when the underlying distribution is from a simple exponential family was considered for example by Weisberg and Welsh (1994), Ruckstuhl and Welsh (1999) and Muggeo and Ferrara (2008). Weisberg and Welsh (1994) proposed to estimate regression coefficients using the canonical link and then estimate the link via kernel smoothers given the estimated parameters. Then parameters are reestimated. Alternating between estimation of link and parameters yields consistent estimates. But all these approaches do not select predictors.

The main advantage of the presented approach is that it combines estimation of the link function with variable selection. In the last decade the traditional forward/backward procedures for the selection of variables have been widely replaced by regularized estimation methods that implicitly select predictors, among them the LASSO (Tibshirani, 1996), which was adapted to GLMs by Park and Hastie (2007b), the Dantzig selector (James and Radchenko, 2008), SCAD (Fan and Li, 2001) and boosting approaches (Bühlmann and Hothorn, 2007; Tutz and Binder, 2006). However, in all of these procedures selection is always based on a known response function. If the assumed response function is wrong the performance of these selection procedures can be strongly affected. For illustration let us consider a small simulation study.

We fitted a Poisson model with the true response function having sigmoidal form $h_T(\eta) = 10/(1 + \exp(-5 \cdot \eta))$. The parameter vector of length $p = 20$ was $\boldsymbol{\beta}^T = (0.2, 0.4, -0.4, 0.8, 0, \ldots, 0)$ and covariates were drawn from a normal distribution $\boldsymbol{X} \sim N(\mathbf{0}_p, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma} = \{\sigma_{ij}\}_{i,j\in\{1,\ldots,p\}}$ where $\sigma_{ij} = 0.5$, $i \neq j$, $\sigma_{ii} = 1$.

We generated $N = 50$ data sets with $n = 200$ observations and fitted the model by using the usual maximum likelihood (ML) procedure based on the canonical log-link (without variable selection). In addition, we applied three alternative fitting methods that include variable selection: the nonparametric flexible link procedure derived in Section 5.2.2, the LASSO for generalized linear models (Lokhorst et al., 2007) and a boosting procedure (Hothorn et al., 2009, 2010). The latter procedure is based on componentwise boosting, which is also the selection procedure used in the flexible link procedure. While the flexible link procedure selects a link function, ML estimates as well as LASSO and boosting use the canonical link. It is seen in the upper four panels from Figure 5.1 that LASSO and boosting, which include variable selection perform distinctly better than classical maximum likelihood fitting. But the best results are obtained if the link function is estimated

nonparametrically. In particular the parameters of predictors that are not influential are estimated more stably and closer to zero. The dominance of the flexible procedure is also seen in the two lower panels from Figure 5.1, which shows the mean squared error for the estimation of the parameter vector and the predictive deviance on an independently drawn test data set with $n = 1000$. For more details see Section 5.3.

For normally distributed responses various estimation methods for single-index models have been proposed. One popular technique is based on average derivative estimation (see Stoker, 1986; Powell et al., 1989; Hristache et al., 2001). Alternatively M-estimation has been applied, which considers the unknown link function as an infinite dimensional nuisance parameter (see e.g. Klein and Spady, 1993). Other authors focus (more) on the estimation of $h(.)$. Based on kernel regression techniques, Härdle et al. (1993) investigated the optimal amount of smoothing in single-index models when simultaneous estimating $\boldsymbol{\beta}$ and the bandwidth. Yu and Ruppert (2002) suggested to use penalized regression splines. They also allow for partially linear terms in the model and report more stable estimates compared to earlier approaches based on local regression (e.g. Carroll et al., 1997). Tutz and Leitenstorfer (2011) proposed a boosted version of the penalized regression splines approach, but without variable selection. More recently, Gaiffas and Lecue (2007) proposed an aggregation algorithm with local polynomial fits and investigated optimal convergence rates. Bayesian approaches were proposed by Antoniadis et al. (2004). More general distribution models have been considered by Weisberg and Welsh (1994) who proposed an algorithm that alternates between the estimation of $\boldsymbol{\beta}$ and $h(.)$.

In the following we will extend the penalized regression splines approach used by Yu and Ruppert (2002) and Tutz and Leitenstorfer (2011) to the more general case of responses that follow a simple exponential family and include variable selection. Both articles, Yu and Ruppert (2002) as well as Tutz and Leitenstorfer (2011), consider Gaussian responses only and do not provide tools for variable selection. In Section 5.2 the estimation procedure is given, in Section 5.3 the method is compared to competitors. In Section 5.4 a modified version that allows to reduce the false positives is introduced. Applications are given in Section 5.5.

5.2 Estimation

5.2.1 Data Generating and Approximating Model

We assume that the *data generating model* is

$$E(y_i|\boldsymbol{x}_i) = \mu_i = h_T(\eta_i),$$

where $h_T(.)$ is the unknown true transformation function and $\eta_i = \boldsymbol{x}_i^T\boldsymbol{\beta}$ is the linear predictor. Given $\boldsymbol{x}_i$ the y_i are (conditionally) independent observations from a simple exponential family

$$f(y_i|\theta_i, \phi) = \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{\phi} + c(y_i, \phi)\right\}, \tag{5.1}$$

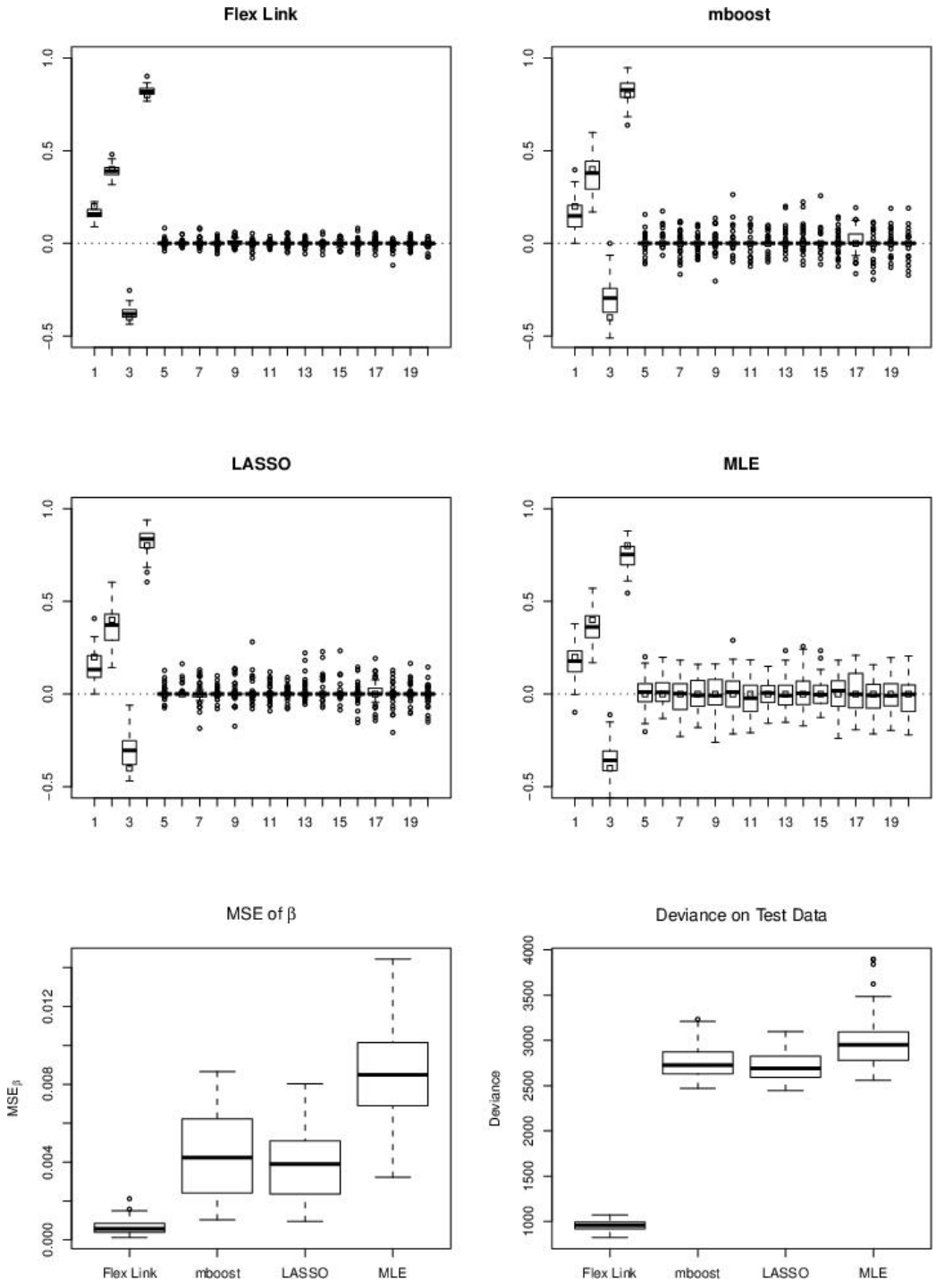

FIGURE 5.1: *Estimates of coefficient vector in simulation study for flexible link, boosting, LASSO and ML and the mean squared error for parameter vector and predictive deviance for simulation setting.*

where θ_i is the natural parameter of the family, ϕ is a scale or dispersion parameter and $b(.), c(.)$ are specific functions corresponding to the type of the family. For uniqueness we will assume that for the true parameter $||\boldsymbol{\beta}|| = 1$ holds and that the linear predictor η_i contains no intercept. Thus, the magnitude of $||\boldsymbol{\beta}||$ and the intercept are absorbed into $h_T(.)$. With $||.||$ we denote the Euclidean norm.

The *approximating model* that is fitted has the form

$$\mu_i = h_0(h(\eta_i)),$$

where $h_0(.)$ is a fixed transformation function, which has to be chosen. The function $h(.)$ is considered as unknown and has to be estimated. Typically, the choice of $h_0(.)$ depends on the distribution of the response. When the response is binary a canonical choice is the logistic distribution function. The main advantage of specifying a fixed link function is that it may be selected such that the predictor is automatically mapped into the admissible range of the mean response. For example, the logistic distribution function has values from [0, 1], which is appropriate for binary responses. Thus, in contrast to procedures that estimate $h_T(.)$ directly, we estimate the inner function $h(.)$, as for example Muggeo and Ferrara (2008) did.

The function $h(.)$ will be approximated by expansion in basis functions

$$h(\eta_i) = \sum_{s=1}^{k} \alpha_s \phi_s(\eta_i) = \Phi_i^T \boldsymbol{\alpha}, \tag{5.2}$$

where $\phi_1, \ldots, \phi_k$ denote the basis functions. As basis functions we use natural B-splines of degree 3 (see Dierckx, 1993), which are provided by the `fda` package in `R` (see Ramsey and Silverman, 2005; Ramsay et al., 2010). One problem with basis functions is that a sequence of knots $\{\tau_j\}_1^k$ has to be placed in a certain domain $[\eta_{\min}, \eta_{\max}]$ where the response function is to be estimated. Since the parameter vector is normalized by setting $||\boldsymbol{\beta}|| = 1$, one can infer from the Cauchy-Schwarz-inequality that the range of $\eta_i = \boldsymbol{x}_i^T\boldsymbol{\beta}$, $i \in \{1, ..., n\}$ is restricted to $[-u, u]$ where $u = \max_{i=1,\ldots,n}\{||\boldsymbol{x}_i||\}$. We will use equidistant knots on $[-u, u]$. As in P-spline regression (Eilers and Marx, 1996), a high number of knots is used and the smoothness of the function estimate is controlled by appropriate penalization. As penalty term for the estimation of $\boldsymbol{\alpha}$ we use the integral of the squared second derivation of the approximated response function $h(.)$ given by (5.2), $\int_{-u}^{u} \left(\frac{d^2}{d\eta^2} h(\eta)\right)^2 d\eta$, which can be given in matrix form as $\boldsymbol{P} = \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha}$ with symmetric matrix $\boldsymbol{K} = \{k_{ij}\}$, $k_{ij} = \int_{-u}^{u} \left(\frac{d^2}{d\eta^2}\phi_i(\eta)\right)\left(\frac{d^2}{d\eta^2}\phi_j(\eta)\right) d\eta$. We will use the B-spline basis of the `fda` package (Ramsey and Silverman, 2005; Ramsay et al., 2010) where the penalty matrix $\boldsymbol{K}$ is implemented.

5.2.2 Estimation of Parameters Including Variable Selection

Componentwise boosting techniques have been successfully used to select relevant predictors in classical linear and generalized linear models (see, for example, the overview given

by Bühlmann and Hothorn (2007). The basic principle is to update within one step only one single component, in our case one coefficient of the predictor. With the link function being unknown also the coefficients of basis functions have to be estimated. In contrast to the selection procedure for the components of $\boldsymbol{\beta}$ the estimation of the coefficients of basis functions includes no selection step. Since the underlying link function is assumed to be smooth, estimates are updated by using penalized estimation.

We will use likelihood-based boosting techniques, which aim at the maximization of the log-likelihood. As usual in boosting no explicit penalty on the log-likelihood is specified. Regularization is obtained implicitly by stopping the iteration procedure. The specific advantages of boosting techniques concerning the trade-off between bias and variance have been derived by Bühlmann and Yu (2003). Moreover, it has been shown that in special cases boosting is very similar to LASSO regularized estimates (see Efron et al., 2004). The penalization techniques that are used here follow the same principles as likelihood-based boosting outlined in Tutz and Binder (2006).

Computation of estimates uses boosting techniques in two stages, once for the estimation of the parameter vector $\boldsymbol{\beta}$ and once for the estimation of the vector of basis coefficients $\boldsymbol{\alpha}$. Before giving the algorithm we will consider the two stages (and initialization) separately. For simplicity we will use matrix notation with $\boldsymbol{X}$ denoting the design matrix of predictors, and $\widehat{\boldsymbol{\beta}}^{(l)}$, $\widehat{\boldsymbol{\eta}}^{(l)} = \boldsymbol{X}\widehat{\boldsymbol{\beta}}^{(l)}$ denoting the parameter estimate and the fitted predictor in the lth step. Moreover, $\boldsymbol{\Phi}^{(l)} = (\boldsymbol{\Phi}_1^{(l)}, \ldots, \boldsymbol{\Phi}_n^{(l)})^T$ with $\boldsymbol{\Phi}_i^{(l)} = (\phi_1(\widehat{\eta}_i^{(l)}), \ldots, \phi_k(\widehat{\eta}_i^{(l)}))^T$ is the current design matrix for the basis functions. In the following, for data $(y_i, \boldsymbol{x}_i)$, $i = 1, \ldots, n$, let $l(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_i (y_i\theta_i - b(\theta_i))/\phi$ denote the log-likelihood of the model. It depends on $\boldsymbol{\alpha}$ through $h(\eta_i) = \boldsymbol{\Phi}_i^T\boldsymbol{\alpha}$ and on $\boldsymbol{\beta}$ through $\mu_i = h_0(h(\boldsymbol{x}_i^T\boldsymbol{\beta}))$, where $\theta_i = \theta(\mu_i)$ for a known function $\theta(.)$.

Initialization

We need two initialization values, $\widehat{\boldsymbol{\beta}}^{(0)}$ and $\widehat{\boldsymbol{\alpha}}^{(0)}$. For $\widehat{\boldsymbol{\beta}}^{(0)}$ we choose $\widehat{\boldsymbol{\beta}}^{(0)} = \mathbf{0}_p$. The initialization value for the coefficient vector of the basis functions $\widehat{\boldsymbol{\alpha}}^{(0)}$ is generated by approximating $h(\eta)$ by a linear function, $s \cdot \eta + t$, where $t = h_0^{-1}(\bar{\boldsymbol{y}})$ and the slope is chosen as a small value ($s = 0.0001$).

Boosting for Fixed Predictor

For fixed predictor $\widehat{\boldsymbol{\eta}}^{(l-1)} = \boldsymbol{X}\widehat{\boldsymbol{\beta}}^{(l-1)}$ the estimation of the response function corresponds to fitting the model $\boldsymbol{\mu} = h_0(\boldsymbol{\Phi}^{(l-1)}\widehat{\boldsymbol{\alpha}}^{(l-1)} + \boldsymbol{\Phi}^{(l-1)}\widehat{\boldsymbol{a}}^{(l)})$ where $\boldsymbol{\Phi}^{(l-1)}\widehat{\boldsymbol{\alpha}}^{(l-1)}$ is a fixed offset that represents the previously fitted value. One step of penalized Fisher scoring has the form

$$\begin{aligned}\widehat{\boldsymbol{a}}^{(l)} &= \nu_h \cdot \Big((\boldsymbol{\Phi}^{(l-1)})^T(\widehat{\boldsymbol{D}}^{(l-1)})^T(\widehat{\boldsymbol{\Sigma}}^{(l-1)})^{-1}\widehat{\boldsymbol{D}}^{(l-1)}\boldsymbol{\Phi}^{(l-1)} + \lambda\boldsymbol{K}\Big)^{-1} \times \\ &\quad \times(\boldsymbol{\Phi}^{(l-1)})^T(\widehat{\boldsymbol{D}}^{(l-1)})^T(\widehat{\boldsymbol{\Sigma}}^{(l-1)})^{-1}(\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^{(l-1)})\end{aligned} \tag{5.3}$$

where

$$\widehat{\boldsymbol{D}}^{(l-1)} = \text{diag}\left\{\frac{\partial h_0(\widehat{h}^{(l-1)}(\widehat{\eta}_i^{(l-1)}))}{\partial \widehat{h}^{(l-1)}(\eta)}\right\}_{i=1}^{n} \tag{5.4}$$

is the estimate of the derivative matrix evaluated at the estimate of the previous step $h_0(\widehat{h}^{(l-1)}(\eta))$ and

$$\widehat{\boldsymbol{\Sigma}}^{(l-1)} = \text{diag}\left\{\sigma^2(h_0(\widehat{h}^{(l-1)}(\widehat{\eta}_i^{(l-1)})))\right\}_{i=1}^{n} \tag{5.5}$$

is the matrix of variances evaluated at $h_0(\widehat{h}^{(l-1)}(\eta))$. $\boldsymbol{K}$ is the penalty matrix which penalizes the second derivation of the estimated (approximated) response function. The shrinkage parameter, which makes the procedure a weak learner, is fixed by $\nu_h = 0.1$.

Componentwise Boosting for Fixed Response Function

Let $\widehat{h}^{(l-1)}(.)$ be fixed and the design matrix have the form $\boldsymbol{X} = (\boldsymbol{x}_1|...|\boldsymbol{x}_p)$ with corresponding response vector $\boldsymbol{y} = (y_1, ..., y_n)^T$. Componentwise boosting means to update one parameter within one boosting step. Therefore one fits the model $\boldsymbol{\mu} = h_0(\widehat{h}^{(l-1)}(\boldsymbol{X}\widehat{\boldsymbol{\beta}}^{(l-1)} + \boldsymbol{x}_j^T\widehat{b}_j^{(l)}))$, where $\boldsymbol{X}\widehat{\boldsymbol{\beta}}^{(l-1)}$ is a fixed offset and only the variable $\boldsymbol{x}_j$ is included in the model. Then penalized Fisher scoring for parameter $\widehat{b}_j^{(l)}$ has the form

$$\widehat{b}_j^{(l)} = \nu_p \left(\boldsymbol{x}_j^T(\widehat{\boldsymbol{D}}_\eta^{(l-1)})^T(\widehat{\boldsymbol{\Sigma}}^{(l-1)})^{-1}\widehat{\boldsymbol{D}}_\eta^{(l-1)}\boldsymbol{x}_j\right)^{-1}\boldsymbol{x}_j^T(\widehat{\boldsymbol{D}}_\eta^{(l-1)})^T(\widehat{\boldsymbol{\Sigma}}^{(l-1)})^{-1}(\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^{(l-1)}), \tag{5.6}$$

where $\nu_p = 0.1$ and

$$\begin{aligned}\widehat{\boldsymbol{D}}_\eta^{(l-1)} &= \text{diag}\left\{\frac{\partial h_0(\widehat{h}^{(l-1)}(\widehat{\eta}_i^{(l-1)}))}{\partial \eta}\right\}_{i=1}^{n} \\ &= \text{diag}\left\{\frac{\partial h_0(\widehat{h}^{(l-1)}(\widehat{\eta}_i^{(l-1)}))}{\partial \widehat{h}^{(l-1)}(\eta)} \cdot \frac{\partial \widehat{h}^{(l-1)}(\widehat{\eta}_i^{(l-1)})}{\partial \eta}\right\}_{i=1}^{n}\end{aligned} \tag{5.7}$$

is the matrix of derivatives evaluated at the values of the previous iteration and

$$\widehat{\boldsymbol{\Sigma}}^{(l-1)} = \text{diag}\left\{\sigma^2(h_0(\widehat{h}^{(l-1)}(\widehat{\eta}_i^{(l-1)})))\right\}_{i=1}^{n} \tag{5.8}$$

is the variance from the previous step.

The basic algorithm given below computes updates of the parameter vector and the coefficients of the basic functions. In each step it is decided which update is best and only one is executed. Thus in each step either the parameter vector or the coefficients of the basic functions are refitted.

Algorithm: FlexLink

Step 1 (Initialization)

Set $\widehat{\boldsymbol{\beta}}^{(0)} = \mathbf{0}$, $\widehat{\boldsymbol{\eta}}^{(0)} = \mathbf{0}$ and determine $\widehat{\boldsymbol{\alpha}}^{(0)}$ as described previously. Compute $\widehat{\boldsymbol{D}}^{(0)} = \text{diag}\left\{\partial h_0(\widehat{h}^{(0)}(\widehat{\eta}_i^{(0)}))/\partial \widehat{h}^{(0)}(\eta)\right\}_{i=1}^{n}$, $\widehat{\boldsymbol{D}}_{\eta}^{(0)} = \text{diag}\left\{\partial h_0(\widehat{h}^{(0)}(\widehat{\eta}_i^{(0)}))/\partial \eta\right\}_{i=1}^{n}$, $\widehat{\boldsymbol{\Sigma}}^{(0)} = \text{diag}\left\{\sigma^2(h_0(\widehat{h}^{(0)}(\widehat{\eta}_i^{(0)})))\right\}_{i=1}^{n}$.

Step 2 (Iteration)

For $l = 1, 2, \ldots, M$

1. *Predictor update*
   - Compute for every $j \in \{1, \ldots, p\}$ the penalized estimate $\widehat{\boldsymbol{b}}_j^{(l)} = (0, \ldots, \widehat{b}_j^{(l)}, \ldots, 0)$ based on one-step Fisher scoring (5.6) and determine the candidate update $\widehat{\boldsymbol{\beta}}_j^{(l)} = \widehat{\boldsymbol{\beta}}^{(l-1)} + \widehat{\boldsymbol{b}}_j^{(l)}$.
   - Standardize $\widehat{\boldsymbol{\beta}}_j^{(l)}$ by computing $\widehat{\boldsymbol{\beta}}_j^{(l)}/||\widehat{\boldsymbol{\beta}}_j^{(l)}||$ and the corresponding log-likelihood function $l(\widehat{\boldsymbol{\alpha}}^{(l-1)}, \widehat{\boldsymbol{\beta}}_j^{(l)})$.
   - Choose the parameter vector $\widehat{\boldsymbol{\beta}}_{opt}^{(l)} = \text{argmax}_j\, l(\widehat{\boldsymbol{\alpha}}^{(l-1)}, \widehat{\boldsymbol{\beta}}_j^{(l)})$, which maximizes the log-likelihood function and set $\widehat{\boldsymbol{\beta}}^{(l)} = \widehat{\boldsymbol{\beta}}_{opt}^{(l)}$
2. *Response function update*
   - Compute $\widehat{\boldsymbol{a}}^{(l)}$ as described in (5.3) and set $\widehat{\boldsymbol{\alpha}}^{(l)} = \widehat{\boldsymbol{\alpha}}^{(l-1)} + \widehat{\boldsymbol{a}}^{(l)}$
   - Compute $\widehat{h}^{(l)}(\widehat{\boldsymbol{\eta}}^{(l-1)}) = \boldsymbol{\Phi}^{(l-1)}\widehat{\boldsymbol{\alpha}}^{(l)}$ and $l(\widehat{\boldsymbol{\alpha}}^{(l)}, \widehat{\boldsymbol{\beta}}^{(l-1)})$.
3. *Update choice*
   - If $l(\widehat{\boldsymbol{\alpha}}^{(l)}, \widehat{\boldsymbol{\beta}}^{(l-1)}) > l(\widehat{\boldsymbol{\alpha}}^{(l-1)}, \widehat{\boldsymbol{\beta}}^{(l)})$ then $\widehat{\boldsymbol{\alpha}}^{(l)}$ is updated and $\widehat{\boldsymbol{\beta}}$ remains unchanged, $\widehat{\boldsymbol{\beta}}^{(l)} = \widehat{\boldsymbol{\beta}}^{(l-1)}$.
   - If $l(\widehat{\boldsymbol{\alpha}}^{(l)}, \widehat{\boldsymbol{\beta}}^{(l-1)}) \leq l(\widehat{\boldsymbol{\alpha}}^{(l-1)}, \widehat{\boldsymbol{\beta}}^{(l)})$ then $\widehat{\boldsymbol{\beta}}^{(l)}$ is updated and $\widehat{\boldsymbol{\alpha}}$ remains unchanged, $\widehat{\boldsymbol{\alpha}}^{(l)} = \widehat{\boldsymbol{\alpha}}^{(l-1)}$.

Further Details

If the transformation $h_T(.)$ in the generating model is considered as a response function it has to be monotone. The approximating transformation is given by $h_0(\widehat{h}(.))$ where the outer function $h_0(.)$ is already a monotonically increasing link function. In order to obtain

a monotonically increasing response function $h_0(\widehat{h}(.))$ we have to restrict the estimation of $\widehat{h}(.)$ by a monotonicity constraint.

A sufficient condition for the B-Spline basis expansion to be monotonically increasing is that the components of the coefficient vector $\boldsymbol{\alpha}$ are ordered such that $\alpha_i \leq \alpha_{i+1}$ holds. In boosting methods this inequality must hold after every update step, $\widehat{\alpha}_i^{(l-1)} + \widehat{a}_i^{(l)} \leq \widehat{\alpha}_{i+1}^{(l-1)} + \widehat{a}_{i+1}^{(l)}$. Therefore we constrain every update step $\widehat{\boldsymbol{a}}$ to be from

$$\mathcal{A} = \left\{ \boldsymbol{a}^{(l)} : a_2^{(l)} - a_1^{(l)} \geq \widehat{\alpha}_1^{(l-1)} - \widehat{\alpha}_2^{(l-1)}, \ldots, a_k^{(l)} - a_{k-1}^{(l)} \geq \widehat{\alpha}_{k-1}^{(l-1)} - \widehat{\alpha}_k^{(l-1)} \right\}. \tag{5.9}$$

Monotone functions can be obtained in several ways. After computing $\widehat{\boldsymbol{\alpha}}^{(l)}$ in the lth step one can monotonize the components by use of isotone regression, provided for example by the R-routine `isoreg`. Alternatively, one can solve the optimization problem that is behind the Fisher step in (5.3) with the additional restriction that $\widehat{\boldsymbol{a}}$ is from $\mathcal{A}$. Therefore one minimizes

$$\boldsymbol{a}^T \boldsymbol{\Phi}^T \widehat{\boldsymbol{W}}^{(l-1)} \boldsymbol{\Phi} \boldsymbol{a} - 2 \boldsymbol{a}^T \boldsymbol{\Phi}^T \widehat{\boldsymbol{W}}^{(l-1)} (\widehat{\boldsymbol{D}}^{(l-1)})^{-1} (\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^{(l-1)}), \text{ s.t. } \boldsymbol{a} \in \mathcal{A}$$

where $\widehat{\boldsymbol{W}}^{(l-1)} = (\widehat{\boldsymbol{D}}^{(l-1)})^T (\widehat{\boldsymbol{\Sigma}}^{(l-1)})^{-1} \widehat{\boldsymbol{D}}^{(l-1)}$. Solutions can be obtained by use of the R-package `quadprog` (see Turlach, 2009) which is able to solve a quadratic optimization problem with linear constraints. Results are very similar. In our applications we use `quadprog`. For the use of similar constraints see also Gertheiss et al. (2009).

In step 3 of the algorithm a selection step is included in which it is determined if the coefficients of parameters or the link function is updated. We tried several alternatives, but updating one of the sets of coefficients turned out to be most efficient.

Choice of Tuning Parameter

There are two tuning parameters in the model: the number of boosting iterations m which mainly steers variable selection and λ which controls the smoothness of the response function and the number of response function updates. For determining the appropriate pair of tuning parameters $\boldsymbol{\pi} = (m, \lambda)$ we use K-fold cross validation (CV). There are several reasons to use this procedure and not to work with information-based criteria as used for example by Tutz and Leitenstorfer (2011). On the one hand Hastie (2007) suggests to use CV in boosting procedures because the effective degrees of freedom can be underestimated by using the trace of the hat-matrix. On the other hand the complexity of a SIM fit involves two stages, first the complexity of the predictor fit, and second the fit of the response function. Therefore, a hat matrix that includes both stages is not available. In addition, the two restrictions (monotonicity of the response function and normalization of $\boldsymbol{\beta}$) make the problem of finding appropriate hat matrices even more difficult.

In K-fold cross validation the data set is split K-times into a test data set of size n/K and a training data set of size $n - n/K$. For every tuple of tuning parameters $\boldsymbol{\pi}$ the model is fitted on the κ-th training data set obtaining $\widehat{\boldsymbol{\gamma}}_{\boldsymbol{\pi}}^{\kappa} = (\widehat{\boldsymbol{\alpha}}_{\boldsymbol{\pi}}^{\kappa}, \widehat{\boldsymbol{\beta}}_{\boldsymbol{\pi}}^{\kappa})$. Then the predictive deviance

on the κ-th test set $\text{Dev}(\boldsymbol{y}^{\kappa}_{\text{test}}, \widehat{\boldsymbol{\mu}}(\widehat{\boldsymbol{\gamma}}^{\kappa}_{\boldsymbol{\pi}}|\boldsymbol{X}^{\kappa}_{\text{test}}))$ is computed. The final $\boldsymbol{\pi}_{opt}$ is determined by

$$\boldsymbol{\pi}_{opt} = \text{argmin}_{\boldsymbol{\pi}\in\mathcal{M}\times\Lambda}\left\{\sum_{\kappa=1}^{K}\text{Dev}(\boldsymbol{y}^{\kappa}_{\text{test}}, \widehat{\boldsymbol{\mu}}(\widehat{\boldsymbol{\gamma}}^{\kappa}_{\boldsymbol{\pi}}|\boldsymbol{X}^{\kappa}_{\text{test}}))\right\} \tag{5.10}$$

Unless otherwise mentioned we choose $m \in \mathcal{M} := \{1, ..., 1000\}$ and $\lambda \in \Lambda := \{0.01, 0.1, 1, 10, 100\}$.

5.3 Simulation Studies

Measures of Model Assessment

Some care should be taken when estimates are compared. We assume $\mu_i = h_T(\boldsymbol{x}_i^T\boldsymbol{\beta})$ where $h_T(.)$ is the unknown true transformation function and for the true parameter (without intercept) $||\boldsymbol{\beta}|| = 1$ holds and the magnitude of $||\boldsymbol{\beta}||$ as well as the intercept are absorbed into $h_T(.)$. Let the generating model without restrictions be given by $\mu_i = h_G(\beta_0 + \boldsymbol{x}_i^T\boldsymbol{\beta}_0)$ with unrestricted parameter vector $\boldsymbol{\beta}_0$, where h_G is any monotone function. Then the model can always be rewritten in the corresponding standardized true response function $h_T(.)$ by

$$\mu_i = h_G(\beta_0 + \boldsymbol{x}_i^T\boldsymbol{\beta}_0) = h_G(\beta_0 + ||\boldsymbol{\beta}_0||(\boldsymbol{x}_i^T\boldsymbol{\beta}_0/||\boldsymbol{\beta}_0||)) = h_T(\eta_i),$$

with $\eta_i = \beta_0 + ||\boldsymbol{\beta}_0||\eta_i$, $\eta_i = \boldsymbol{x}_i^T\boldsymbol{\beta}$, $\boldsymbol{\beta} = \boldsymbol{\beta}_0/||\boldsymbol{\beta}_0||$. In particular when a given link function like the canonical link is used, estimates cannot be compared directly to the parameters $||\boldsymbol{\beta}_0||$ for some generating link function $h_G(.)$. Therefore estimated parameters are also standardized and one considers $\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}_{can}/||\widehat{\boldsymbol{\beta}}_{can}||$, where $\widehat{\boldsymbol{\beta}}_{can}$ is the estimate resulting from the canonical link model.

Comparisons in this article always refer to corresponding standardized estimates $\widehat{\boldsymbol{\beta}}$. Therefore the difference between $\boldsymbol{\beta}$ and $\widehat{\boldsymbol{\beta}}$ is measured by

$$\text{MSE}_{\boldsymbol{\beta}} = ||\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}||^2,$$

where $||\boldsymbol{\beta}|| = 1$ and $||\widehat{\boldsymbol{\beta}}|| = 1$. In addition, the accuracy of prediction is investigated by use of the predictive deviances on an independent test set

$$\text{Dev}_{\text{test}} = \text{Dev}(\boldsymbol{y}_{\text{test}}, \widehat{\boldsymbol{\mu}}(\widehat{\boldsymbol{\gamma}}_{\boldsymbol{\pi}_{opt}}|\boldsymbol{X}_{\text{test}})).$$

The number of observations in the test data set is chosen by $n_{test} = 5 \cdot n_{train}$.

Procedures and Results

We compare our procedure with three other procedures:

- The boosting procedure `mboost` with canonical link function (see Hothorn et al., 2009, 2010).

- $L1$ penalized GLM with canonical link function computed by the `R`-package `lasso2` (see Lokhorst et al., 2007).
- The ML-estimator with canonical link function for the full model.

In addition, we include a modified version of FlexLink which truncates small coefficients to zero. Although the FlexLink algorithm is able to return null estimates, it can happen that some variables are selected just once or twice and the corresponding estimates have non-zero but very small values. In particular, false positive error rates improve by setting these small values to zero. For details of the modified algorithm, which is referred to as FlexLink (cut), see Section 5.4.

The predictor matrix was generated as a $N(\mathbf{0}_p, \boldsymbol{\Sigma})$-distribution with $\boldsymbol{\Sigma} = \{\sigma_{ij}\}_{i,j\in\{1,\ldots,p\}}$ where $\sigma_{ij} = 0.5,\ i \neq j,\ \sigma_{ii} = 1$. We use two parameter vectors with $p = 20$,

$$\begin{aligned} \boldsymbol{\beta}_a &= (0.2,\ 0.4,\ -0.4,\ 0.8,\ 0,\ \ldots,\ 0)^T, \\ \boldsymbol{\beta}_b &= (0.5,\ 0.5,\ -0.5,\ -0.5,\ 0,\ \ldots,\ 0)^T, \end{aligned}$$

to generate $\eta = \boldsymbol{X}\boldsymbol{\beta}$. As distributions of the response we consider normal, Poisson and binomial distribution. Further we consider two different response functions for every distribution. So 12 different simulations settings were investigated. They are denoted in the following way, ⟨dis⟩ ⟨resp⟩ ⟨beta⟩. For example, the setting Bin2b has binomial distributed response, uses the second response function and $\boldsymbol{\beta}_b$ is the true parameter vector. The true response functions that are used below, an approximation by the canonical link and the 50 estimated response function are shown in Figure 5.2. The estimated response functions are for the case $\boldsymbol{\beta}_a$. First we will describe the considered distributions and then discuss results.

(1) *Normal Distribution*

In the normal distribution case we use the response functions

1. $h_T(\eta) = 3 \cdot \eta^3$
2. $h_T(\eta) = \text{sgn}(\eta) 5 \cdot \sqrt[3]{\eta}$

which are shown in the first row of Figure 5.2. In addition an approximation of $h_T(.)$ to the canonical response function, which in this case is linear, is shown Therefore, $h_{can}(\eta) = a + b \cdot \eta$ is computed where a and b are chosen to minimize $\int_{-2}^{2}(h(\eta) - h_{can}(\eta))^2 d\eta$. The approximation is shown by the grey line. For the first first response function the error term is $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, 9\boldsymbol{I})$ and for the second $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \boldsymbol{I})$ where $\boldsymbol{I} = \text{diag}\, 1_{i=1}^{n}$.

(2) *Poisson Distribution*

In the Poisson case we consider the response functions:

1. $h_T(\eta) = \dfrac{10}{1 + \exp(-5 \cdot \eta)}$
2. $h_T(\eta) = \dfrac{10}{1 + \exp(-10 \cdot \eta - 10)} + \dfrac{10}{1 + \exp(-10 \cdot \eta + 10)}$

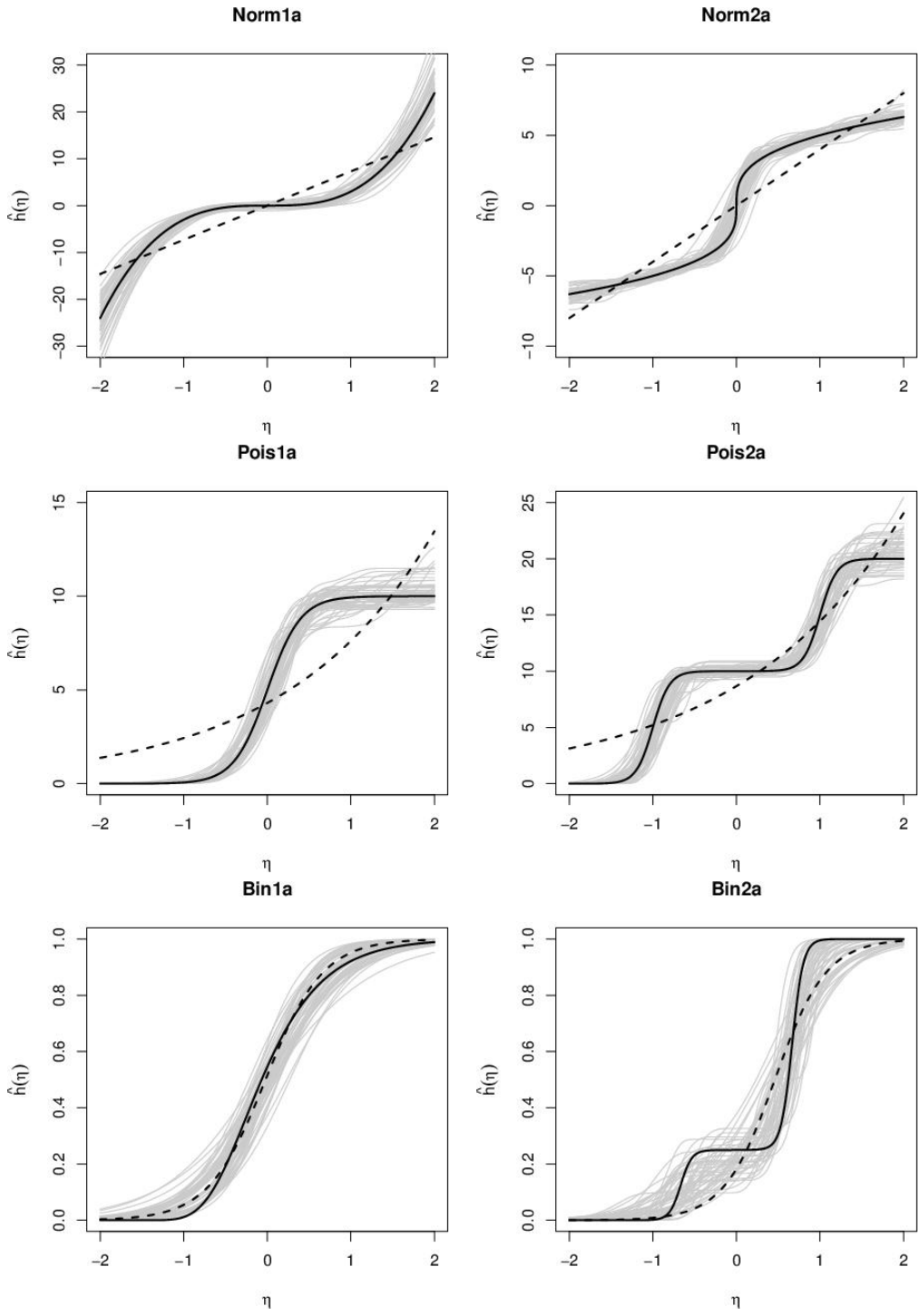

FIGURE 5.2: *True response functions (black lines), approximating canonical response functions (dashed lines) and estimated response functions (grey) of simulation study.*

They are shown in the second row of Figure 5.2. Also the approximation of $h_T(.)$ by the canonical response function is given.

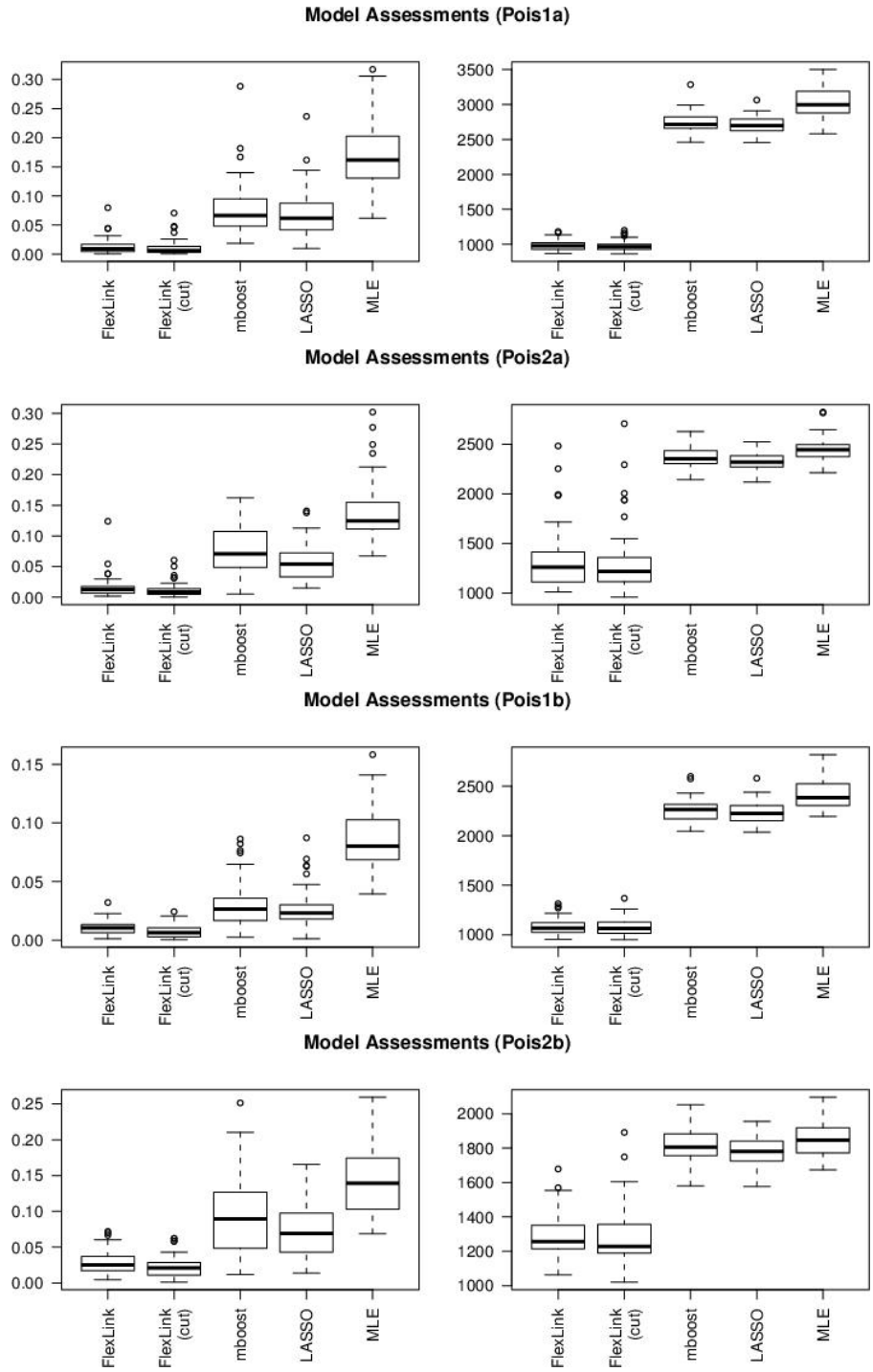

FIGURE 5.3: *Boxplots of model assessment measurements* MSE_β *(left) and* Dev_{test} *(right) in the Poisson case.*

(3) *Binomial Distribution*
For binomial responses the true response functions are

1. $h_T(\eta) = \exp(-\exp(-2 \cdot \eta - 0.5))$,
2. $h_T(\eta) = \dfrac{0.25}{1+\exp(-15 \cdot \eta - 10)} + \dfrac{0.75}{1+\exp(-15 \cdot \eta + 10)}$.

Figure 5.2 shows the response function and the approximating canonical response function. The second response function corresponds to the Gumbel-link and can be approximated by the canonical logit-link quite well.

The results of the simulations are summarized in Table 5.1. Box plots are given only for the Poisson distribution in Figure 5.3. Performance in terms of MSE and predictive deviance is about the same as for FlexLink. It is seen from Table 5.1 and Figure 5.3 that for the normal and the Poisson distribution FlexLink (and FlexLink (cut)) distinctly outperform LASSO and mboost in all settings. MSE as well as prediction is strongly improved by allowing for flexible link functions and selection of variables.

The picture is not so clear for binomial distributions. The binomial case is a challenge for the estimation of the unknown link function because the information in $0-1$ observations is weak and the true link function does not differ so much from the canonical link. It is seen that all the selection procedures yield better results than MLE but there is not so much difference among the selection procedures. Nevertheless, the flexible procedure yields the best results in three of the four scenarios in terms of MSE. Of course artificial response functions can be constructed that are far from the canonical link and therefore will favour flexible procedures. But we preferred to use link functions which are not too strange. Even in the case of link functions that are not too far from the canonical link the flexible procedure is a strong competitor.

5.4 Modified Estimator and Selection of Predictors

MSE and predictive deviance are important criteria in the comparison of fitting procedures. However, in selection procedures the performance should also refer to the precision of the selection. Criteria by which selection can be measured are in particular hit rate (proportion of correctly identified influential variables) and false positives (proportion of non-influential variables dubbed influential).

One problem with simple boosting procedures is that some predictors are selected just once or twice. The corresponding estimated parameters are very small but are not equal to zero. Performance of selection can be easily improved by cutting off these small values. In the procedure called FlexLink (cut) we use a truncated version of $\widehat{\boldsymbol{\beta}}$. The components of estimate $\widehat{\boldsymbol{\beta}}$ are compared to $1/p$, where p is the number of predictors, and components that are smaller than $1/p$ are set to zero. Then the new estimate is re-standardized to have Euclidean norm 1. When used in the cross-validation procedure (5.10) one obtains the new optimal tuning parameter $\tilde{\boldsymbol{\pi}}_{opt}$. Of course, the threshold for the cut-off could be optimized, we simply tried the threshold $1/p$ and found that it worked well.

		FlexLink	FlexLink (cut)	mboost	LASSO	MLE
Normal distribution						
Norm1a	MSE_{β}	0.0072	0.0046	0.0620	0.2705	0.1224
	Dev(test)	0.7493	0.7778	3.4226	4.3310	3.5184
Norm2a	MSE_{β}	0.0044	0.0018	0.0163	0.0159	0.0419
	Dev(test)	2.8895	2.8265	9.1622	9.0386	9.5827
Norm1b	MSE_{β}	0.0439	0.0401	0.0604	0.0446	0.1431
	Dev(test)	19.3224	19.4391	27.5937	28.0246	28.3730
Norm2b	MSE_{β}	0.0032	0.0011	0.0065	0.0080	0.0224
	Dev(test)	4.5385	4.3072	12.0554	12.0587	12.5424
Poisson distribution						
Pois1a	MSE_{β}	0.0092	0.0063	0.0664	0.0615	0.1619
	Dev(test)	979.60	966.41	2711.83	2696.37	2995.60
Pois2a	MSE_{β}	0.0123	0.0088	0.0708	0.0539	0.1246
	Dev(test)	1262.12	1218.33	2354.94	2320.07	2445.86
Pois1b	MSE_{β}	0.0105	0.0065	0.0266	0.0232	0.0803
	Dev(test)	1067.16	1063.71	2265.70	2226.17	2384.33
Pois2b	MSE_{β}	0.0253	0.0208	0.0896	0.0691	0.1395
	Dev(test)	1256.51	1229.08	1806.42	1780.39	1846.76
Binomial distribution						
Bin1a	MSE_{β}	0.0761	0.0804	0.0797	0.0843	0.1798
	Dev(test)	813.30	809.57	802.25	796.64	886.73
Bin2a	MSE_{β}	0.0732	0.0734	0.0905	0.0800	0.2197
	Dev(test)	760.35	761.97	818.05	789.27	1336.62
Bin1b	MSE_{β}	0.0836	0.0788	0.0610	0.0719	0.1515
	Dev(test)	981.58	982.99	967.42	979.92	1070.33
Bin2b	MSE_{β}	0.0904	0.0948	0.0939	0.0930	0.1938
	Dev(test)	988.73	998.56	1029.98	986.38	1411.32

TABLE 5.1: *Medians of the model assessment measures for the settings of the simulation study.*

As is seen from Table 5.2, which gives the means of hits and false positive rates for all settings, the truncated version of FlexLink shows distinct improvement. False positive rates are much smaller, hit rates are in most cases the same as in simple FlexLink, or slightly smaller. Comparison to mboost and LASSO are strongly in favour of FlexLink. The effect is illustrated in Figure 5.4 where hits and false positive rates for one setting are plotted in a ROC-type way. The best performance would be the point (false positive rate, hit rate)=(0,4). Among the considered procedures FlexLink (cut) shows the best approximation to the optimal point.

In our (not yet optimized implementation) computational cost is not too high. For fixed tuning parameter λ the search across 100 boosting iterations to select the best one takes 15 sec for the noisy miner data considered in the next section. The function `gl1ce` from package `lasso2`, which fits the LASSO, takes about 10 sec if optimized over 100 tuning

parameters. The boosting procedure `mboost`, which computes fits for the canonical link, is much faster.

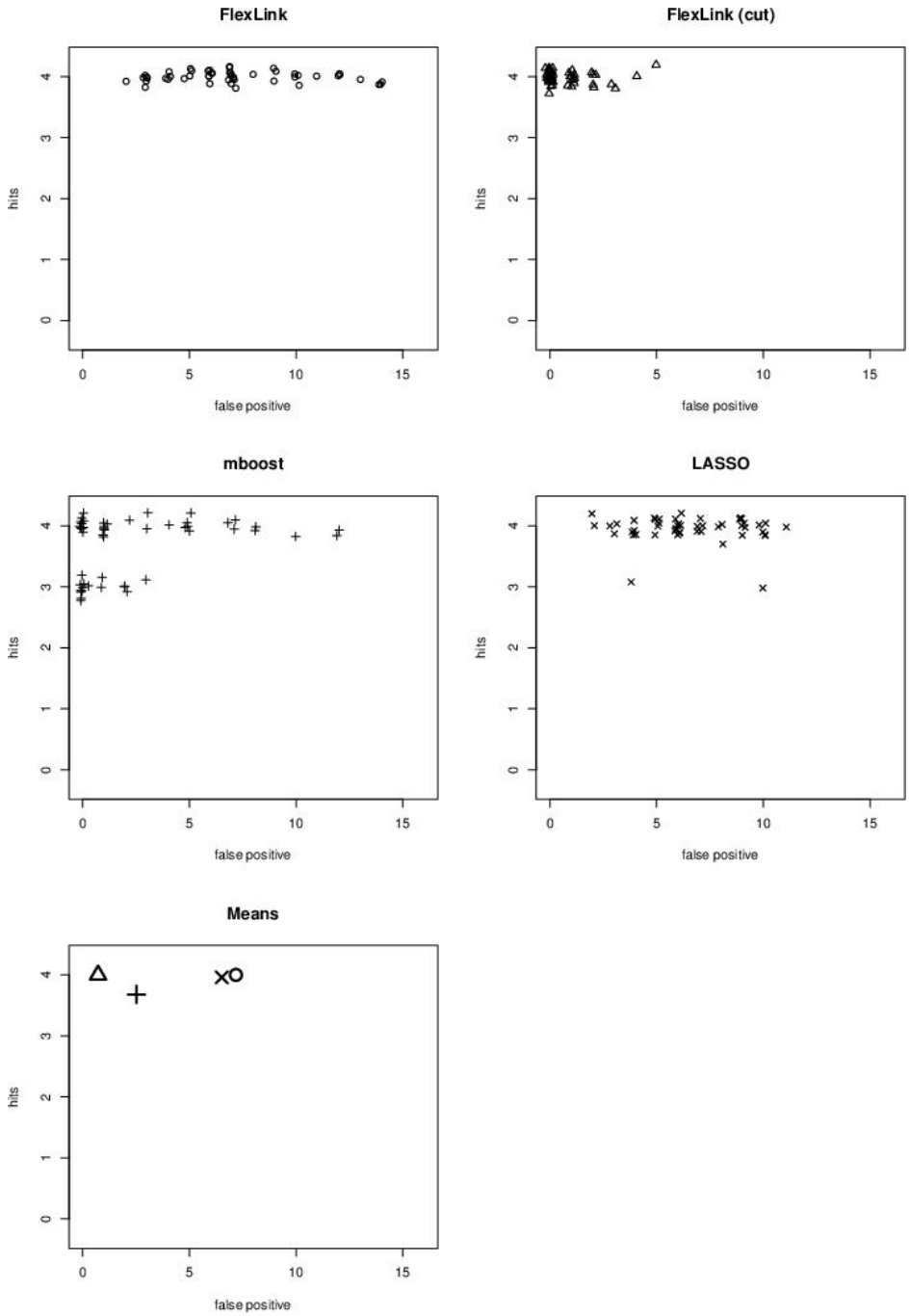

FIGURE 5.4: *Hits and false positive rates for setting Pois2a with jittered values. Last panel shows the means over simulations.*

	FlexLink		FlexLink (cut)		mboost		LASSO	
	hits	false pos.	hits	false pos.	hits	false pos.	hits	false pos.
Normal case								
Norm1a	1.000	0.504	1.000	0.015	0.980	0.355	0.530	0.004
Norm2a	1.000	0.445	1.000	0.006	1.000	0.369	1.000	0.406
Norm1b	1.000	0.575	1.000	0.158	1.000	0.376	1.000	0.165
Norm2b	1.000	0.504	1.000	0.000	1.000	0.355	1.000	0.431
Poisson case								
Pois1a	0.995	0.370	0.990	0.038	0.965	0.265	0.980	0.326
Pois2a	1.000	0.449	1.000	0.045	0.920	0.158	0.990	0.408
Pois1b	1.000	0.439	1.000	0.025	1.000	0.283	1.000	0.309
Pois2b	1.000	0.665	1.000	0.104	1.000	0.223	1.000	0.456
Binomial case								
Bin1a	0.915	0.244	0.870	0.100	0.960	0.366	0.975	0.409
Bin2a	0.980	0.306	0.955	0.133	0.975	0.390	0.985	0.430
Bin1b	1.000	0.304	1.000	0.128	1.000	0.401	1.000	0.451
Bin2b	1.000	0.394	1.000	0.160	1.000	0.405	0.446	1.000

TABLE 5.2: *Means of the hits and false positive rates.*

5.5 Applications

5.5.1 Medical Care Data

In this section, we consider the health care data from Dep and Trivedi (1997). The original data is from the US National Medical Expenditure Survey and is available from the data archive of the Journal of Applied Econometrics (`http://www.econ.queensu.ca/jae/1997-v12.3/deb-trevidi/`). We use the `data.frame` from Zeileis (2006). The response variable that is considered is the *number of physician office visits* (`ofp`), which potentially depends on the variables given in Table (5.3). In our investigation we use only male patients, which reduces the sample size to $n = 1778$ from the total available sample of 4406 individuals.

We compare the same estimating procedures as in Section 5.3. For measuring the prediction performance, 25 splits into a training data set of $n_{train} = 1185$ and a test data set of $n_{test} = 593$ were used. For reducing the computational costs, the tuning parameter λ of the FlexLink is determined by 5fold cross validation on the complete data set to $\lambda = 100$. So only the number of optimal boosting iterations has to determined inner each split. Figure 5.5 shows the predictive deviances in the test data and the fitted link functions (for male patients which visit physician office maximum 30 times). It is seen

that the link function for the flexible model differs from the canonical link in particular for large values of the linear predictor. While the canonical link still increases distinctly, the flexible link is very flat. The estimated link functions are very stable across splits. It is also seen from Figure 5.5 that prediction for the flexible model with variable selection distinctly outperforms the competitors. From Table 5.4 it is seen that the flexible link procedure reduces the number of coefficients.

Label	Explanation
`exclhlth`	= 1 if self-perceived health is excellent
`poorhlth`	= 1 if self-perceived health is poor
`numchron`	number of chronic conditions (cancer, heart attack, gall bladder problems, emphysema , arthritis, diabetes, other heart disease)
`adldiff`	= 1 if the person has a condition that limits activities of daily living
`noreast`	= 1 if the person lives in northeastern US
`midwest`	= 1 if the person lives in the midwestern US
`west`	= 1 if the person lives in the western US
`age`	age in years divided by 10
`black`	= 1 if the person is African American
`married`	= 1 if the person is married
`school`	number of years of education
`faminc`	family income in $10 000
`employed`	= 1 if the person is employed
`privins`	= 1 if the person is covered by private health insurance
`medicaid`	= 1 if the person is covered by Medicaid

TABLE 5.3: *Variable description for medical care data*

The estimated parameters are given in Table 5.4. Since data are strongly overdispersed ($\widehat{\Phi} = 7.736$) we give quasi-likelihood estimates (QLE) instead of the maximum likelihood estimates. It is seen that all covariates with p-values smaller than 0.05 for QLE were selected by FlexLink an FlexLink (cut). The latter procedures select two more covariates, covariate 7 and 10. In Figure 5.6 we show the error bars across 300 bootstrap samples. The circles mark the parameter estimate from Table 5.4 and the whiskers are the 0.975- and 0.025-quantiles determined by bootstrapping. We used simple pairwise bootstrap. The data contains $n = 1778$ pairs $(y_i, \boldsymbol{x}_i)$, $i = 1, ..., n$, where y_i is the response value and $\boldsymbol{x}_i$ is the corresponding vector of covariates. We sample $b = 300$ bootstrap samples. Each bootstrap sample is sampled by drawing n pairs $(y_i, \boldsymbol{x}_i)$ with replacement. We achieve $(\boldsymbol{y}_b^*, \boldsymbol{X}_b^*)$, $b = 1, ..., 300$, bootstrap samples with n observations whereby some observations are equal. Then we fit models on $(\boldsymbol{y}_b^*, \boldsymbol{X}_b^*)$, $b = 1, ..., 300$, and achieve the corresponding estimates $\widehat{\boldsymbol{\beta}}_b^*$. Finally we computed the quantiles of the distribution of the components of

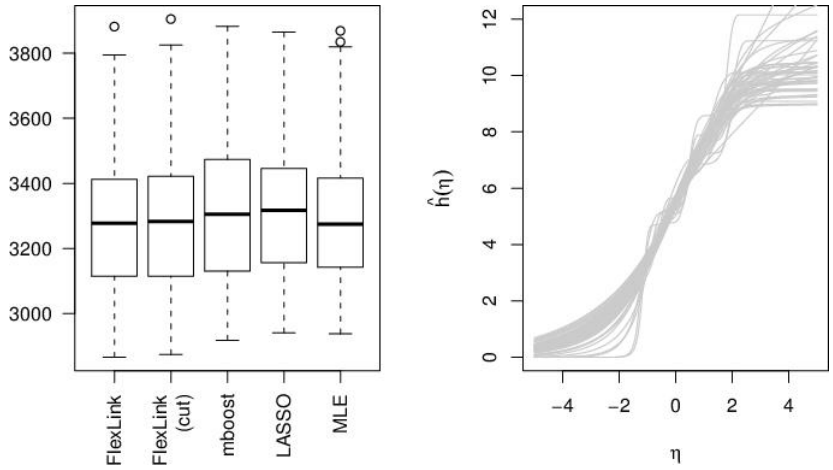

FIGURE 5.5: *Left panel: Boxplots of predictive deviance on test data sets across 50 random splits of medical care data. Right panel: Estimated response function for medical care data across 50 random splits.*

Number	Variable	FlexLink	FlexLink (cut)	mboost	LASSO	QLE (p-value)
1	exclhlth	0.0000	0.0000	-0.2023	-0.0382	-0.1979 (0.050)
2	poorhlth	0.2083	0.3614	0.3203	0.2546	0.3134 (0.000)
3	numchron	0.9492	0.8849	0.7042	0.7971	0.6854 (0.000)
4	adldiff	0.0000	0.0000	0.0250	0.0000	0.0258 (0.766)
5	noreast	0.0000	0.0000	-0.1623	-0.0916	-0.1318 (0.177)
6	midwest	0.0000	0.0000	0.0451	0.0000	0.0636 (0.483)
7	west	0.0419	0.0866	0.0725	0.0062	0.0840 (0.359)
8	age	0.0000	0.0000	-0.0592	0.0000	-0.0142 (0.875)
9	black	0.0000	0.0000	-0.1693	0.0000	-0.1276 (0.208)
10	married	0.0364	0.0878	0.1166	0.0734	0.1420 (0.119)
11	school	0.1725	0.1965	0.3949	0.4817	0.4224 (0.000)
12	faminc	0.0000	0.0000	0.0000	0.0000	-0.0064 (0.939)
13	employed	0.0000	0.0000	0.0138	0.0000	0.0254 (0.772)
14	privins	0.1508	0.1801	0.3318	0.2292	0.3555 (0.001)
15	medicaid	0.0000	0.0000	0.1196	0.0000	0.1462 (0.124)

TABLE 5.4: *Parameter estimates for medical care data set.*

estimates $\widehat{\boldsymbol{\beta}}_b^*$, $b = 1, ..., 300$.

It is remarkable that mboost selects nearly all variables. The LASSO and the FlexLink select a similar set of variables. Further by estimating the response function flexibly there is a tendency that the smaller values of y_i are accumulated on the left side which seems to be reasonable for an increasing response function. This effect can not be found for the other procedures.

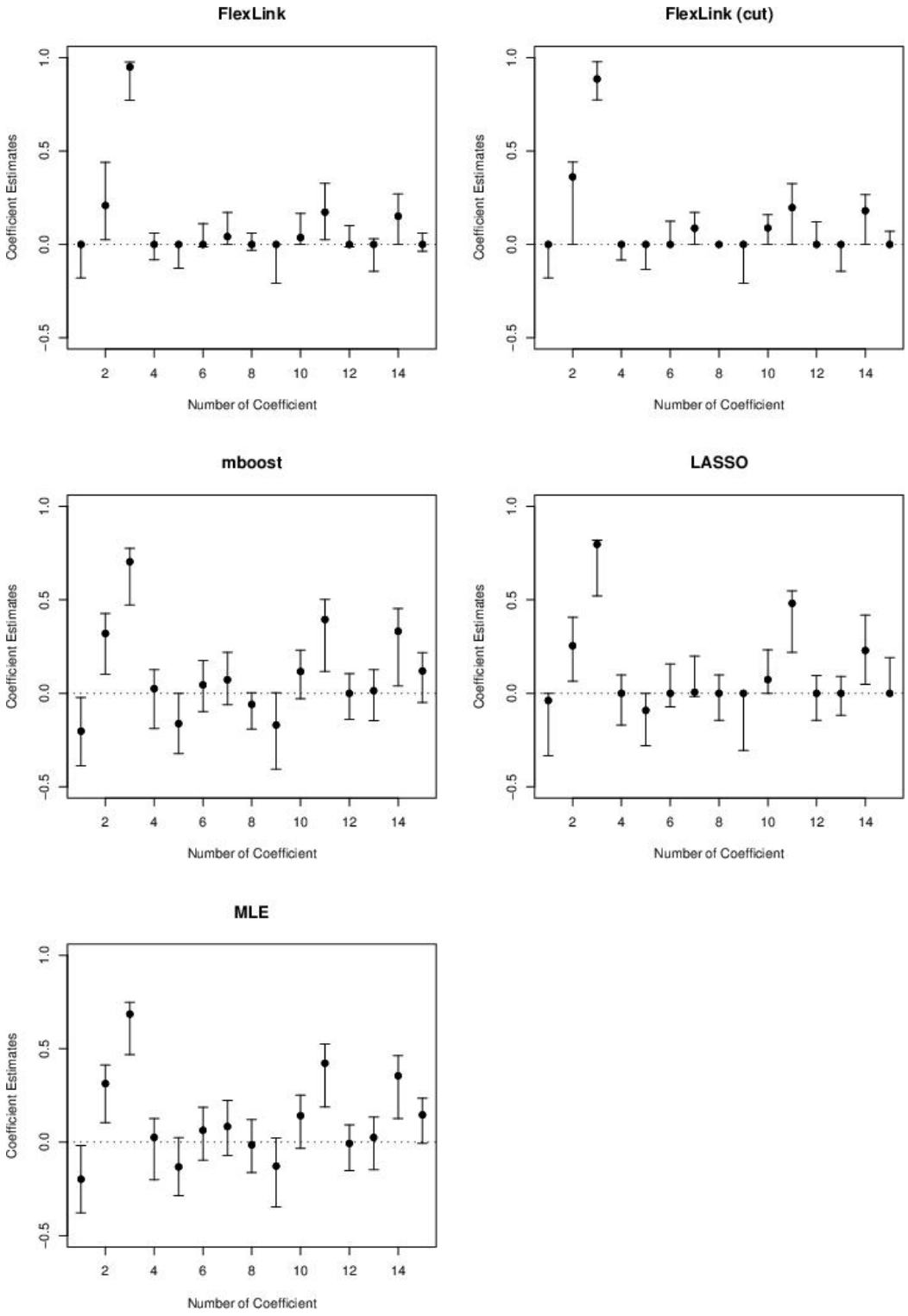

FIGURE 5.6: *The error bar plot for medical care data across 300 bootstrap samples.*

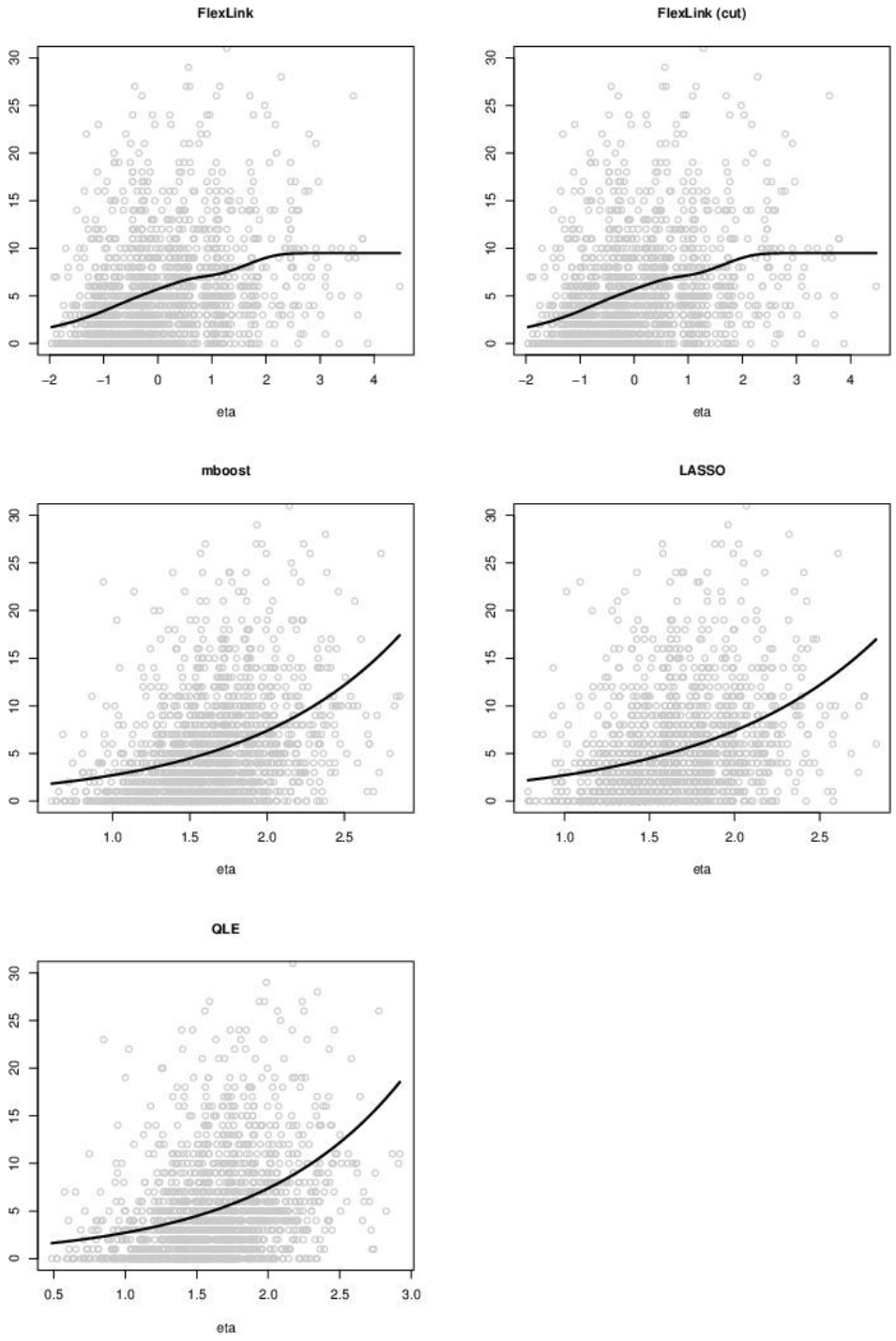

FIGURE 5.7: *Response function of the five procedures with optimal tuning parameter determined by* 5*-fold cross validation and the QLE . Grey circles mark the observed response values* y_i *at the estimated value* $\widehat{\eta}_i$.

5.5.2 Noisy Miner Data

In this section we consider the noisy miner data from Maron (2007), which are available at `http://www.sci.usq.edu.au/staff/dunn/ Datasets/tech-glms.html`. The data set has a biological background. Three 20 minutes surveys were conducted in each of 31 $2ha$ belt transects in buloke woodland patches within the Wimmera Plains of western Victoria, Australia. The considered response is the number of species from a list of birds (*number of declining species*, in short `declinerab`). It is of particular interest how the number of species is affected by the presence of the noisy miner, which is an aggressive competing bird. The collected explanatory variables are given in Table (5.5). Figure 5.8 shows the fitted

Number	Label	Explanation
1	`eucs`	number of eucalypts in each $2ha$ transect
2	`area`	area $[ha]$ of remnant patch vegetation in which the transect was located
3	`grazed`	whether the area was grazed (= 1) or not (= 0)
4	`shrubs`	whether shrubs were present (= 1) or not (= 0)
5	`buloke`	number of buloke trees in each transect
6	`timber`	number of pieces of fallen timber
7	`finelitt`	percentage of fallen litter on the ground
8	`minerab`	number of observed noisy miners

TABLE 5.5: *Variable description of the noisy miner data.*

response function together with the approximated canonical link function. It is seen that, in particular for large values of the linear predictor, the two link functions differ strongly; in that area the flexible response function is much steeper than the canonical response function. The prediction performance is measured by using 50 splits into a training data set of $n_{train} = 21$ and a test data set of $n_{test} = 10$. The tuning parameter λ of the FlexLink was determined by 5-fold cross validation. Figure 5.8 shows the predictive deviances in the test data. It is seen that prediction for the flexible model with variable selection distinctly outperforms the competitors. Table 5.6 shows parameter estimates for the various models. It turned out that Flex Link selects one variable, namely the number of noisy miners, which seems to be responsible for the decrease in species. In contrast, mboost selects five predictors and LASSO three. Since the data are strongly overdispersed ($\widehat{\Phi} = 4.647$) we used the quasi-likelihood estimator (QLE) instead of the MLE. QLE also suggests that only one variable in the linear predictor is relevant. The estimates turn out to be very stable for the flexible link procedure (not shown). It seems that the exclusion of spurious predictors together with the fitting of a link function that is far from the canonical link makes the flexible procedure the dominating procedure in this example. This is in line with the simulation results where spurious predictors were included and the link was different from the standard link.

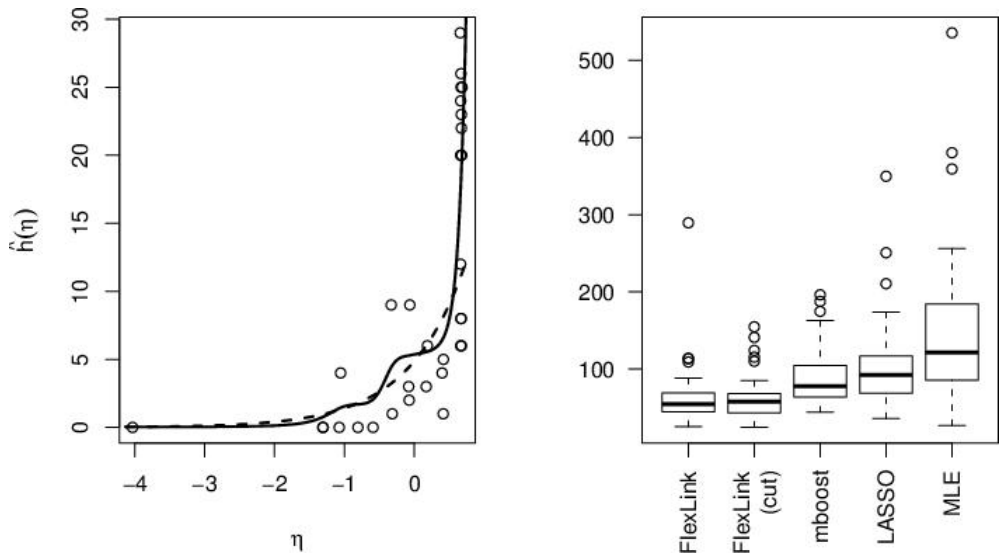

FIGURE 5.8: *Estimated response function for the noisy miner data and approximation by the canonical response function (left panel), box plots for deviances over 50 random splits (right panel)*

Number	Variable	FlexLink	FlexLink (cut)	mboost	LASSO	QLE (p-value)
1	eucs	0.0000	0.0000	0.0000	0.0000	0.0557 (0.264)
2	area	0.0000	0.0000	-0.0494	-0.0726	-0.0249 (0.168)
3	grazed	0.0000	0.0000	0.0000	0.0000	-0.5010 (0.430)
4	shrubs	0.0000	0.0000	0.1074	0.0000	-0.2569 (0.717)
5	buloke	0.0000	0.0000	0.0938	0.0505	0.0032 (0.345)
6	timber	0.0000	0.0000	0.0720	0.0000	0.0024 (0.896)
7	finelitt	0.0000	0.0000	0.0000	0.0000	0.0023 (0.910)
8	minerab	-1.0000	-1.0000	-0.9859	-0.9961	-0.8242 (0.001)*

TABLE 5.6: *Standardized parameter estimates for the whole noisy miner data set normed to length equal to 1.*

5.6 Concluding Remarks

A flexible estimation of the response function combined with variable selection is proposed. It has been demonstrated that the method improves parameter estimation and prediction in the presence of irrelevant variables. The method works for generalized linear models, improvement is usually strong, but less impressive for binary responses where information is weak. The modified version FlexLink (cut) shows much better variable selection performance without suffering in accuracy concerning estimation and prediction.

We focused on the estimation of link functions for generalized linear models and therefore included a monotonicity restriction for the response function. In future work the mono-

tonicity restriction will be dropped, resulting in generalized single-index models (compare Cui et al., 2009). Then information-based criteria like AIC can be used since the hat matrix of boosting can be derived (for AIC in single-index models compare Naik and Tsai 2001). The use of information-based criteria is attractive because it could reduce computational costs.

Chapter 6

Estimating Generalized Additive Models with Flexible Link Function Including Variable Selection

A natural extension of the Flex Link Algorithm (see chapter 5 or Tutz and Petry (2011)) is the flexibilization of the linear predictor $\eta_i = \boldsymbol{x}_i^T\boldsymbol{\gamma}$ to an additive predictor $\eta_i = \sum_{j=1}^{p} f_j(x_{ij})$. Since additionally to the more flexible predictor the response function is unknown and has to be estimated. This leads to the class of generalized additive models with flexible response function. We will combine the Flex Link Algorithm with componentwise boosting of smooth functions proposed by Tutz and Binder (2006) for solving the new modeling problem.

6.1 Introduction

Methods for the estimation of the unknown link function with linear predictors have been proposed under the name single index models (SIMs). Let data be given by $(y_i, \boldsymbol{x}_i)$, $i = 1, \ldots, n$, where y_i denotes the response and $\boldsymbol{x}_i = (x_{i1}, ..., x_{ip})^T$ the vector of p covariates. In SIMs as discussed, for example, by Weisberg and Welsh (1994), Ruckstuhl and Welsh (1999), Härdle et al. (1993), and Muggeo and Ferrara (2008), the conditional expectation of y_i given $\boldsymbol{x}_i$, $y_i|\boldsymbol{x}_i$, is modeled by

$$E(y_i|\boldsymbol{x}_i) = \mu_i = h_T(\eta_i),$$

where $h_T(.)$ is the unknown response function and $\eta_i = \boldsymbol{x}_i^T\boldsymbol{\gamma}$ is the predictor which contains no intercept. The unknown $h_T(.)$ has to be estimated nonparametrically. Muggeo and Ferrara (2008) used a P-Spline approach whereas Härdle et al. (1993) used kernel functions. For uniqueness, typically the Euclidean norm of the parameter vector $\boldsymbol{\gamma}$ is fixed at 1, that is, $\|\boldsymbol{\gamma}\| = 1$. A multiplicative factor of $\boldsymbol{\gamma}$ and the intercept are absorbed into the response function $h_T(.)$. To guarantee uniqueness of the estimates of a SIM a second constraint is necessary. Yu and Ruppert (2002) and Cui et al. (2009) set one specific component of $\boldsymbol{\gamma}$ to

be positive. Alternatively, a monotonicity restriction on the response function guarantees uniqueness. As in Tutz and Petry (2011) we will consider only monotonically increasing (isotone) response functions, $\partial h_T(t)/\partial t > 0$. This induces invertibility of the response function, which is fundamental in generalized linear models (GLMs). The inverse of the response function is the link function, $h_T^{-1}(.) = g_T(.)$. With the monotonicity constraint SIMs are equivalent to GLMs with unknown response function. Typically, when GLMs are used the response function is considered as fixed and known. In most applications the canonical response function $h_0(.)$ is chosen for $h_T(.)$. However, among others, Czado and Santner (1992) showed that misspecified response functions can lead to a substantial bias in the estimate of $\boldsymbol{\gamma}$.

The same holds for the more general class of generalized additive models (GAMs). In contrast to GLMs, where the predictor is a linear combination of covariates, in GAMs the predictor is a sum of covariate specific unspecified functions. The conditional expectation is modeled by a transformation $h_T(.)$ of a sum of covariate specific functions and an intercept β_0 in the form

$$\mu_i = E(y_i|\boldsymbol{x}_i) = h_T(\beta_0 + \sum_{j=1}^{p} f_j(x_{ij})), \text{ s.t. } \int_{\min\{\boldsymbol{x}_j\}}^{\max\{\boldsymbol{x}_j\}} f_j(t)dt = 0,\ j = 1, ..., p. \tag{6.1}$$

The constraint $\int_{\min\{\boldsymbol{x}_j\}}^{\max\{\boldsymbol{x}_j\}} f_j(t)dt = 0$, $j = 1, ..., p$ is needed to obtain uniqueness because a shift of a function $\widetilde{f}_j(.) = f_j(t) + c_j$ can be compensated by a shift of the intercept $\widetilde{\beta}_0 = \beta_0 - c_j$. An extensive discussion of GAMs was given by Hastie and Tibshirani (1990) and Wood (2006).

As in GLMs, in GAMs usually the canonical response function $h_0(.)$ is chosen. The focus of this chapter is on GAMs where the response function is unknown and has to be estimated. For the estimation of the functions $f_j(.)$, which are assumed as unknown, we use common tools of approximation. In particular, we use a B-spline basis expansion

$$f_j(x) \approx \boldsymbol{\psi}_j^T(x)\boldsymbol{\beta}_j,\ j = 1, ..., p, \tag{6.2}$$

where $\boldsymbol{\psi}_j(x)$ is the vector of the m_j basis functions evaluated at x and $\boldsymbol{\beta}_j$ is the corresponding coefficient vector. Let $\boldsymbol{\psi}_{ij} := \boldsymbol{\psi}_j(x_{ij})$ denote the vector of basis function expansion of the jth predictor function evaluated at the ith observation, that is, at the data point x_{ij}. B-spline basis expansions have been proposed by De Boor (1978) and have become very popular in many fields (see Eilers and Marx, 1996; Ramsey and Silverman, 2005). The estimating model in the case of GAMs is

$$\mu_i = E(y_i|\boldsymbol{x}_i) = h_T(\beta_0 + \sum_{j=1}^{p} \boldsymbol{\psi}_{ij}^T\boldsymbol{\beta}_j).$$

For a compact notation, let

$$\boldsymbol{\Psi}_j = (\boldsymbol{\psi}_{1j}, ..., \boldsymbol{\psi}_{nj})^T \tag{6.3}$$

denote the $n \times m_j$-dimensional matrix of basis function evaluated at all observation of the jth covariate and

$$\boldsymbol{\Psi} = (\boldsymbol{\Psi}_1, .., \boldsymbol{\Psi}_p) \tag{6.4}$$

denote the $n \times (\sum_{j=1}^{p} m_j)$ total design matrix without intercept.

In GAMs it is assumed that the response is from a simple exponential family

$$f(y_i|\theta_i, \phi) = \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{\phi} + c(y_i, \phi)\right\} \tag{6.5}$$

where θ_i is the natural parameter of the family, ϕ is a dispersion parameter and $b(.)$, $c(.)$ are specific functions corresponding to the type of the family.

Procedures for the fitting of GAMs are typically based on roughness penalties that regularize the regression problem, see, in particular, Hastie and Tibshirani (1990) and Wood (2006). Let $\boldsymbol{\beta} = (\beta_0, \boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_p)^T$ denote the coefficient vector including the coefficients on basis functions as and the intercept, then the corresponding log-likelihood function is

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n}(y_i\theta_i - b(\theta_i))/\phi). \tag{6.6}$$

An often used penalized log-likelihood estimate has the form

$$\widehat{\boldsymbol{\beta}} = \text{argmax}_{\boldsymbol{\beta}}\left\{l(\boldsymbol{\beta}) + \sum_{j=1}^{p}\lambda_j \int \frac{\partial^2}{\partial t^2}\left(\boldsymbol{\psi}_j^T(t)\boldsymbol{\beta}_j\right)^2 dt, \text{ s.t. } \int \boldsymbol{\psi}_j^T(t)\boldsymbol{\beta}_j dt = 0\right\},$$

where λ_j, $j = 1, ..., p$, are tuning parameters which have to be chosen.

For the estimation of high dimensional models it makes sense only to select and the most relevant covariates. Breiman (1996) showed that variable selection can improve the predictive property of an estimated model. Further variable selection tackles the problem of overfitting by including only highly influential covariates in the estimated model. Nowadays many algorithms and strategies for variable selection have been developed.

For GLMs the $L1$-penalization is a very popular way to generate variable selection. It has been introduced by Tibshirani (1996) for linear normal regression. For GLMs computationally effective algorithms to solve the corresponding penalized likelihood problem were given by Goeman (2010a) and Park and Hastie (2007b). Another strategy for variable selection is componentwise boosting with early stopping. Tutz and Binder (2006) and Bühlmann and Hothorn (2007) presented such boosting techniques for GLMs. Also the FlexLink algorithm from Tutz and Petry (2011) are also based on componentwise boosting.

Both techniques can be generalized to GAMs. In the case of penalization predictors are be selected by adding penalty terms which shrink the predictor functions with low influence on the response to a zero line. Such penalty terms were described by Avalos et al. (2007) and Wood (2011). As in GLMs the regularization by componentwise boosting enforces variable selection by early stopping. In each boosting iteration only one covariate is updated. So only variables that have been updated before stopping the algorithm by an

appropriate criterion are included in the model. For details of componentwise boosting see Schmid and Hothorn (2008), Hothorn et al. (2010) and Tutz and Binder (2006).

The advantages of variable selection in GAMs are analogously to the advantages in GLMs. If variables with low or no influence are excluded noise is eliminated and the predictive performance of the estimated model increases. As in GLMs a misspecified response function is impairing the estimates and the predictive performance. If the true link function differs strongly from the assumed link function variable selection becomes biased and prediction suffers. In the following a fitting procedure is proposed that fits the unknown functions of the additive predictor with simultaneous selection of the relevant predictors and estimates the unknown link function.

6.2 Flexible Link with Additive Predictor (FLAP)

The model that is assumed to hold has the form

$$\mu_i = E(y_i|\boldsymbol{x}_i) = h_T(\sum_{j=1}^{p} f_j(x_{ij})). \tag{6.7}$$

For unspecified link function the model is not identifiable because it is equivalent to the model $E(y_i|\boldsymbol{x}_i) = \tilde{h}_T(a \cdot \sum_{j=1}^{p} f_j(x_{ij}) + b)$ for constants a, b and appropriately chosen response function $\tilde{h}_T$. Therefore additional constraints are needed. Typically, SIMs are constrained by $\|\boldsymbol{\beta}\| = 1$ in order to guarantee uniqueness. Similar to this constraint we postulate for a constant $c > 0$

$$\sum_{j=1}^{p} \int_{\min\{\boldsymbol{x}_j\}}^{\max\{\boldsymbol{x}_j\}} f_j(t)^2 dt = c. \tag{6.8}$$

In addition, each predictor function is centered by postulating

$$\int_{\min\{\boldsymbol{x}_j\}}^{\max\{\boldsymbol{x}_j\}} f_j(t)dt = 0,\ j = 1,\ ...,\ p. \tag{6.9}$$

Moreover, we assume that the response function is monotonically increasing, that is,

$$\frac{\partial h_T(t)}{\partial t} \geq 0. \tag{6.10}$$

In summary, the *data generating model* is given by (6.7) together with the constraints (6.8), (6.9), and (6.10). In the *estimating model* the true response function is approximated by a composition of functions of the form $h_T(.) = h_0(h(.))$, where $h_0(.)$ is a fixed response function. The inner (unknown) function $h(.)$ that has to be estimated is specified by a basis expansion

$$h(\eta_i) = \Phi^T(\eta_i)\boldsymbol{\alpha} = \sum_{k=1}^{m_h} \phi_k(\eta_i)\alpha_k,$$

where $\Phi(\eta_i) = \Phi_i$ is the vector of m_h B-spline basis functions evaluated at η_i and $\boldsymbol{\alpha}$ is the corresponding basis coefficient vector of the inner function. For the predictor we use the basis expansion described in (6.2), (6.3), and (6.4). Thus for the ith observation the estimating model is

$$\mu_i = h_0(\Phi^T(\sum_{j=1}^{p} \boldsymbol{\psi}_{ij}^T \boldsymbol{\beta}_j)\boldsymbol{\alpha}), \tag{6.11}$$

subject to constraints corresponding to (6.8), (6.9), and (6.10). Let $\boldsymbol{\Phi} = (\Phi_1, ..., \Phi_n)^T$ denote the basis expansion evaluated at each observation. Note that the intercept and a multiplicative factor is absorbed into the inner function $h(\eta) = \Phi^T(\eta)\boldsymbol{\alpha}$.

6.2.1 Estimation Procedure

Let the estimating model (6.11) for all observations be given in vector form as

$$\boldsymbol{\mu}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = h_0\left(\boldsymbol{\Phi}(\boldsymbol{\Psi}\boldsymbol{\beta})\boldsymbol{\alpha}\right). \tag{6.12}$$

As in GAMs we assume that the response is from a simple exponential family as described in (6.5). While the log-likelihoods of GAMs given in (6.6) depends only on $\boldsymbol{\beta}$ the log-likelihood function for the more general model is

$$l(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i\theta_i - b(\theta_i))/\phi.$$

It depends on $\boldsymbol{\alpha}$ by $h(\eta_i) = \boldsymbol{\Phi}_i^T\boldsymbol{\alpha}$ and on $\boldsymbol{\beta}$ by $\mu_i = h_0(h(\boldsymbol{x}_i^T\boldsymbol{\beta}))$. Estimates are obtained by minimizing the log-likelihood function $l(\boldsymbol{\alpha}, \boldsymbol{\beta})$, subject to constraints. We present an algorithm which is based on boosting techniques. Each boosting iteration splits into two steps

1. Update of the response function with the predictor fixed.
2. Update of the predictor with the response function fixed.

Both updates are based on penalized and constrained Fisher scoring, respectively. In each boosting iteration only one of these steps is carried out. For the updating of the predictor we use componentwise boosting. Therefore, only one predictor function is updated within one iteration. Early stopping ensures that not all variables are updated and variable selection is obtained. First we give an unrestricted version of the algorithm and then we will add the necessary constraints (6.8), (6.9), and (6.10).

Estimation of Response Function for Fixed Predictor

Let $\widehat{\boldsymbol{\eta}}^{(l-1)} = \boldsymbol{\Psi}\widehat{\boldsymbol{\beta}}^{(l-1)}$ be the fixed estimate of the predictor of the previous step . Then the estimation of the response function corresponds to fitting the model $\boldsymbol{\mu} = h_0(\widehat{\boldsymbol{\Phi}}^{(l-1)}\widehat{\boldsymbol{\alpha}}^{(l-1)} +$

$\widehat{\boldsymbol{\Phi}}^{(l-1)}\widehat{\boldsymbol{a}}^{(l)}$) where $\widehat{\boldsymbol{\Phi}}^{(l-1)}\widehat{\boldsymbol{\alpha}}^{(l-1)}$ is the previously fitted value, which is included as an offset. Note that $\widehat{\boldsymbol{\Phi}}^{(l-1)}$ denotes the evaluation of the basis functions at the current $\widehat{\boldsymbol{\eta}}^{(l-1)}$. One step of penalized Fisher scoring has the form

$$\begin{aligned}\widehat{\boldsymbol{a}}^{(l)} &= \nu_h \cdot \Big((\widehat{\boldsymbol{\Phi}}^{(l-1)})^T(\widehat{\boldsymbol{D}}_h^{(l-1)})^T(\widehat{\boldsymbol{\Sigma}}^{(l-1)})^{-1}\widehat{\boldsymbol{D}}_h^{(l-1)}\widehat{\boldsymbol{\Phi}}^{(l-1)} + \lambda_h \boldsymbol{K}_h\Big)^{-1} \times \\ &\quad \times(\widehat{\boldsymbol{\Phi}}^{(l-1)})^T(\widehat{\boldsymbol{D}}_h^{(l-1)})^T(\widehat{\boldsymbol{\Sigma}}^{(l-1)})^{-1}(\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^{(l-1)})\end{aligned} \tag{6.13}$$

where

$$\widehat{\boldsymbol{D}}_h^{(l-1)} = \text{diag}\left\{\frac{\partial h_0(\widehat{h}^{(l-1)}(\widehat{\eta}_i^{(l-1)}))}{\partial \widehat{h}^{(l-1)}(\eta)}\right\}_{i=1}^n \tag{6.14}$$

is the estimated derivative matrix evaluated at the estimate of the previous step $h_0(\widehat{h}^{(l-1)}(\eta))$, and

$$\widehat{\boldsymbol{\Sigma}}^{(l-1)} = \text{diag}\left\{\sigma^2(h_0(\widehat{h}^{(l-1)}(\widehat{\eta}_i^{(l-1)})))\right\}_{i=1}^n \tag{6.15}$$

is the matrix of variances evaluated at $h_0(\widehat{h}^{(l-1)}(\eta))$. $\boldsymbol{K}_h$ is the penalty matrix which penalizes the second derivative of the estimated (approximated) response function. The matrix $\boldsymbol{K}_h$ is symmetric and each entry has the form

$$\boldsymbol{K}_h = \{k_{ij}\}, \text{ with } k_{ij} = \int \left(\frac{d^2}{d\eta^2}\phi_i(t)\right)\left(\frac{d^2}{d\eta^2}\phi_j(t)\right) dt. \tag{6.16}$$

The main idea of boosting is to approximate the optimum in small steps. If the step size is too large the procedure suffers. Therefore, one uses the concept of weak learning proposed by Shapire (1990) and Bühlmann and Yu (2003). In our procedure the weakness of learners is enforced by large λ_h and small ν_h. The latter is fixed by using $\nu_h = 0.1$. Since λ_h only penalizes the second derivative of the functions the additional shrinkage parameter $\nu_h = 0.1$ is helpful to make the learner weak (see also Tutz and Binder, 2006; Schmid and Hothorn, 2008; Hothorn et al., 2010).

Componentwise Boosting for Fixed Response Function

Let $\widehat{h}^{(l-1)}(.)$ be the fixed estimate of the response function of the previous step. The design matrix of the predictor is $\boldsymbol{\Psi} = (\boldsymbol{\Psi}_1, ..., \boldsymbol{\Psi}_p)$ and $\boldsymbol{\beta}^T = (\boldsymbol{\beta}_1^T, ..., \boldsymbol{\beta}_p^T)$ is the corresponding parameter vector. Componentwise boosting for additive predictors means that within one boosting step only one subvector $\boldsymbol{\beta}_j$ of $\boldsymbol{\beta}$ is updated. So we fit the model $\boldsymbol{\mu} = h_0(\widehat{h}^{(l-1)}(\boldsymbol{\Psi}\widehat{\boldsymbol{\beta}}^{(l-1)} + \boldsymbol{\Psi}_j\boldsymbol{b}_j))$, where $\boldsymbol{\Psi}\widehat{\boldsymbol{\beta}}^{(l-1)}$ is a fixed offset representing the previous update. Therefore only the covariate $\boldsymbol{x}_j$ is included in the model. The penalized Fisher scoring for parameter $\boldsymbol{\beta}_j$ has the form

$$\begin{aligned}\widehat{\boldsymbol{b}}_j^{(l)} &= \nu_f \cdot \Big(\boldsymbol{\Psi}_j^T(\widehat{\boldsymbol{D}}_{\boldsymbol{\eta}}^{(l-1)})^T(\widehat{\boldsymbol{\Sigma}}^{(l-1)})^{-1}\widehat{\boldsymbol{D}}_{\boldsymbol{\eta}}^{(l-1)}\boldsymbol{\Psi}_j + \lambda_f \boldsymbol{K}_j\Big)^{-1} \times \\ &\quad \times\boldsymbol{\Psi}_j^T(\widehat{\boldsymbol{D}}_{\boldsymbol{\eta}}^{(l-1)})^T(\widehat{\boldsymbol{\Sigma}}^{(l-1)})^{-1}(\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^{(l-1)}),\end{aligned} \tag{6.17}$$

where $\nu_f = 0.1$ is a fixed shrinkage parameter and

$$\begin{aligned} \widehat{\boldsymbol{D}}_{\boldsymbol{\eta}}^{(l-1)} &= \operatorname{diag}\left\{ \frac{\partial h_0(\widehat{h}^{(l-1)}(\widehat{\eta}_i^{(l-1)}))}{\partial \eta} \right\}_{i=1}^{n} \\ &= \operatorname{diag}\left\{ \frac{\partial h_0(\widehat{h}^{(l-1)}(\widehat{\eta}_i^{(l-1)}))}{\partial h^{(l-1)}(\eta)} \cdot \frac{\partial \widehat{h}^{(l-1)}(\widehat{\eta}_i^{(l-1)})}{\partial \eta} \right\}_{i=1}^{n} \end{aligned} \tag{6.18}$$

is the matrix of derivatives evaluated at the values of the previous iteration, and

$$\widehat{\boldsymbol{\Sigma}}^{(l-1)} = \operatorname{diag}\left\{ \sigma^2(h_0(\widehat{h}(\widehat{\eta}_i^{(l-1)}))) \right\}_{i=1}^{n} \tag{6.19}$$

is the variance from the previous step and $\boldsymbol{K}_j$ is a penalty matrix which penalizes the second derivatives of the predictor functions. $\boldsymbol{K}$ is a symmetric matrix and is similar to (6.16). $\psi_{jk}(.)$ denotes the kth basis function of the jth predictor function then

$$\boldsymbol{K}_j = \{k_{j|kl}\} = \int \left(\frac{d^2}{dt^2} \psi_{jk}(t) \right) \left(\frac{d^2}{dt^2} \psi_{jl}(t) \right) dt.$$

As for the update of the response function we fix $\nu_f = 0.1$ to make the procedure a weak learner.

Constraints

As already mentioned, for uniqueness three constraints must be fulfilled. First we consider the constraints of the predictor. The predictor is constrained in two ways. The first set of constraints is

$$\int_{\min\{\boldsymbol{x}_j\}}^{\max\{\boldsymbol{x}_j\}} \sum_{k=1}^{m_j} \psi_{jk}(t) \widehat{\beta}_{jk} dt = 0,\ j = 1,\ ...,\ p$$

is fulfilled if in each update step

$$\sum_{k=1}^{m_j} w_{jk} \beta_{jk} = 0,\ j = 1,\ ...,\ p \text{ with } w_{jk} = \int \psi_{jk}(t) dt \tag{6.20}$$

holds for each predictor functions. The restricted quadratic optimization problem that corresponds to the penalized Fisher scoring step (6.17) with the constraints (6.20) is

$$\begin{aligned} \widehat{\boldsymbol{b}}_j = \nu_f \cdot \operatorname{argmin}_{\boldsymbol{b} \in \mathbb{R}^{m_j}} \Big\{ & \boldsymbol{b}_j^T \left(\boldsymbol{\Psi}_j^T (\widehat{\boldsymbol{D}}_{\boldsymbol{\eta}}^{(l-1)})^T (\widehat{\boldsymbol{\Sigma}}^{(l-1)})^{-1} \widehat{\boldsymbol{D}}_{\boldsymbol{\eta}}^{(l-1)} \boldsymbol{\Psi}_j + \lambda_f \boldsymbol{K}_j \right) \boldsymbol{b}_j \\ & -2 \boldsymbol{b}_j^T \boldsymbol{\Psi}_j^T (\widehat{\boldsymbol{D}}_{\boldsymbol{\eta}}^{(l-1)})^T (\widehat{\boldsymbol{\Sigma}}^{(l-1)})^{-1} (\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^{(l-1)}),\ \text{s.t. (6.20)} \Big\}. \end{aligned} \tag{6.21}$$

We use the R-package `quadprog` from Turlach (2009) to solve the linear restricted quadratic optimization problem.

The second constraint of the predictor is

$$\sum_{j=1}^{p} \int_{\min\{\boldsymbol{x}_j\}}^{\max\{\boldsymbol{x}_j\}} f_j(t)^2 dt = c.$$

By basis expansion we get

$$\sum_{j=1}^{p} \int_{\min\{\boldsymbol{x}_j\}}^{\max\{\boldsymbol{x}_j\}} \left[\sum_{k=1}^{m_j} (\beta_{jk} \psi_{jk}(t)) \right]^2 dt = c, \tag{6.22}$$

where the choice of c is arbitrary. After updating the jth subvector of $\boldsymbol{\beta}$ by (6.21) we scale $\boldsymbol{\beta}$ to Euclidean norm 1, $\|\boldsymbol{\beta}\| = 1$, and by (6.22) c is fixed. We use natural cubic B-splines (compare Dierckx, 1993). This basis expansion is provided by the `fda` package in `R` (Ramsay et al., 2010). The B-spline basis for 8 equidistant inner knots on $[-1, 1]$ are shown in Figure 6.1. The maximum of all basis functions $\psi_{jk}(t)$ inner the knots is 1. This

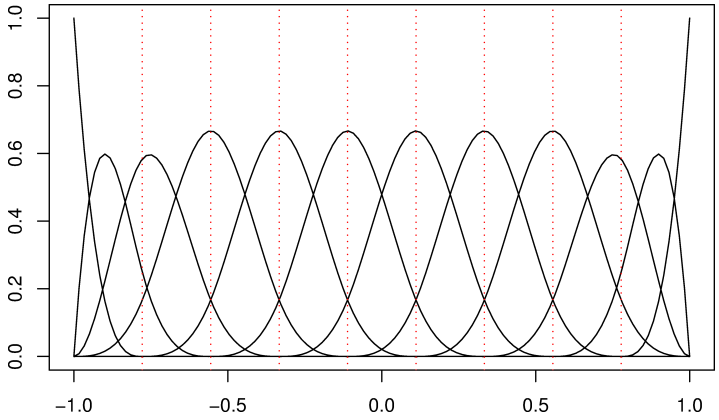

FIGURE 6.1: *The cubic natural B-spline basis for* 8 *equidistant inner knots on the interval* $[-1, 1]$.

maximum is realized for the first and the last basis function of each predictor function $f_j(t)$ if the extreme values of $\boldsymbol{x}_j$ are applied, i.e. $\psi_{jm_j}(\max\{\boldsymbol{x}_j\}) = 1$ or $\psi_{j1}(\min\{\boldsymbol{x}_j\}) = 1$. Further $\|\boldsymbol{\beta}\| = 1$ holds and the predictor $\eta = \sum_{j=1}^{p} \boldsymbol{\psi}_j^T \boldsymbol{\beta}_j$ is restricted to $[-\sqrt{p}, \sqrt{p}]$. This can be shown as follows. Let us consider $\boldsymbol{x}_{\max} = (\max\{\boldsymbol{x}_1\}, ..., \max\{\boldsymbol{x}_p\})^T$. The corresponding vector of basis functions is $\boldsymbol{\psi}_{\max} = (\boldsymbol{\psi}_1^T(\max\{\boldsymbol{x}_1\}), ..., \boldsymbol{\psi}_p^T(\max\{\boldsymbol{x}_p\}))^T$ and p entries of $\boldsymbol{\psi}_{\max}$ are 1 and the remaining are zero. The predictor is maximized if each non zero component of $\boldsymbol{\psi}_{\max}$ is multiplied by $\beta_{jm_j} = \pm 1/\sqrt{p}$. In this case the maximal entries of $\boldsymbol{\psi}_{\max}$ are weighted with maximal amount entries of $\boldsymbol{\beta}$ subject to $\|\boldsymbol{\beta}\| = 1$ which maximizing the $L1$-norm of $\boldsymbol{\beta}$. p components of $\boldsymbol{\psi}_{\max}$ are 1 and the remaining are 0. So the range of $\eta = \sum_{j=1}^{p} \boldsymbol{\psi}_j^T \boldsymbol{\beta}_j$ is in $[-\sqrt{p}, \sqrt{p}]$. This estimation of the range is very conservative. On the one hand the upper bound $\boldsymbol{\psi}_{\max}$ is very rough. $\boldsymbol{x}_{\max}$ conforms that all componentwise maxima are realized in the same observation. On the other hand the

choice of $\beta_{jm_j} = \pm 1/\sqrt{p}$ for all $j = 1, ..., p$ generates (non centered) functions which are zero except for a peak of height $\pm 1/\sqrt{p}$ at the right. In the simulation study (section 6.3) and the data example (section 6.4) we fix the knots of the basis expansion of $h(.)$ equidistantly on this $[-1, 1]$ and the limits are never exhausted.

Constraints for the Response function

We assume that the response function $h_T(.) = h_0(h(.))$ is monotonically non-decreasing. Since the canonical link function is non-decreasing we have to estimate a monotonically non-decreasing inner function $h(.)$.

The inner function is approximated by a basis expansion $h(\eta) \approx \Phi^T(\eta)\boldsymbol{\alpha}$. Φ is the matrix of basis functions evaluated at η and $\boldsymbol{\alpha}$ is the corresponding coefficient vector. $h(\eta) = \Phi^T(\eta)\boldsymbol{\alpha}$ is monotonically non-decreasing if the components of the coefficient vector $\boldsymbol{\alpha}$ are monotonically non-decreasing, i.e. $\alpha_i \leq \alpha_{i+1}$ for all $i = 1, ..., m_h - 1$. A boosting update has the form $\widehat{\boldsymbol{\alpha}}^{(l)} = \widehat{\boldsymbol{\alpha}}^{(l-1)} + \widehat{\boldsymbol{a}}^{(l)}$. So after each update step the system of inequations $\widehat{\alpha}_i^{(l-1)} + \widehat{a}_i^{(l)} \leq \widehat{\alpha}_{i+1}^{(l-1)} + \widehat{a}_{i+1}^{(l)}$, $i = 1, ..., m_h - 1$, must be fulfilled . Each update step is restricted on the following space

$$\mathcal{A} = \left\{ \boldsymbol{a}^{(l)} : a_2^{(l)} - a_1^{(l)} \geq \widehat{\alpha}_1^{(l-1)} - \widehat{\alpha}_2^{(l-1)}, ..., a_k^{(l)} - a_{k-1}^{(l)} \geq \widehat{\alpha}_{k-1}^{(l-1)} - \widehat{\alpha}_k^{(l-1)} \right\}. \tag{6.23}$$

$\mathcal{A}$ can be rewritten as a system of inequations. In the same way as for restricted updates of the predictor functions we use the corresponding quadratic optimization problem of (6.13) with linear constraints $\mathcal{A}$

$$\begin{aligned} \widehat{\boldsymbol{a}} = \nu_h \cdot \text{argmin}_{\boldsymbol{a}} \Big\{ & \boldsymbol{a}^T \left(\boldsymbol{\Phi}^T \widehat{\boldsymbol{D}}^{(l-1)} (\widehat{\boldsymbol{\Sigma}}^{(l-1)})^{-1} \widehat{\boldsymbol{D}}^{(l-1)} \boldsymbol{\Phi} + \lambda_h \boldsymbol{K}_h \right) \boldsymbol{a} \\ & -2\boldsymbol{a}^T \boldsymbol{\Phi}^T \widehat{\boldsymbol{D}}^{(l-1)} (\widehat{\boldsymbol{\Sigma}}^{(l-1)})^{-1} (\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^{(l-1)}), \text{ s.t. } \boldsymbol{a} \in \mathcal{A} \Big\}. \end{aligned} \tag{6.24}$$

The R-package `quadprog` (see Turlach, 2009) is able to solve such optimization problems.

6.2.2 Algorithm

The basic algorithm is given below and shows the interplay of the two steps. In each iteration step the updates of each predictor function and the updates of the response functions are computed. In the last step of each iteration it is evaluated which update is to be preferred. Only the maximizer of the log-likelihood function is used used in the final update. Thus in each iteration step either one predictor function or the response function is updated.

Algorithm: FLAP

Step 1 (Initialization)

Set $\hat{\boldsymbol{\beta}}^{(0)} = \mathbf{0}$ and $\hat{\boldsymbol{\eta}}^{(0)} = \mathbf{0}$. Determine $\boldsymbol{\alpha}^{(0)}$ so that $\boldsymbol{\phi}(t)^T\boldsymbol{\alpha}^{(0)} = g(\bar{\boldsymbol{y}}) + 0.01t$ is a line with small gradient and intercept $g(\bar{\boldsymbol{y}})$, where $\bar{\boldsymbol{y}} = \sum_{i=1}^{n} y_i$. Compute $\widehat{\boldsymbol{D}}^{(0)}$, $\widehat{\boldsymbol{D}}_{\boldsymbol{\eta}}^{(0)}$ and $\widehat{\boldsymbol{\Sigma}}^{(0)}$.

Step 2 (Iteration)

For $l = 1, 2, \ldots, M$

1. *Predictor update*
   - Compute for each $j \in \{1, \ldots, p\}$ the update $\widehat{\boldsymbol{b}}_j^{(l)}$ as described in (6.21) and set $\boldsymbol{b}_j^{(l)} = (\mathbf{0}^T, \ldots, (\widehat{\boldsymbol{b}}_j^{(l)})^T, \ldots, \mathbf{0}^T)^T$ and determine the update candidate
   $$\boldsymbol{\beta}_j^{(l)} = \widehat{\boldsymbol{\beta}}^{(l-1)} + \boldsymbol{b}_j^{(l)}.$$
   - Compute $\widehat{\boldsymbol{\beta}}_j^{(l)} = \boldsymbol{\beta}_j^{(l)}/||\boldsymbol{\beta}_j^{(l)}||$ and the corresponding log-likelihood function $l(\boldsymbol{\alpha}^{(l-1)}, \widehat{\boldsymbol{\beta}}_j^{(l)})$.
   - Choose the parameter vector $\widehat{\boldsymbol{\beta}}_{opt}^{(l)} = \text{argmax}_{\widehat{\boldsymbol{\beta}}_j^{(l)}, j=1,\ldots,p} l(\boldsymbol{\alpha}^{(l-1)}, \widehat{\boldsymbol{\beta}}_j^{(l)})$ which minimizes the log-likelihood function and set $\widehat{\boldsymbol{\beta}}^{(l)} = \widehat{\boldsymbol{\beta}}_{opt}^{(l)}$
2. *Response function update*
   - Compute $\widehat{\boldsymbol{a}}^{(l)}$ as described in (6.24) and set $\widehat{\boldsymbol{\alpha}}^{(l)} = \widehat{\boldsymbol{\alpha}}^{(l-1)} + \widehat{\boldsymbol{a}}^{(l)}$
   - Compute $\hat{h}^{(l)}(\boldsymbol{\eta}^{(l-1)}) = \boldsymbol{\Phi}^{(l-1)}\widehat{\boldsymbol{\alpha}}^{(l)}$ and the corresponding log-likelihood function $l(\widehat{\boldsymbol{\alpha}}^{(l)}, \widehat{\boldsymbol{\beta}}^{(l-1)})$.
3. *Update choice*
   - If $l(\widehat{\boldsymbol{\alpha}}^{(l)}, \widehat{\boldsymbol{\beta}}^{(l-1)}) > l(\widehat{\boldsymbol{\alpha}}^{(l-1)}, \widehat{\boldsymbol{\beta}}^{(l)})$ then $\boldsymbol{\alpha}^{(l)}$ is updated and $\widehat{\boldsymbol{\beta}}$ remains unchanged, $\widehat{\boldsymbol{\beta}}^{(l)} = \widehat{\boldsymbol{\beta}}^{(l-1)}$.
   - If $l(\widehat{\boldsymbol{\alpha}}^{(l)}, \widehat{\boldsymbol{\beta}}^{(l-1)}) \leq l(\widehat{\boldsymbol{\alpha}}^{(l-1)}, \widehat{\boldsymbol{\beta}}^{(l)})$ then $\widehat{\boldsymbol{\beta}}^{(l)}$ is updated and $\widehat{\boldsymbol{\alpha}}$ remains unchanged, $\hat{\boldsymbol{\alpha}}^{(l)} = \boldsymbol{\alpha}^{(l-1)}$.

Note that we transform the domain of each function by $\widetilde{\boldsymbol{x}}_j = \boldsymbol{x}_j/(\max\{\boldsymbol{x}_j\} - \min\{\boldsymbol{x}_j\})$, and the range of each domain is normed to 1. By this transformation the update of each function becomes more similar.

Choice of Tuning Parameter

The FLAP procedure uses three tuning parameter: λ_f for the smoothing of the predictor function, λ_h for the smoothing of the response function and m_{stop} for the number of boosting iterations. We use 5-fold cross-validation for determining these parameters. In each case we use a grid of three tuning parameter values with $\lambda_h \in \{2, 1, 0.5\}$ and $\lambda_f \in \{2, 1, 0.5\}$ as candidates. The maximal number of boosting iteration was set to $M = 5000$. All in all, we have to cross-validate the model for nine tuning parameter constellations over 5000 boosting iterations.

Cut Version

An unsatisfying property of the presented boosting procedure is that some predictors are updated only once or twice. To enforce variable selection we also present a cut version in which predictors with small estimated functions are excluded based on a threshold. If in the (l)th iteration the Euclidean length of the coefficient vector of jth predictor function is smaller than $1/p$, $\|\boldsymbol{\beta}_j^{(l)}\| < 1/p$, we set this jth subvector to $\mathbf{0}$, $\boldsymbol{\beta}_j^{(l)} = \mathbf{0}$. The new cut parameter vector is restandardized to Euclidean norm 1. The optimal tuning parameter for the cut version $\widetilde{\lambda}_h$, $\widetilde{\lambda}_f$ and $\widetilde{m}_{stop}$ are also determined by cross-validation. In the simulation study the threshold $1/p$ works quite well. Of course the threshold limits could be optimized.

6.3 Simulation Study

To evaluate the performance of the FLAP procedure we compare it with three established procedures:

GAMBoost, which is a likelihood based boosting procedure which performs variable selection by early stopping (Tutz and Binder, 2006).

mboost, which is a boosting procedure proposed by Hothorn et al. (2010) that also enforces variable selection by early stopping. The corresponding R-package is `mboost` (Hothorn et al., 2009).

mgcv, which fits a GAM with variable selection based on penalization. For details see Wood (2006) and Wood (2011).

We use two model assessment measurements for the comparison of models. After determining the optimal model by 5-fold crossvalidation we predict $\widehat{\boldsymbol{\mu}}_{test} = h_0(\Phi(\boldsymbol{\Psi}_{test}\widehat{\boldsymbol{\beta}})\widehat{\boldsymbol{\alpha}})$ based on an independently chosen data set $(\boldsymbol{y}_{test}, \boldsymbol{X}_{test})$ and evaluate the predictive deviance

$$\text{Dev(test)} = -2(l(\boldsymbol{y}_{test}, \widehat{\boldsymbol{\mu}}_{test}) - l(\boldsymbol{y}_{test}, \boldsymbol{y}_{test})).$$

The other measure for accuracy of the estimated predictor functions is created by is the

$$\text{MSE}_f = \sum_{j=1}^{p} \int (\tilde{f}_j(t) - \hat{\tilde{f}}_j(t))^2 dt,$$

which compares two scaled versions of the functions.

$$\tilde{f}_j(t) = f_j(t) \cdot F$$

is the true function $f_j(t)$ scaled by

$$F = \left(\sum_{j=1}^{p} \int |f_j(t)| dt \right)^{-1}$$

and

$$\hat{\tilde{f}}_j(t) = \hat{f}_j(t) \cdot \hat{F}$$

is the corresponding estimate where

$$\hat{F} = \left(\sum_{j=1}^{p} \int |\hat{f}_j(t)| dt \right)^{-1}.$$

We have to normalize each function by the sum of integrals of the absolutes of all functions to make the results comparable. With MSE_f we measure the goodness of fit of the predictor in two ways. First the form of each normalized estimate is compared with the original normalized function. Second MSE_f compares the proportion of the altitudes of normalized estimated functions with altitude of the original normalized functions which is equal to the comparison with original function. Note that MSE_f measures only the similarity of forms and proportions of altitudes between the estimates and the true functions. It measures not the deviation between the estimates and the true functions.

We investigate three cases of distribution: normal, Poisson, and binomial with non-canonical response function. For the *normal case* we use a sigmoid response function

$$h_{\text{Norm}}(\eta) = \frac{20}{1 + \exp(-5 \cdot \eta)}$$

and so the response is generated by $y_i = N(h_{\text{Norm}}(\eta_i), 1)$. In the *Poisson case* we use a sigmoid response function similar to the normal case

$$h_{\text{Pois}}(\eta) = \frac{10}{1 + \exp(-5 \cdot \eta)}$$

but the response is generated by $y_i = Pois(h_{\text{Pois}}(\eta_i))$. For the *binomial case* we choose an increasing smooth step function

$$h_{\text{Bin}}(\eta) = \frac{0.25}{1 + \exp(-10 \cdot \eta - 15)} + \frac{0.75}{1 + \exp(-10 \cdot \eta + 15)}$$

with three levels 0, 0.25, and 1. The changeovers between the three levels are quite smooth. The response is generated by $y_i = Bin(h_{\text{Bin}}(\eta_i))$. In each setting the predictor has the same form. The predictor η is generated by p covariate characteristic functions, $\eta = \sum_{j=1}^{p} f_j(x_j)$.

Beyond the distribution and the response function each setting is given by the number of covariates, $p = 5, 10, 25$. Only the first five covariates have influence on the response. The predictor function are

$$\begin{aligned} f_1(x_1) &= \sin(4 \cdot x_1), \\ f_2(x_2) &= \cos(4 \cdot x_2), \\ f_3(x_3) &= 0.5 \cdot x_3^2, \\ f_4(x_4) &= -0.5 \cdot x_4^2, \\ f_5(x_5) &= x_5^3/9, \\ f_j(x_j) &= 0, \; j = 6, \ldots, p. \end{aligned}$$

The covariates are drawn from a truncated normal distribution $N_{\text{trunc}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}^2)$ to avoid problems with outliers. We use the R-package `tmvtnorm` (see Genz et al., 2011) with the range for each covariate being restricted to $[-\pi, \pi]$. The mean of the generating distribution is fixed to $\boldsymbol{\mu} = \mathbf{0}_p$ and the covariance matrix is $\boldsymbol{\Sigma}^2 = \{\sigma_{jk}^2\}_{j,\, k=1,\, \ldots,\, p}$ where $\sigma_{jk}^2 = 1$ for $j = k$ and $\sigma_{jk}^2 = 0.5$ otherwise. For the normal and the Poisson case the number of observations of the training dataset is $n_{train} = 250$ and the test datasets have $n_{test} = 1000$ observations. In the binomial case the 0-1 information of the response is quite weak. Thus in contrast to the both other cases we increase the number of observations to $n_{train} = 1000$ and $n_{test} = 4000$. In all cases the maximal number of boosting iterations is set to $M = 5000$ which is never exhausted over all settings. Each predictor function $f_1(.)$, $f_2(.)$, ..., $f_5(.)$ is expanded by cubic B-splines basis with 20 (inner) knots. The response function is expanded in the same way with 50 (inner) knots (see Figure 6.1).

The results are summarized in Table 6.1. It is seen that the FLAP and the FLAP (cut) outperform the procedures with fixed canonical response function by prediction in nearly all cases. Only the binomial case with 25 predictors the `mboost` outperforms both FLAP procedures. Especially in the Poisson case the FLAP and the FLAP (cut) work very good. In addition to the the predictive deviance the both FLAP procedures outperform the competitors by MSE_f. In the normal case the `mboost` beats the other procedures in terms of MSE_f although the response function is misspecified. But the FLAP procedures are the next best in accuracy of predictor estimation measured by MSE_f. In general the `mboost` performs quite well for MSE_f. This is founded by the special update step of the `mboost` which is controlled by the degrees of freedom (cf. Hothorn et al., 2010; Hofner et al., 2009, 2011). This strategy seems to be very successful even if the response function is misspecified. The MSE_f of the `mboost` is quite close to the MSE_f of the FLAP procedures. In the binomial case the FLAP procedures are strong competitors too.

The cut version of the FLAP does not outperform the other methods in terms of MSE_f or predictive deviance (Dev(test)) but the false positive rates of FLAP (cut) are the best across all settings. Generally `mgcv` includes all covariates in each setting although the option variable selection was chosen. Further in some cases the procedure does not converge. In the binomial case we leave out the `GAMBoost`. The high number of observations increases the computational costs immensely.

For illustration we show the boxplots of MSE_f and Dev(test) for the Poisson case in Figure 6.2.

		FLAP	FLAP (cut)	`mgcv`*	`GAMBoost`**	`mboost`
Normal distribution						
$p = 5$	MSE_f	0.0132	0.0132	0.0153	0.0238	0.0106
	Dev(test)	10884.34	10954.11	26810.77	25874.14	42457.43
	hits	1.000	0.988	1.000	1.000	1.000
	false pos.	—	—	—	—	—
$p = 10$	MSE_f	0.0172	0.0166	0.0174	0.0235	0.0130
	Dev(test)	15589.11	14254.97	28275.76	27325.15	40147.39
	hits	1.000	1.000	1.000	1.000	1.000
	false pos.	1.000	0.608	1.000	0.972	0.952
$p = 25$	MSE_f	0.0209	0.0208	0.0299	0.0235	0.0188
	Dev(test)	25375.87	25112.45	38757.15	27150.12	45861.43
	hits	1.000	1.000	1.000	1.000	1.000
	false pos.	0.906	0.539	1.000	0.753	0.834
Poisson distribution						
$p = 5$	MSE_f	0.0103	0.0103	0.0295	0.0408	0.0176
	Dev(test)	1610.23	1610.23	3921.14	4593.06	4226.54
	hits	1.000	1.000	1.000	1.000	1.000
	false pos.	—	—	—	—	—
$p = 10$	MSE_f	0.0143	0.0138	0.0322	0.0530	0.0205
	Dev(test)	2017.60	2033.99	5432.79	8127.55	4570.11
	hits	1.000	1.000	1.000	1.000	1.000
	false pos.	0.996	0.304	1.000	0.912	0.884
$p = 25$	MSE_f	0.0253	0.0258	0.0489	0.0482	0.0262
	Dev(test)	2877.95	2872.24	1025052	5382.05	4637.24
	hits	1.000	1.000	1.000	1.000	1.000
	false pos.	0.789	0.413	1.000	0.803	0.643
Binomial distribution						
$p = 5$	MSE_f	0.0132	0.0139	0.0183	—	0.0135
	Dev(test)	4226.13	4264.90	4280.19	—	4235.23
	hits	1.000	0.952	1.000	—	1.000
	false pos.	—	—	—	—	—
$p = 10$	MSE_f	0.0182	0.0184	0.0232	—	0.0171
	Dev(test)	4335.16	4324.11	4356.42	—	4325.13
	hits	1.000	0.996	1.000	—	1.000
	false pos.	0.912	0.232	1.000	—	0.984
$p = 25$	MSE_f	0.0216	0.0221	0.0295	—	0.0228
	Dev(test)	4455.67	4461.87	4627.87	—	4439.88
	hits	0.992	0.988	1.000	—	1.000
	false pos.	0.529	0.237	1.000	—	0.745

TABLE 6.1: *Medians of the* Dev(*test*) *and* MSE_f *for each setting of the simulation study and the hits false positive rates across the replications.* * *No convergence in each replication.* ** *No results in the binomial case because of high computational costs.*

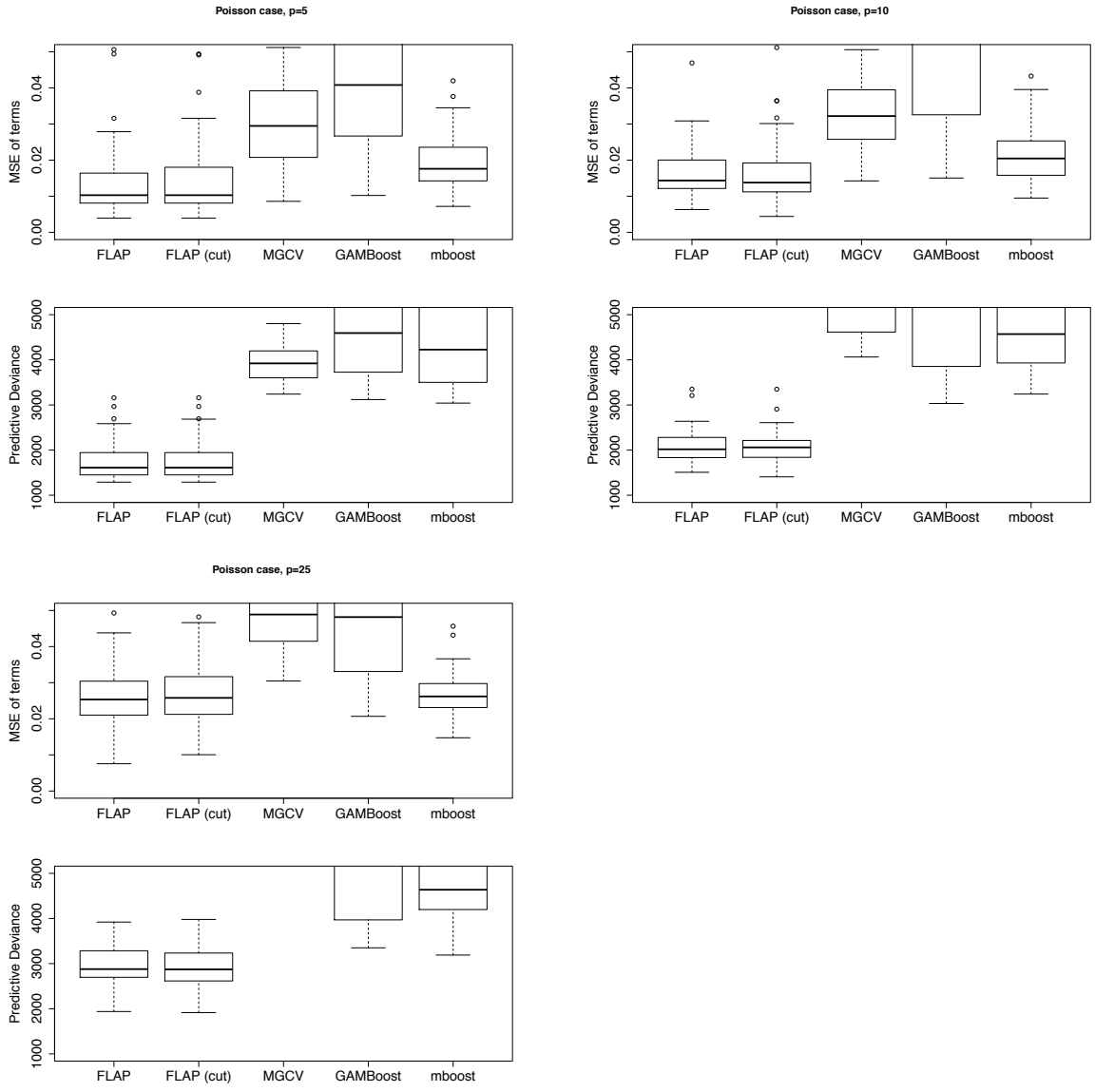

TABLE 6.2: *Boxplots of the predictive deviance (*Dev(*test*)*) and the accuracy of the predictor estimate (*MSE_f*) for the three Poisson setting.*

6.4 Data Example

The method is illustrated by modeling the death rate in the metropolitan area of Sao Paulo. The data were recorded from January 1994 to December 1997 $n = 1351$ days and are available at `http://www.ime.usp.br/~jmsinger/Polatm9497.zip`. We use a sub data set which was also used by Leitenstorfer and Tutz (2007) for the modeling of monotone functions. The response is the number of daily deaths caused by respiratory reasons of people which are 65 years or older `RES65`. The covariates are given in Table 6.3.

Label	Explanation
`TEMPO`	Time in days
`SO2ME.2`	The 24-hours mean of SO_2 concentration (in μ/m^3) over all monitoring measurement stations.
`TMIN.2`	The daily minimum temperature.
`UMID`	The daily relative humidity.
`DIASEM`	Day of the week. (1 =Tuesday, 2 =Wednesday, ..., 7 =Monday)
`CAR65`	Cardialogical caused deaths per day.
`OTH65`	Other (non respiratory or cardiological) caused deaths per day.

TABLE 6.3: *Table of covariates and their labels of the Sao Paulo air pollution data set.*

For `SO2ME.2` and `TMIN.2` we consider the measurements taken 2 days before as influential. This lag was proposed by Conceicao et al. (2001). All predictors are modeled nonparametrically. For the FLAP we used 20 knots for all covariates and for the `mboost` we used the default values. For the `mgcv`, the default values were used, but for the covariate `DIASEM` we had to reduce the number of knots to 7. We determined the optimal tuning parameter by a 5-fold cross-validation, where $\lambda_h, \lambda_f \in \{100, 10, 1, 0.1, 0.01\}$. For both versions of the FLAP we got $\lambda_h = 1$ and $\lambda_f = 0.01$. For all boosting procedures the maximal number of boosting iterations was fixed to 1000. In the Figures 6.2, 6.3, 6.4, and 6.5 we show the estimated functions and the estimated expectation values for the different methods. We do not show the `GAMBoost` because the procedure does not work well on this dataset.

For `TEMPO` the periodic character is identified by all procedures. The SO_2 concentration (`SO2ME.2`) has an clearly increasing trend. If we neglect the high valued outliers this covariate seems to have only a very weak influence. Increasing temperature `TMIN.2` has a decreasing influence on the response `RES65`. This characteristic was detected by all procedures. With the FLAP this trend seems to be stronger. Beyond the outliers, the covariates `UMID` and `DIASEM` have only a small influence on the response. The cut version of FLAP (FLAP (cut)) does not include these covariates. Increasing number of non respiratory caused deaths (`CAR65` and `OTH65`) tends to increase the number respiratory caused deaths.

In contrast to the established methods with canonical link the models with estimated link function have only two main influential covariates, `TEMPO` and `TMIN.2`. The models

with canonical link functions are more complex. In them also the covariates SO2ME.2, CAR65, and OTH65 seem to be influential.

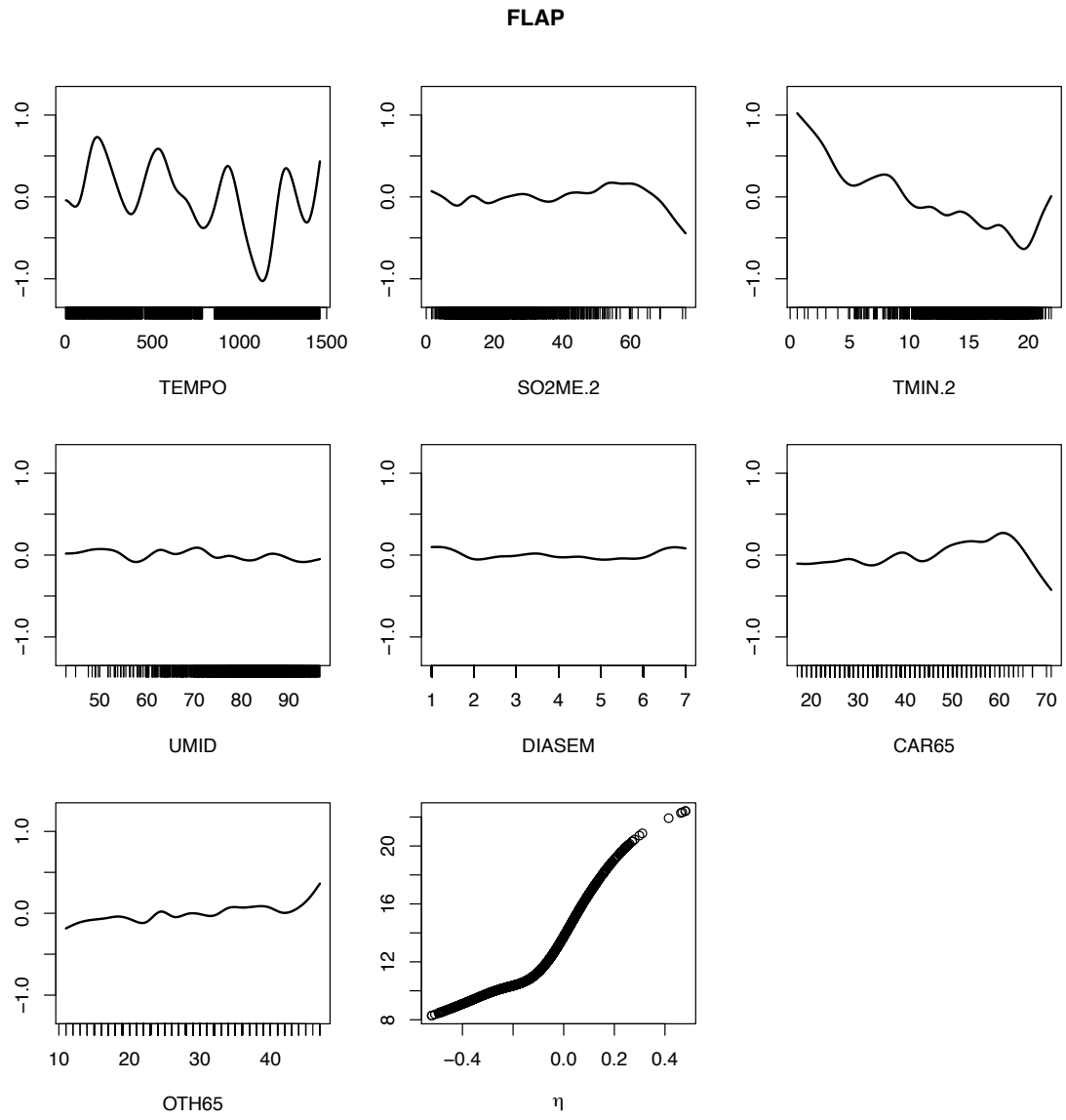

FIGURE 6.2: *The estimated functions and the prediction of Sao Paulo data set estimated by FLAP*

Figures 6.2 and 6.3 also show the estimated response functions of both FLAP procedures, which are different from the canonical response functions shown in Figures 6.4 6.5.

In addition we evaluated the prediction across 50 random splits. The training data set contains 1000 observations and the remaining observations are used as test data. For reducing the computational costs we determined the tuning parameter λ_F and λ_h on the

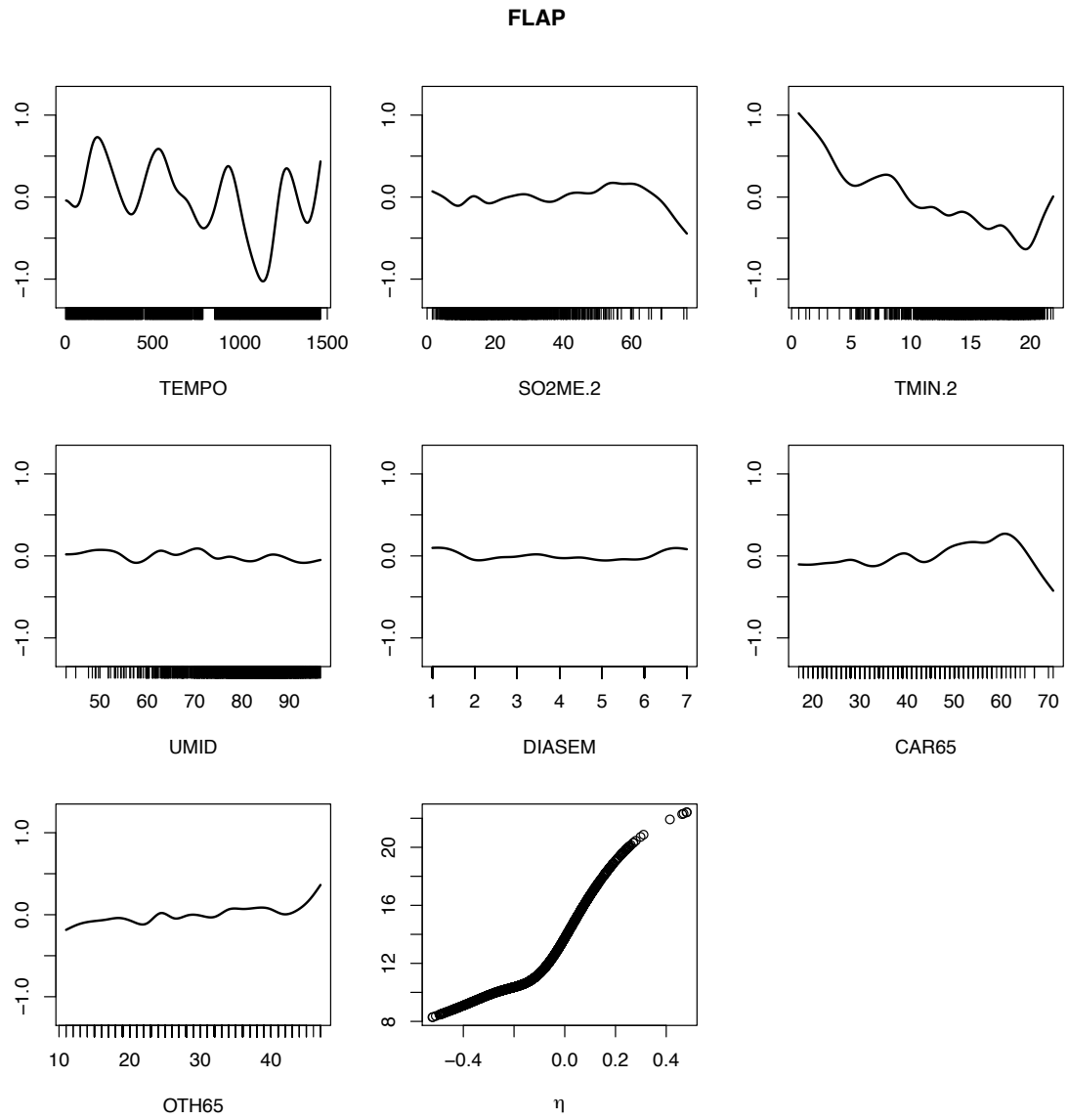

FIGURE 6.3: *The estimated functions and the prediction of Sao Paulo data set estimated by the cut version of the FLAP algorithm.*

complete data set ($n = 1351$) by 5fold cross-validation, and fixed the resulting $\lambda_h = 1$ and $\lambda_f = 0.01$ for the following investigation of prediction. Since we only had to determine the number of optimal boosting iterations by a 5fold cross-validation on the training data set the computational costs were strongly reduced. We used the training data for fitting the model for given tuning parameters and measured the prediction on the test data. We give the medians of the predictive deviances across the random splits and the deviance for complete data set in Table 6.4. The boxplots of the predictive deviances on the test data are given in Figures 6.6. The predictive deviance across the random splits underlines the

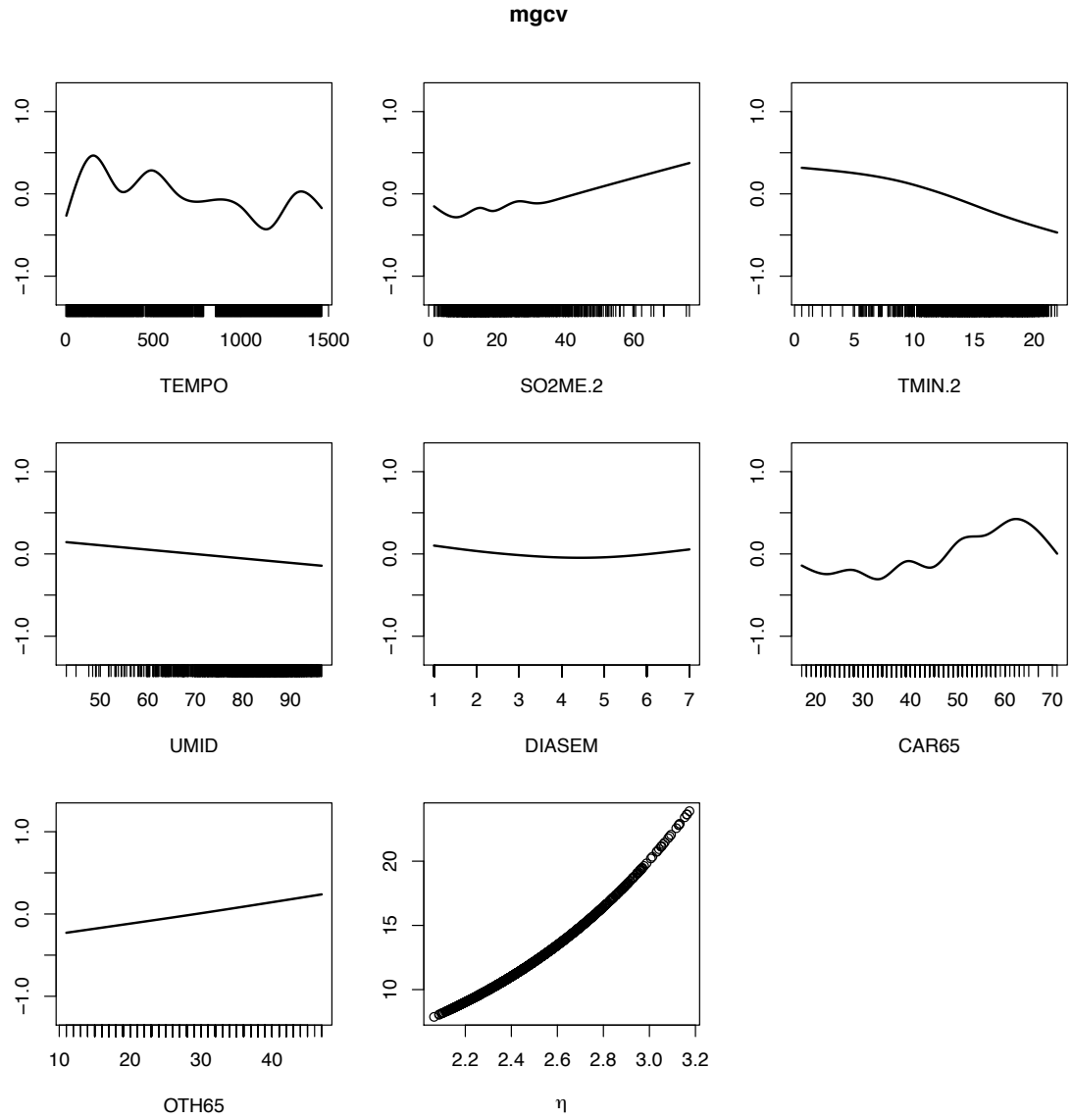

FIGURE 6.4: *The estimated functions and the prediction of Sao Paulo data set estimated by* `mgcv`.

	FLAP	FLAP (cut)	`mgcv`	`mboost`
complete data set	1383.06	1412.30	1547.58	1407.42
random splits	411.26	414.13	437.63	430.52

TABLE 6.4: *Prediction measurements of the Sao Paulo data set for the different procedures. First row: The deviance on the complete data set. Second row: Median across 50 random splits.*

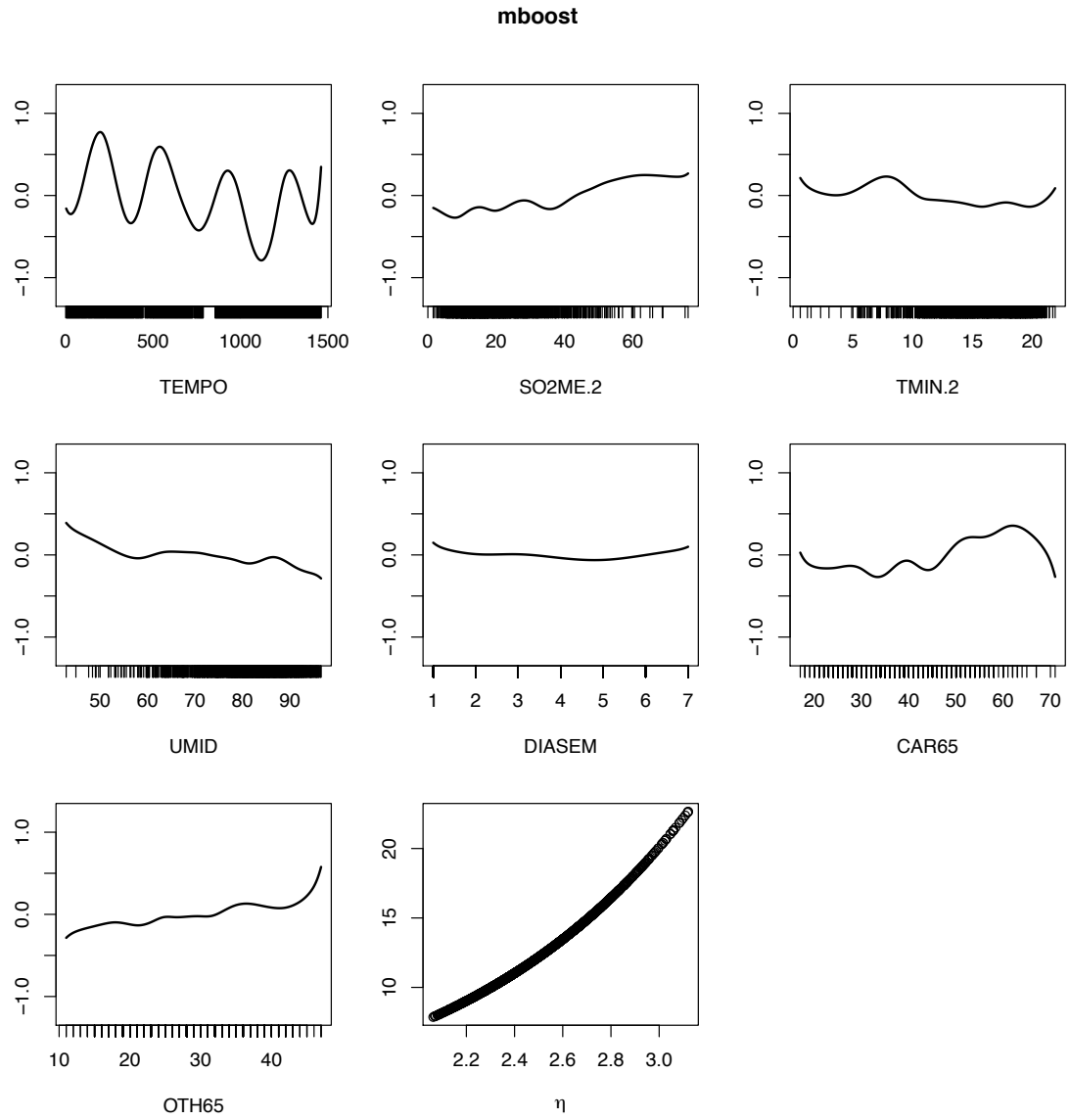

FIGURE 6.5: *The estimated functions and the prediction of Sao Paulo data set estimated by* `mboost`.

results of the simulations study, prediction tends to be better when allowing for flexible link functions.

6.5 Conclusion and Perspectives

A competitive method for estimating GAMs that model the response function non parametrically is presented. The method is based on componentwise boosting, by early stopping

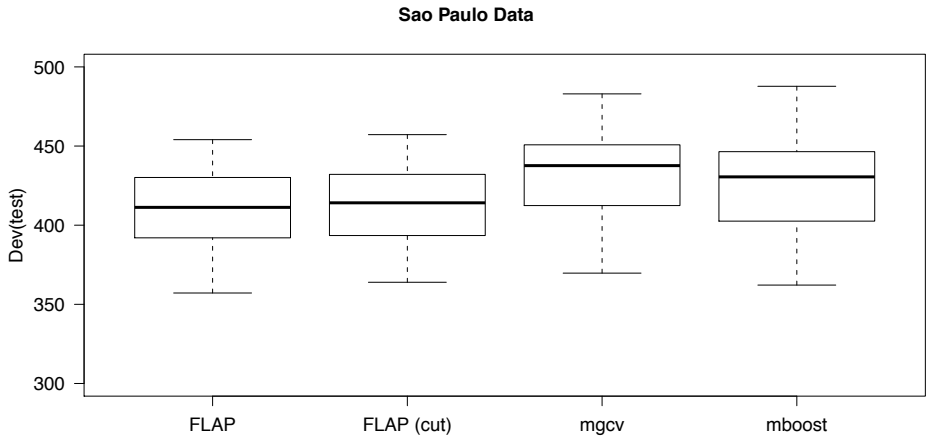

FIGURE 6.6: *Boxplots of the predictive deviances across the 50 random samples of the Sao Paulo data set.*

variable selection is obtained. Especially the predictive performance improves in nearly all simulation settings. Further the accuracy of the predictor estimate change for the better in the Poisson and the binomial case. The `mboost` outperforms both versions of FLAP in the normal case. The performance of the FLAP procedures depends strongly on the choice of tuning parameters in this point. So other and finer grids of tuning parameters could improve the results in the normal case. In all cases the variable selection of the FLAP and FLAP (cut) works quite well which is reported by the hits and false positive rates. Especially the variable selection of the cut version works very well. As in the linear case the estimation of the response function for binary response is a challenge because of the weak information given by 0/1. So we increase the number of observation for this settings.

By small modifications the FLAP procedures can be generalized to semiparametrical models where smooth, linear, and categorial influences on the response are modeled. Generally the `mboost` performs very good concerning the MSE_f. This is an argument for the degree of freedom based update criterion proposed by Hofner et al. (2009, 2011). By a modification in the predictor update of the FLAP procedures it is possible to use this kind of update. The degree of freedom update can also be applied to linear, categorical and nominal covariates. Based on this update a very generic algorithm for the estimation of a wide class of regression models with unknown link function can be designed.

Chapter 7

L1-Penalized Single Index Models

In this chapter we present an algorithm which combines penalization of the parameter vector with estimation of the response function. Additionally we are interested in variable selection and so we penalize the parameter vector by a LASSO term $\sum_{j=1}^{p}|\beta_j|$. Furthermore the roughness of the response function is penalized to avoid overfitting. In contrast to chapter 5 no boosting technique is used for the estimation.

7.1 Introduction

For the linear normal regression problem Hoerl and Kennard (1970) have shown that the Euclidean norm of the maximum likelihood estimate tends to be longer than the Euclidean norm of the true parameter vector. They propose the ridge estimator which penalizes the Euclidean norm of the estimate. The ridge penalty term is given by $\lambda\sum_{j=1}^{p}\beta_j^2$, where β_j is the jth component of the parameter vector $\boldsymbol{\beta}$. Nelder and Wedderburn (1972) introduced the generalized linear model (GLM) which describes a monotonically increasing influence of a linear combination of covariates on a non-normal distributed response variable. For the corresponding estimates the result from Hoerl and Kennard (1970) holds, too. So it makes sense to shrink the parameter estimates to improve the predictive performance of a model. Another very effective strategy to enhance the predictive performance of a model is to select only the group of relevant variables. This effect is discussed by Breiman (1996). A very popular estimator which combines variable selection and coefficient shrinkage is the $L1$-penalization also known as LASSO proposed by Tibshirani (1996). The LASSO penalty term is given by $\lambda\sum_{j=1}^{p}|\beta_j|$. In the last years many algorithms for the estimation of $L1$-penalized GLMs have been proposed, for example Park and Hastie (2007b), Friedman et al. (2010b), and Goeman (2010a). These algorithms are available in `R`-packages, namely `glmpath` (see Park and Hastie, 2007a), `glmnet` (see Friedman et al., 2008), and `penalized` (see Goeman, 2010b).

To be more concrete we consider a GLM for given data $(y_i, \boldsymbol{x}_i)$, $i = 1, \ldots, n$. The conditional expectation of $y_i|\boldsymbol{x}_i$, $\mu_i = E(y_i|\boldsymbol{x}_i)$, is modeled by

$$g(\mu_i) = \eta_i \quad \text{or} \quad \mu_i = h(\eta_i),$$

where $\eta_i = \beta_0 + \boldsymbol{x}_i^T\boldsymbol{\beta}$ is the linear predictor, $g(.)$ is the link function and $h(.) = g^{-1}(.)$ is the response function. Given $\boldsymbol{x}_i = (x_{i1}, ..., x_{ip})^T$ the y_i are (conditionally) independent observations from a simple exponential family

$$f(y_i|\theta_i, \phi) = \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{\phi} + c(y_i, \phi)\right\}, \tag{7.1}$$

where θ_i is the natural parameter of the family, ϕ is a scale or dispersion parameter and $b(.), c(.)$ are specific functions corresponding to the type of the family. In general we consider centered and standardized covariates, i.e. $\sum_{i=1}^n x_{ji} = 0$ and $n^{-1}\sum_{i=1}^n x_{ji}^2 = 1$, $j = 1, ..., p$.

For GLMs the response function $h(.)$ is fixed and known. Often the canonical link $h_0(.)$ is chosen regardless to correctness. But misspecified link functions can lead to substantial bias in the regression parameters (see Czado, 1992; Czado and Santner, 1992; Czado and Munk, 2000). That may be avoided by flexible modeling of the link. In the following we present a procedure which combines the estimation of the link function and the $L1$-penalization of the parameter vector.

7.2 Data Generating and Approximating Model

Let the *data generating model* be given by

$$E(y_i|\boldsymbol{x}_i) = \mu_i = h_T(\eta_i),$$

where $h_T(.)$ is the unknown true transformation function and $\eta_i = \boldsymbol{x}_i^T\boldsymbol{\beta}$ is the linear predictor. As in GLMs it is assumed that for given $\boldsymbol{x}_i$ the response y_i comes from a simple exponential family). In contrast to GLMs let the linear predictor η_i contain no intercept because the intercept is absorbed into $h_T(.)$. Also the scaling of the parameter vector $c \cdot \boldsymbol{\beta}$ can be compensated by the response function, i.e. $h(\boldsymbol{x}_i\boldsymbol{\beta}) = \widetilde{h}(c \cdot \boldsymbol{x}_i\boldsymbol{\beta})$, $c > 0$. Therefore, usually the Euclidean length of $\boldsymbol{\beta}$ is fixed to 1, that is, $||\boldsymbol{\beta}||_2 = 1$ (for example, Härdle et al., 1993). If $\boldsymbol{\beta}$ is restricted to the Euclidean norm 1 and the response function $h_T(.)$ is monotonically increasing then the estimate $\widehat{\boldsymbol{\beta}}$ is unique in our case. We will discuss later that this restriction fails to guarantee uniqueness in our case.

We want to fit the *approximating model*

$$\mu_i = h_0(h(\eta_i)),$$

where $h_0(.)$ is a fixed transformation which has to be chosen appropriately. The inner function $h(.)$ is considered as unknown and has to be estimated. Typically, for $h_0(.)$ the canonical link function is chosen in particular, to ascertain that. So it is ensured that μ_i is in an admissible range. For example, in the binary case, the logistic distribution function automatically maps the inner function $h(.)$ into [0, 1]. We approximate the function $h(.)$ by expansion in basis functions

$$h(\eta_i) = \sum_{s=1}^k \alpha_s\phi_s(\eta_i) = \Phi_i^T\boldsymbol{\alpha},$$

where $\phi_1(\eta_i), \ldots, \phi_k(\eta_i)$ denote the basis functions evaluated at η_i. With $\boldsymbol{\Phi} = (\Phi_1, ..., \Phi_n)^T$ we denote the matrix containing the basis expansions evaluated at each observation. As basis functions we use natural cubic B-splines (compare Dierckx, 1993) which are provided by the `fda` package in `R` (Ramsay et al., 2010). The B-spline basis for 8 equidistant inner knots on $[-1, 1]$ is shown in Figure 7.1

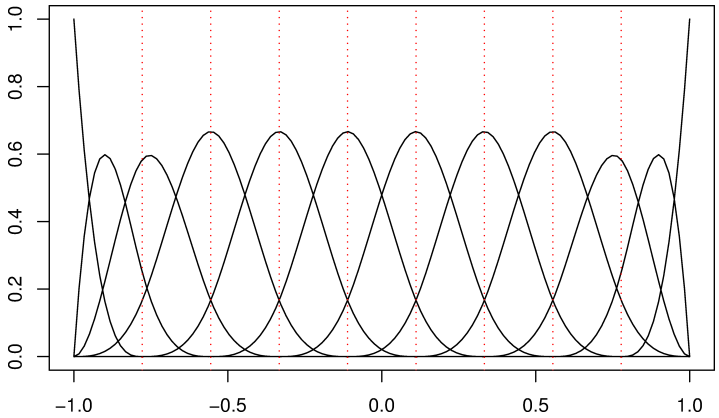

FIGURE 7.1: *The cubic natural B-spline basis for* 8 *equidistant inner knots on the interval* $[-1, 1]$.

7.3 Likelihood and Identification Problem

Let $l(\boldsymbol{\alpha}, \boldsymbol{\beta})$ denote the log-likelihood function

$$l_{pen}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i \theta_i - b(\theta_i))/\phi),$$

which depends on $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ by

$$\mu_i = h_0(\sum_{s=1}^{k} \alpha_s \phi_s(\boldsymbol{x}_i^T \boldsymbol{\beta})), \tag{7.2}$$

where $\theta_i = \theta(\mu_i)$ for a known function $\theta(.)$. The penalized likelihood problem we consider is based on the penalized log-likelihood

$$l_p(\boldsymbol{\alpha}, \boldsymbol{\beta}) = l(\boldsymbol{\alpha}, \boldsymbol{\beta}) - \lambda_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_1 - \lambda_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha}, \tag{7.3}$$

where $l(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is the log-likelihood, $\boldsymbol{K} = (k_{ij})_{i,j \in \{1,..,k\}}$ is a symmetric matrix which penalizes the roughness of the response function by the integral of the squared second derivative of the inner function

$$k_{ij} = \int_{-\infty}^{\infty} \left(\frac{d^2}{d\eta^2} \phi_i(\eta) \right) \left(\frac{d^2}{d\eta^2} \phi_j(\eta) \right) d\eta, \tag{7.4}$$

and $\|\boldsymbol{\beta}\|_1$ is the well known $L1$-penalization of the parameter vector of the linear predictor.

As already mentioned the scaling of $\boldsymbol{\beta}$ can be compensated by the response function. But shrinkage procedures are based on scaling the coefficients by penalization. So fixing the Euclidean length of $\boldsymbol{\beta}$ to 1 as proposed by Härdle et al. (1993) or Weisberg and Welsh (1994) fails. If the penalization of $\boldsymbol{\beta}$ is strong, i.e. $\lambda_{\boldsymbol{\beta}}$ is large, the coefficients become small and the slope of the response function becomes steeper because the steepness is not penalized by the roughness. This effect must be tackled by additional constraints. As additional constraint, we fix the boundary coefficients of the basis expansion of the inner function $\sum_{s=1}^{k} \alpha_s \phi_s(\eta)$, that is, α_1 and α_k are fixed during the estimation procedure.

First we determine the maximum likelihood estimate (MLE) $\widehat{\boldsymbol{\beta}}_0^{ML} = (\widehat{\beta}_0^{ML}, (\widehat{\boldsymbol{\beta}}^{ML})^T)^T$ for the canonical response function $h_0(.)$. If the inner function is $h(\boldsymbol{x}^T\widehat{\boldsymbol{\beta}}^{ML}) = \widehat{\beta}_0^{ML} + \boldsymbol{x}^T\widehat{\boldsymbol{\beta}}^{ML}$ then by $h_0(h(\boldsymbol{x}^T\widehat{\boldsymbol{\beta}}^{ML})) = h_0(\widehat{\beta}_0^{ML} + \boldsymbol{x}^T\widehat{\boldsymbol{\beta}}^{ML})$ the ML model is fitted. By fixing the first and the last coefficient α_1 and α_k of the basis expansion of the inner function (cf. (7.2)) we estimate a monotonically increasing function $h(\eta) = \sum_{s=1}^{k} \alpha_s \phi_s(\eta_i)$ which is fixed at the endpoints of the line given by $\widehat{\beta}_0^{ML} + \eta$, where $\eta \in [b_l,\, b_u]$ is from a bounded interval of $\mathbb{R}$. Hence we have to assess the domain of the linear predictor $\eta = \boldsymbol{x}^T\boldsymbol{\beta}$. Therefor we compute the linear predictors of the MLE without intercept $\widehat{\boldsymbol{\eta}}^{ML} = \boldsymbol{X}\widehat{\boldsymbol{\beta}}^{ML}$ where $\boldsymbol{X}$ is the matrix of covariates and determine for an appropriately chosen c the range of the domain of the linear predictor in (7.2) respectively (7.3) to

$$[-u,\, u], \text{ with } u = c \cdot \max\left\{|\widehat{\boldsymbol{\eta}}^{ML}|\right\} \text{ and } c \geq 1. \tag{7.5}$$

The choice of c is a crucial point in this procedure especially if there are outliers in the new data set which is used for prediction. However, we initialize $\boldsymbol{\alpha}^{(0)}$ such that the line $\eta + \hat{\beta}_0^{ML}$ is approximately best in the domain $[-u,\, u]$ by $\Phi(\eta)\widehat{\boldsymbol{\alpha}}^{(0)}$, i.e. $\Phi(\eta)\widehat{\boldsymbol{\alpha}}^{(0)} \approx \eta + \hat{\beta}_0^{ML}$. Within the algorithm we fix $\alpha_1^{(l)} = \alpha_1^{(0)}$ and $\alpha_k^{(l)} = \alpha_k^{(0)}$ for each step l. So the set of all feasible solutions is restricted to smooth monotonically increasing functions with the boundary points $(-u,\, \alpha_1^{(0)})$ and $(u,\, \alpha_k^{(0)})$. At the boundaries of the domain of η all basis functions except for two are 0, namely $\phi_s(u) = \phi_s(-u) = 0$, $s = 2, ..., k-1$, and the boundary basis functions are 1, $\phi_1(-u) = \phi_k(u) = 1$ (see Figure 7.1). The constraints $\alpha_1^{(l)} = \alpha_1^{(0)}$ and $\alpha_k^{(l)} = \alpha_k^{(0)}$ are realized by solving a linear restricted weighted least squares problem in each step of the algorithm by using the `solve.QP` routine of the `R`-package `quadprog` (see Turlach, 2009).

In GLMs the response function is monotonically increasing. This property of the response function guarantees uniqueness. So we are also interested in monotonically increasing response functions. Generally a B-spline expansion generates a monotonically increasing function if the basis coefficients are monotonically increasing: $\alpha_s - \alpha_{s-1} \geq 0$, $s = 2, ..., k$. All in all, this is maintained by solving the corresponding weighted least

squares problem in the algorithm under the constraints

$$\begin{array}{rcl} \alpha_0^{(l)} & = & \alpha_0^{(0)} \\ \alpha_k^{(l)} & = & \alpha_k^{(0)} \\ \alpha_s^{(l)} - \alpha_{s-1}^{(l)} & \geq & 0, \ s = 2, \ ..., \ k. \end{array} \tag{7.6}$$

The constraints (7.6) are sumed up in the set

$$\mathcal{A} = \left\{ \boldsymbol{\alpha} \in \mathbb{R}^k \ : \ \alpha_1 = \alpha_1^{(0)}, \ \alpha_k = \alpha_k^{(0)}, \ \alpha_s - \alpha_{s-1} \geq 0, \ s = 2, \ ..., \ k \right\} \tag{7.7}$$

The constant c from (7.5) is not a tuning parameter but fixed. The model has two tuning parameters $\lambda_{\boldsymbol{\alpha}}$ and $\lambda_{\boldsymbol{\beta}}$. The choice of c is not only crucial because of outliers. Further if $\lambda_{\boldsymbol{\alpha}}$ becomes small c must increase to increase the range of the domain. We stop the algorithm if the estimate of the linear predictor is out of the fixed range. All in all, the penalized likelihood problem with the constraints is given by

$$\begin{array}{rcl} \left(\widehat{\boldsymbol{\alpha}}^T, \widehat{\boldsymbol{\beta}}^T \right)^T & = & \text{argmin} \left\{ -l_{pen}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \right\} \\ & = & \text{argmin} \Big\{ - l(\boldsymbol{\alpha}, \boldsymbol{\beta}) + \lambda_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_1 + \lambda_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha}, \\ & & \qquad \text{s.t.} \ -u \leq \boldsymbol{x}_i^T \boldsymbol{\beta} \leq u, \ i = 1, \ ..., \ n, \ u \text{ from (7.5)}, \\ & & \qquad \boldsymbol{\alpha} \in \mathcal{A} \text{ from (7.7)} \Big\}. \end{array} \tag{7.8}$$

7.3.1 Estimation Procedure

The goal is to solve the penalized likelihood problem with constraints (7.8). So we consider the penalized score equations $\boldsymbol{s}_{pen}(\boldsymbol{\beta}, \boldsymbol{\alpha}) = (\boldsymbol{s}_{pen}(\boldsymbol{\beta})^T, \boldsymbol{s}_{pen}(\boldsymbol{\alpha})^T)^T$. The corresponding (unpenalized) score equations are

$$\begin{array}{rcl} \boldsymbol{s}(\boldsymbol{\beta}) & = & \boldsymbol{x}^T \mathbf{D}_{\boldsymbol{\beta}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{y} - \boldsymbol{\mu}) \\ \boldsymbol{s}(\boldsymbol{\alpha}) & = & \boldsymbol{\Phi}(\eta)^T \mathbf{D}_{\boldsymbol{\alpha}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{y} - \boldsymbol{\mu}) \end{array} \tag{7.9}$$

where

$$\mathbf{D}_{\boldsymbol{\beta}} = \text{diag} \left\{ \frac{\partial h_0(h(\eta_i))}{\partial h(\eta)} \cdot \frac{\partial h(\eta_i)}{\partial \eta} \right\}_{i=1}^n \tag{7.10}$$

$$\mathbf{D}_{\boldsymbol{\alpha}} = \text{diag} \left\{ \frac{\partial h_0(h(\eta_i))}{\partial h(\eta)} \right\}_{i=1}^n \tag{7.11}$$

are the matrix of derivatives and

$$\boldsymbol{\Sigma} = \text{diag} \left\{ \sigma^2(h_0(h(\eta_i))) \right\}_{i=1}^n \tag{7.12}$$

is the matrix of variances. The score vectors corresponding to (7.3) are

$$\boldsymbol{s}_{pen}(\boldsymbol{\beta}) = \boldsymbol{s}(\boldsymbol{\beta}) - \lambda_{\boldsymbol{\beta}} \sum_{j=1}^{p} \text{sign}(\beta_j), \tag{7.13}$$

$$\boldsymbol{s}_{pen}(\boldsymbol{\alpha}) = \boldsymbol{s}(\boldsymbol{\alpha}) - 2\lambda_{\boldsymbol{\alpha}} \boldsymbol{K}\boldsymbol{\alpha}, \tag{7.14}$$

where

$$\text{sign}(\beta) = \begin{cases} 1, & \beta > 0 \\ 0, & \beta = 0 \\ -1, & \beta < 0. \end{cases}$$

If there exists a minimum for (7.3) $\boldsymbol{s}_{pen}(\boldsymbol{\beta}, \boldsymbol{\alpha}) = \boldsymbol{0}$ holds. The problem of discontinuity in (7.13) is tackled by Goeman (2010a). We adapt the algorithm from Goeman (2010a) and the corresponding `R`-package `penalized`. Goeman (2010a) proposes an alternative form of (7.13). Each component of the new score vector is given by

$$s_{pen}(\boldsymbol{\beta})_j = \begin{cases} s(\boldsymbol{\beta})_j - \lambda_\beta \,\text{sign}(\beta_j), & \beta_j \neq 0 \\ s(\boldsymbol{\beta})_j - \lambda_\beta \,\text{sign}\left(s_{pen}(\boldsymbol{\beta})_j\right), & \beta_j = 0 \text{ and } |s_{pen}(\boldsymbol{\beta})_j| > \lambda_\beta \\ 0, & \beta_j = 0 \text{ and } |s_{pen}(\boldsymbol{\beta})_j| \leq \lambda_\beta. \end{cases} \tag{7.15}$$

The second part of the score vector (7.14) is unchanged. The algorithm solves $s_{pen}(\boldsymbol{\beta})_j = 0$, $j = 1, ..., p$, where $s_{pen}(\boldsymbol{\beta})_j$ is from (7.15) and $s_{pen}(\boldsymbol{\alpha}) = \boldsymbol{0}$ which is from (7.14) alternately until convergence. Below with $^{(l)}$ we indicate the loops for solving $s_{pen}(\boldsymbol{\alpha}) = \boldsymbol{0}$ and $s_{pen}(\boldsymbol{\beta}) = \boldsymbol{0}$, with $s_{pen}(\boldsymbol{\beta})_j$ from (7.15). With $^{(m)}$ we indicate the loop which contains these both loops. In each iteration of the outer $^{(m)}$-loop $s_{pen}(\boldsymbol{\alpha}) = \boldsymbol{0}$ and $s_{pen}(\boldsymbol{\beta}) = \boldsymbol{0}$ are solved alternately.

Schedule of the SIPen-algorithm

- Outer Loop indicated by $^{(m)}$
  - First inner loop solving $s_{pen}(\boldsymbol{\beta}) = \boldsymbol{0}$, with $s_{pen}(\boldsymbol{\beta})_j$ from (7.15). It is indicated by $^{(l)}$. After convergence $\boldsymbol{\beta}^{(l+1)}$ is set to $\boldsymbol{\beta}^{(m)}$.
  - Second inner loop solving $s_{pen}(\boldsymbol{\alpha}) = \boldsymbol{0}$ indicated by $^{(l)}$. After convergence $\boldsymbol{\alpha}^{(l+1)}$ is set to $\boldsymbol{\alpha}^{(m)}$.
- m is incremented to $m + 1$

The solution of (7.14) is achieved by Fisher scoring. The estimation of L_1-penalized coefficients in GLMs with non-canonical link function is more complicate and described in the following.

Estimation for Fixed Response Function

In general a quadratic optimization problem is given by

$$\widehat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}}\{Q(\boldsymbol{\beta})\} = \text{argmin}_{\boldsymbol{\beta}}\left\{0.5 \cdot \boldsymbol{\beta}^T \boldsymbol{W} \boldsymbol{\beta} - \boldsymbol{c}^T \boldsymbol{\beta} - d\right\}.$$

$\widehat{\boldsymbol{\beta}}$ can be found by iterating

$$\widehat{\boldsymbol{\beta}}^{(l)} = \widehat{\boldsymbol{\beta}}^{(l-1)} - t \cdot \boldsymbol{g}(\widehat{\boldsymbol{\beta}}^{(l-1)})$$

until convergence. The procedure is well known as gradient descrent procedure. Here $\boldsymbol{g}(\widehat{\boldsymbol{\beta}}^{(l-1)}) = \frac{\partial}{\partial \boldsymbol{\beta}} Q(\widehat{\boldsymbol{\beta}}^{(l-1)})$ is the gradient and $0 < t$ a small step size. In the case of quadratical optimization problems the optimal step size is

$$t_{opt}^{(l-1)} = -\frac{\boldsymbol{g}(\widehat{\boldsymbol{\beta}}^{(l-1)})^T \boldsymbol{g}(\widehat{\boldsymbol{\beta}}^{(l-1)})}{\boldsymbol{g}(\widehat{\boldsymbol{\beta}}^{(l-1)})^T \boldsymbol{W} \boldsymbol{g}(\widehat{\boldsymbol{\beta}}^{(l-1)})}. \tag{7.16}$$

The basis of the algorithm described in the following is a gradient descent algorithm.

Let $\widehat{\boldsymbol{\alpha}}^{(m)}$ denote the basis coefficient vector of the previous step and let $h^{(m)}(.)$ be the corresponding response function. With $l_{pen}(\widehat{\boldsymbol{\alpha}}^{(m)}, \boldsymbol{\beta})$ we denote the log-likelihood function with fixed response function $h_0(\Phi(\eta)\widehat{\boldsymbol{\alpha}}^{(m)})$. The corresponding regression problem $\widehat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}}\left\{-l_{pen}(\boldsymbol{\alpha}^{(m)}, \boldsymbol{\beta})\right\}$ is solved by an iteratively weighted least squares algorithm. $l_{pen}(\boldsymbol{\alpha}^{(m)}, \boldsymbol{\beta})$ is not differentiable in each point but there are 3^p subdomains where $\frac{\partial}{\partial \boldsymbol{\beta}} l_{pen}(\boldsymbol{\alpha}^{(m)}, \boldsymbol{\beta})$ is continuous. In each of these subdomains $\text{sign}(\boldsymbol{\beta}) = (\text{sign}(\beta_1), ..., \text{sign}(\beta_p))^T$ is constant. An upper bound to guarantee this condition is

$$t_{edge}^{(l-1)} = \min\left\{-\frac{\beta_i^{(l-1)}}{s_{pen}(\boldsymbol{\beta})_i^{(l-1)}} \ : \ \text{sign}(\beta_i^{(l-1)}) = -\,\text{sign}(s_{pen}(\boldsymbol{\beta})_i^{(l-1)}) \neq 0\right\}.$$

Using the statistical framework, the optimal stepsize $t_{opt}^{(l-1)}$ from (7.16) becomes

$$t_{opt}^{(l-1)} = -\frac{\boldsymbol{s}_{pen}(\widehat{\boldsymbol{\beta}}^{(l-1)})^T \boldsymbol{s}_{pen}(\widehat{\boldsymbol{\beta}})}{\boldsymbol{s}_{pen}(\widehat{\boldsymbol{\beta}}^{(l-1)})^T \boldsymbol{X}^T \boldsymbol{D}^T_{\widehat{\boldsymbol{\beta}}^{(l-1)}} (\boldsymbol{\Sigma}_{\widehat{\boldsymbol{\beta}}^{(l-1)}})^{-1} \boldsymbol{D}_{\widehat{\boldsymbol{\beta}}^{(l-1)}} \boldsymbol{X} \boldsymbol{s}_{pen}(\widehat{\boldsymbol{\beta}}^{(l-1)})}.$$

Here $\boldsymbol{D}_{\widehat{\boldsymbol{\beta}}^{(l-1)}}$ has the form of (7.10), where the response function $h_0(\widehat{h}^{(m)}(.))$ is fixed and evaluated at the estimate of the previous step $\boldsymbol{\beta}^{(l-1)}$,

$$\mathbf{D}_{\widehat{\boldsymbol{\beta}}^{(l-1)}} = \text{diag}\left\{\frac{\partial h_0(\widehat{h}^{(m)}(\widehat{\eta}_i^{(l-1)}))}{\partial \widehat{h}^{(m)}(\eta)} \cdot \frac{\partial \widehat{h}^{(m)}(\widehat{\eta}_i^{(l-1)})}{\partial \eta}\right\}_{i=1}^{n}. \tag{7.17}$$

According to this

$$\boldsymbol{\Sigma}_{\widehat{\boldsymbol{\beta}}^{(l-1)}} = \text{diag}\left\{\sigma^2(h_0(\widehat{h}^{(m)}(\boldsymbol{x}_i^T \boldsymbol{\beta}^{(l-1)})))\right\}_{i=1}^{n} \tag{7.18}$$

denotes the matrix of variances for fixed response function evaluate at the previous iteration step. So each iteration step has the form

$$\widehat{\boldsymbol{\beta}}^{(l)} = \widehat{\boldsymbol{\beta}}^{(l-1)} - \min\left\{t_{opt}^{(l-1)}, t_{edge}^{(l-1)}\right\} \boldsymbol{s}_{pen}(\widehat{\boldsymbol{\beta}}^{(l-1)}).$$

It is well known that the gradient descent algorithm needs more iterations for convergence in comparison to Fisher scoring. We denote by $\widehat{\boldsymbol{\beta}}_{FS}^{(l)}$ the Fisher scoring based estimate of the lth iteration. Each iteration step of the gradient descent algorithm works in a subdomain where $\text{sign}(\boldsymbol{\beta}_{GD}) := \lim_{t\downarrow 0} \text{sign}(\boldsymbol{\beta} - t\boldsymbol{s}_{pen}(\boldsymbol{\beta}))$ does not change. So if $\text{sign}(\widehat{\boldsymbol{\beta}}_{FS}^{(l)}) = \text{sign}(\widehat{\boldsymbol{\beta}}_{GD}^{(l-1)})$ and $t_{opt}^{(l-1)} < t_{edge}^{(l-1)}$ hold, the Fisher scoring estimate is reasonable. In other words the Fisher scoring step works inner one of the 3^p subdomains where $\text{sign}(\boldsymbol{\beta})$ is constant. So we can restrict the problem on an m-dimensional subspace, where $m < p$ is the number of non-zero elements of $\text{sign}(\widehat{\boldsymbol{\beta}}_{GD}^{(l-1)})$. We denote by $\mathcal{M} = \left\{j : \text{sign}(\widehat{\boldsymbol{\beta}}_{GD,j}^{(l-1)}) \neq 0\right\}$ the index set of corresponding covariates. The parameter vector, the score function and the Fisher information matrix to the corresponding m-dimensional subspace are termed by $\widehat{\boldsymbol{\beta}}_{\mathcal{M}}^{(l-1)}$, $\boldsymbol{s}_{pen}(\boldsymbol{\beta}_{\mathcal{M}}^{(l-1)})$, and

$$\boldsymbol{F}(\widehat{\boldsymbol{\beta}}_{\mathcal{M}}^{(l-1)}) = -\frac{\partial^2 l_{pen}(\boldsymbol{\beta}_{\mathcal{M}}^{(l-1)})}{\partial\boldsymbol{\beta}_{\mathcal{M}}\partial\boldsymbol{\beta}_{\mathcal{M}}^T} = -\boldsymbol{X}_{\mathcal{M}}^T\boldsymbol{D}_{\widehat{\boldsymbol{\beta}}_{\mathcal{M}}^{(l-1)}}^T\boldsymbol{\Sigma}_{\widehat{\boldsymbol{\beta}}_{\mathcal{M}}^{(l-1)}}^{-1}\boldsymbol{D}_{\boldsymbol{\beta}_{\mathcal{M}}^{(l-1)}}\boldsymbol{X}_{\mathcal{M}}.$$

The components of the Fisher information matrix $\boldsymbol{D}_{\widehat{\boldsymbol{\beta}}_{\mathcal{M}}^{(l)}}$ and$\boldsymbol{\Sigma}_{\widehat{\boldsymbol{\beta}}_{\mathcal{M}}^{(l-1)}}$ correspond to (7.17) and (7.18) respectively. So one iteration step of the Fisher scoring is

$$\widehat{\boldsymbol{\beta}}_{\mathcal{M}}^{(l)} = \widehat{\boldsymbol{\beta}}_{\mathcal{M}}^{(l-1)} + \boldsymbol{F}(\widehat{\boldsymbol{\beta}}_{\mathcal{M}}^{(l-1)})^{-1}\boldsymbol{s}_{pen}(\widehat{\boldsymbol{\beta}}_{\mathcal{M}}^{(l-1)}).$$

Note that $\widehat{\boldsymbol{\beta}}_{\mathcal{M}}^{(l)}$ is an m-dimensional vector, where $m = |\mathcal{M}|$. So we have to augment the vector by $p - m$ components which are equal to zero. Hence we create a p-dimensional vector where non-zero components are placed on entries corresponding to $\mathcal{M}$ and zero components are placed on entries indicated by $\overline{\mathcal{M}} = \left\{j : \text{sign}(\widehat{\boldsymbol{\beta}}_{GD,j}^{(l-1)}) = 0\right\}$. By this we obtain $\boldsymbol{\beta}_{FS}^{(l)}$. Summarizing these results each update step has the form

$$\boldsymbol{\beta}^{(l)} = \begin{cases} \boldsymbol{\beta}^{(l-1)} + t_{edge}^{(l-1)}\boldsymbol{s}_{pen}(\boldsymbol{\beta}^{(l-1)}), & t_{opt}^{(l-1)} \geq t_{edge}^{(l-1)} \\ \boldsymbol{\beta}_{FS}^{(l)}, & t_{opt}^{(l-1)} < t_{edge}^{(l-1)} \text{ and } \text{sign}(\boldsymbol{\beta}_{FS}^{(l)}) = \text{sign}(\boldsymbol{\beta}_{GD}^{(l-1)}) \\ \boldsymbol{\beta}^{(l-1)} + t_{opt}^{(l-1)}\boldsymbol{s}_{pen}(\boldsymbol{\beta}^{(l-1)}), & \text{otherwise.} \end{cases} \quad (7.19)$$

Estimation for Fixed Predictor

As mentioned before the estimation of the inner function $\widehat{h}(.)$ with fixed predictor $\boldsymbol{\eta}^{(m)}$ is a Fisher scoring with linear constraints given in (7.7). Each Fisher scoring step has the form

$$\begin{aligned} \widehat{\boldsymbol{\alpha}}^{(l)} \;=\; \text{argmin}_{\boldsymbol{\alpha}} \Big\{ & \boldsymbol{\alpha}^T\boldsymbol{\Phi}^T\boldsymbol{W}_{\widehat{\boldsymbol{\alpha}}^{(l-1)}}\boldsymbol{\Phi}\boldsymbol{\alpha} - 2\boldsymbol{\alpha}^T\boldsymbol{\Phi}^T\boldsymbol{W}_{\widehat{\boldsymbol{\alpha}}^{(l-1)}}(\boldsymbol{D}_{\widehat{\boldsymbol{\alpha}}^{(l-1)}})^{-1}(\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^{(l-1)}) \\ & + \boldsymbol{\alpha}^T\boldsymbol{K}\boldsymbol{\alpha}, \text{ s.t. } \boldsymbol{\alpha} \in \mathcal{A}\Big\}, \end{aligned} \quad (7.20)$$

where $\widehat{\boldsymbol{\mu}}^{(l-1)} = h_0(\boldsymbol{\Phi}\widehat{\boldsymbol{\alpha}}^{(l-1)})$ and

$$\begin{aligned}
\boldsymbol{W}_{\widehat{\boldsymbol{\alpha}}^{(l-1)}} &= \boldsymbol{D}^T_{\widehat{\boldsymbol{\alpha}}^{(l-1)}} \boldsymbol{\Sigma}^{-1}_{\widehat{\boldsymbol{\alpha}}^{(l-1)}} \boldsymbol{D}_{\widehat{\boldsymbol{\alpha}}^{(l-1)}} \\
\boldsymbol{D}_{\widehat{\boldsymbol{\alpha}}^{(l-1)}} &= \operatorname{diag} \left\{ \frac{\partial h_0(\widehat{h}^{(l-1)}(\eta_i^{(m)}))}{\partial \widehat{h}^{(l-1)}(\eta)} \right\}^n_{i=1} \\
\boldsymbol{\Sigma}_{\widehat{\boldsymbol{\alpha}}^{(l)}} &= \operatorname{diag} \left\{ \sigma^2(h_0(\widehat{h}^{(l-1)}(\eta_i^{(m)}))) \right\}^n_{i=1}
\end{aligned}$$

(7.20) is iterated until convergence. One should keep in mind that $\boldsymbol{\Phi}$ depends on $\boldsymbol{\eta}^{(m)}$ and the elements indicated by $^{(m)}$ remain unchanged during this constrained Fisher scoring.

Algorithm: SIPen

Step 1 (Initialization)

Determine the MLE and compute the domain of $\boldsymbol{\eta}$ as described in (7.5). Compute $\widehat{\boldsymbol{\alpha}}^{(0)}$ as described and fix $\widehat{\boldsymbol{\alpha}}^{(0)}_1$ and $\widehat{\boldsymbol{\alpha}}^{(0)}_k$. Choose for $\widehat{\boldsymbol{\beta}}^{(0)} \neq \mathbf{0}$ a p-dimensional vector with small Euclidean length and compute $\widehat{\boldsymbol{\eta}}^{(0)} = \boldsymbol{X}\widehat{\boldsymbol{\beta}}^{(0)}$. Set $m = 0$ and $\widehat{\boldsymbol{\alpha}}^{(m)} = \widehat{\boldsymbol{\alpha}}^{(0)}$. Compute $\boldsymbol{D}_{\widehat{\boldsymbol{\beta}}^{(0)}}$ as described in (7.17) and $\boldsymbol{\Sigma}_{\widehat{\boldsymbol{\beta}}^{(0)}}$ as described in (7.18).

Step 2 (Iteration)

While $\|\widehat{\boldsymbol{\alpha}}^{(m)} - \widehat{\boldsymbol{\alpha}}^{(m-1)}\| / \|\widehat{\boldsymbol{\alpha}}^{(m-1)}\| < \delta$ and $\|\widehat{\boldsymbol{\beta}}^{(m)} - \widehat{\boldsymbol{\beta}}^{(m-1)}\| / \|\widehat{\boldsymbol{\beta}}^{(m-1)}\| < \delta$ and $\boldsymbol{X}\boldsymbol{\beta}^{(m)} \in [-u, u]$.

1. *Predictor estimation*
   - Minimize the negative log-likelihood $-l(\widehat{\boldsymbol{\alpha}}^{(m)}, \boldsymbol{\beta})$ by the gradient descent estimator described in (7.19). This loop is indicated by $^{(l)}$.
   - After convergence of (7.19) set $\widehat{\boldsymbol{\beta}}^{(m)} = \widehat{\boldsymbol{\beta}}^{(l)}$.
2. *Response function estimation*
   - Compute $\boldsymbol{\eta}^{(m)} = \boldsymbol{X}\widehat{\boldsymbol{\beta}}^{(m)}$.
   - Minimize negative log-likelihood $-l(\boldsymbol{\alpha}, \widehat{\boldsymbol{\beta}}^{(m)})$ by the Fisher scoring described in (7.20), where this loop is also indicated by $^{(l)}$.
   - Set $\widehat{\boldsymbol{\alpha}}^{(m)} = \widehat{\boldsymbol{\alpha}}^{(l)}$

Increase m by 1.

7.3.2 Solution Path

In section 7.4 we compare the SIPen algorithm with other procedures. We pick out setting Pois_a for the illustration of a solution path. Pois_a is defined by the parameter vector

$$\boldsymbol{\beta}_a = (0.2,\, 0.4,\, -0.4,\, 0.8,\, \underbrace{0,\, ...,\, 0}_{16})^T$$

and the response function

$$h(\eta) = 10/(1 + \exp(-10 \cdot \eta)).$$

For details see section 7.4. We fix the tuning parameter $\lambda_{\boldsymbol{\alpha}} = 10$. The second tuning parameter $\lambda_{\boldsymbol{\beta}}$ accounts for the path. It is based on an equidistant grid $\boldsymbol{s}$ from log(0.999) to log(0.001) with 100 values, $\boldsymbol{s} = (\log(0.999),\, ...,\, \log(0.001))$. The grid of the second tuning parameter is given by

$$\lambda_{\boldsymbol{\beta}} \in \{\lambda_{\max,\boldsymbol{\beta}} \cdot \exp(\boldsymbol{s})\} \tag{7.21}$$

and is not equidistant. For the difference of two small adjacent components of $\exp(\boldsymbol{s})$ is small and the difference of two large adjacent components becomes large. The exponentiated sequence $\exp(\boldsymbol{s})$ is scale by

$$\lambda_{\max,\boldsymbol{\beta}} = \max\left\{\boldsymbol{D}^{(0)}(\boldsymbol{\Sigma}^{(0)})^{-1}|\boldsymbol{X}^T\boldsymbol{y}|\right\}$$

where $\boldsymbol{D}^{(0)} = \frac{\partial h_0(\bar{\boldsymbol{y}})}{\partial \eta}\boldsymbol{I}$ and $\Sigma^{(0)} = \sigma(h_0(h^{(0)}(\bar{y})))\boldsymbol{I}$. $\lambda_{\max,\boldsymbol{\beta}}$ is an upper bound for sequences of $\lambda_{\boldsymbol{\beta}}$ in the case of GLMs (cf. Park and Hastie, 2007b). In Figure 7.2 the coefficient build ups for decreasing $\lambda_{\boldsymbol{\beta}}$ are shown. Additionally the estimated response functions at different points of the solution path are illustrated. We chose different numbers of elements $\{1,\, 25,\, 35,\, 60,\, 100\}$ of the $\lambda_{\boldsymbol{\beta}}$ sequence for the in illustration of the response functions. The response function chances with increasing $L1$-norm of the parameter vector. While the $L1$-norm of $\boldsymbol{\beta}$, i.e. $|\boldsymbol{\beta}|$, increases the variable selection declines. In Figure 7.3 we use the same setting but chance the tuning parameter $\lambda_{\boldsymbol{\alpha}} = 1$. It is seen, that for small values of $|\boldsymbol{\beta}|$ the response function looks like the canonical response function. The shrinkage effect is equalized by the response function. For increasing length $L1$-norm of $\boldsymbol{\beta}$ the response function is still estimated quite well but it becomes steeper for increasing length $|\boldsymbol{\beta}|$. Similar to the path for $\lambda_{\boldsymbol{\alpha}} = 10$ the shrinkage effect is equalized by the steepness of the response function. The models are sparse for $\lambda_{\boldsymbol{\alpha}} = 10$ and $\lambda_{\boldsymbol{\alpha}} = 1$. The 5fold crossvalidation chose $\lambda_{\boldsymbol{\alpha}} = 1$ because of its lower crossvalidation score, which is 201.2925 for $\lambda_{\boldsymbol{\alpha}} = 1$ in contrast to 269.1787 for $\lambda_{\boldsymbol{\alpha}} = 10$. The shrinkage effect of the non-influential variables for $\lambda_{\boldsymbol{\alpha}} = 1$ is much stronger than for $\lambda_{\boldsymbol{\alpha}} = 10$. This becomes clearer by having a closer look on the

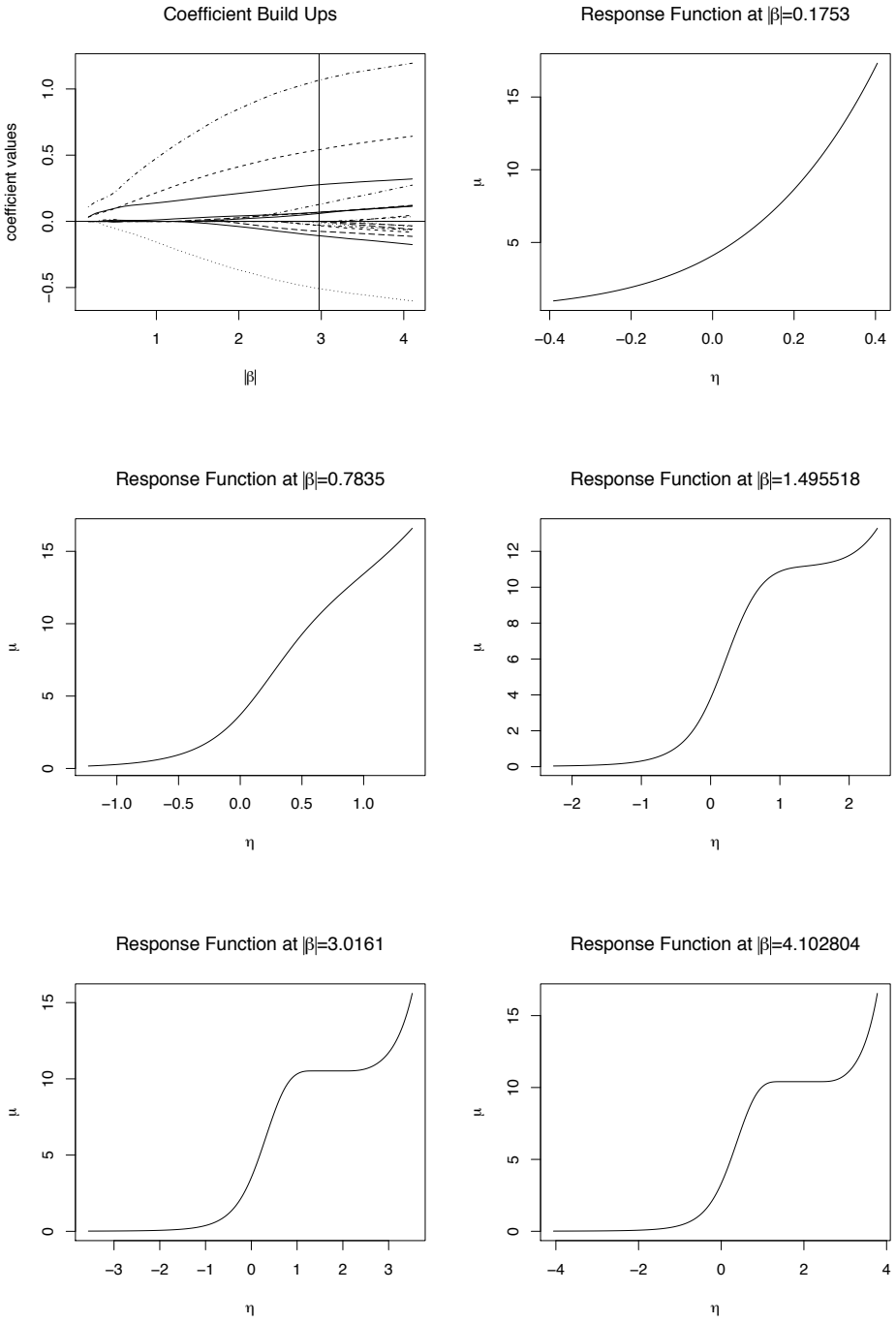

FIGURE 7.2: *Coefficient build ups and response function for* $\lambda_{\boldsymbol{\alpha}} = 10$. *The vertical line in the figure of the coefficient build ups shows the optimal parameter vector determined by* 5*fold crossvalidation.*

estimates for the optimal tuning parameter $\lambda_{\boldsymbol{\beta}}$ determined by a 5fold crossvalidation:

$$\begin{aligned}
\widehat{\boldsymbol{\beta}}_{10} &= (0.2763,\ 0.5413,\ -0.5086,\ 1.0677 \\
&\quad ,\ 0.0000,\ 0.0591,\ 0.0000,\ 0.0000,\ -0.0282,\ 0.0671,\ 0.0692,\ 0.0000 \\
&\quad ,\ 0.0000,\ 0.0000,\ -0.0009,\ -0.1088,\ -0.0317,\ -0.0111,\ 0.1275,\ -0.0757)^T \\
\widehat{\boldsymbol{\beta}}_{1} &= (0.2080,\ 0.4769,\ -0.4432,\ 0.9802 \\
&\quad ,\ 0.0000,\ 0.0231,\ 0.0000,\ 0.0000,\ -0.0113,\ 0.0362,\ 0.0390,\ 0.0000 \\
&\quad ,\ 0.0000,\ 0.0000,\ 0.0000,\ -0.0203,\ -0.0157,\ -0.0078,\ 0.0268,\ -0.0360)^T
\end{aligned}$$

These estimates are given by the horizontal line in the coefficient build ups in Figure 7.2 and 7.3, $\widehat{\boldsymbol{\beta}}_{10}$ corresponding to $\lambda_{\boldsymbol{\alpha}} = 10$ and $\widehat{\boldsymbol{\beta}}_{1}$ corresponding to $\lambda_{\boldsymbol{\alpha}} = 1$. Another important aspect is the form of the response function at the right boundary. It increases very strongly so that outside of the range of the linear predictor given by the data the predictive performance becomes worse.

7.4 Simulation Studies

The settings depend on the distributional assumption, the response function, and the parameter vector of the linear predictor. We consider the following two parameter vectors of length $p = 20$:

$$\begin{aligned}
\boldsymbol{\beta}_a &= (0.2,\ 0.4,\ -0.4,\ 0.8,\ 0,\ ...,\ 0)^T \\
\boldsymbol{\beta}_b &= (0.5,\ 0.5,\ -0.5,\ -0.5,\ 0,\ ...,\ 0)^T.
\end{aligned}$$

We investigate normal, Poisson, and binomial distributed responses. Each setting is combined with one non-canonical response function. For the normal and the Poisson distribution we use a sigmoidal response function

$$h(\eta) = \frac{10}{1 + \exp(-10 \cdot \eta)}.$$

The responses are generated by $y_i = N(h(\eta_i),\ 1)$ and $y_i = Pois(h(\eta_i))$, respectively. In the binomial case we choose a step function

$$h(\eta) = \begin{cases} 0.1 & 2\eta < -1 \\ 0.5 & -1 \leq 2\eta \leq 1 \\ 0.9 & 2\eta > 1 \end{cases}$$

and the response is given by $y_i = Bin(h(\eta_i))$. So we have 6 different settings denoted by $\text{distribution}_{\text{parameter vector}}$, for example Norm_a denotes the setting with normal distributed response and parameter vector β_a.

We compare the new procedure with the cut and uncut version of the FlexLink procedure of Tutz and Petry (2011). The FlexLink estimates the parameter vector and the

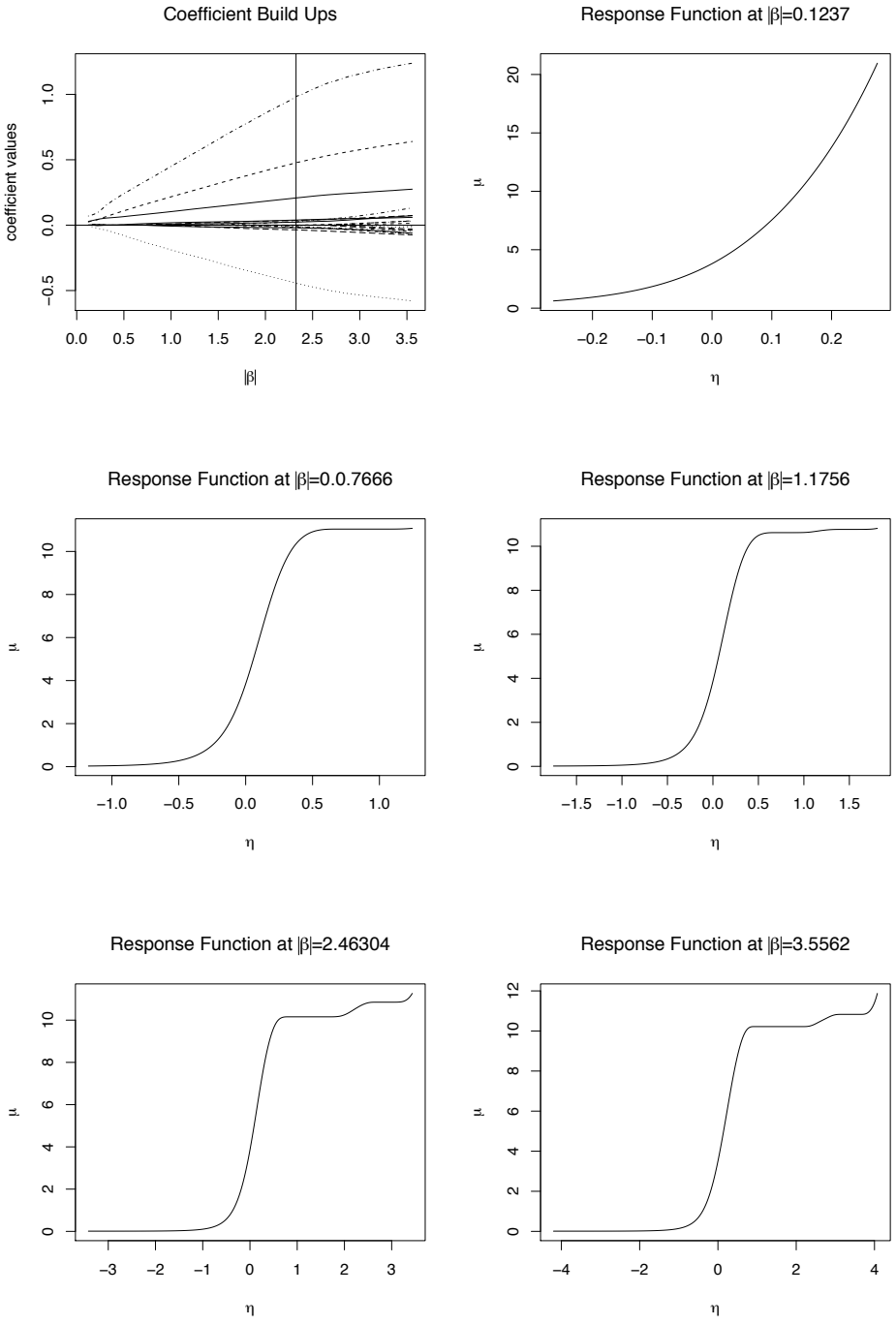

FIGURE 7.3: *Coefficient Build Ups and response function for* $\lambda_{\boldsymbol{\alpha}} = 1$. *The vertical line in the figure of the coefficient build ups shows the optimal parameter vector determined by* 5*fold crossvalidation.*

link function by boosting techniques and is a direct competitor. Further we use two established methods which use the canonical response function, namely `mboost` and `glmnet`. The `mboost` is a boosting technique and the `glmnet` is an efficient algorithm to solve an $L1$-penalized GLM. The algorithms are available in the `R`-packages `mboost` proposed by Hothorn et al. (2009) and `glmnet` proposed by Friedman et al. (2008). All tuning parameters are determined by a 5fold crossvalidation. We use the following parameter values:

SIPen We chose $\lambda_{\boldsymbol{\beta}}$ as described in section 7.3 and $\lambda_{\boldsymbol{\alpha}} \in \{100, 50, 20, 10, 5, 2, 1\}$. Further we use a basis expansion of degree 3 with 100 inner knots.

FlexLink The smoothing parameter of the response function is chosen from $\lambda \in \{100, 10, 1, 0.1, 0.01\}$ and the maximal number of boosting iterations is fixed by $M = 1000$. The B-spline basis expansion has 50 inner knots and degree 3. The update stepsize for the predictor and the response function is $\nu_f = \nu_h = 0.1$.

mboost In general default values are used but the maximal number of boosting iterations is set to $M = 1000$. For details see Hothorn et al. (2009) and Hothorn et al. (2010)

glmnet The default values are used but the maximal number of different λ values is set to 100. The `glmnet` is proposed by Simon et al. (2011) with the corresponding `R`-package by Friedman et al. (2008).

Each data set is split into a training and a test data set. The training data set is used to determine the optimal tuning parameter by 5-fold crossvalidation and contains $n_{train} = 200$ observations. We give the predictive performance by the predictive deviance on the independent test data set with $n_{test} = 1000$ given by

$$\text{Dev(test)} = -2 \left[\sum_{i=1}^{n_{test}} l(y_i^{test}, \widehat{\mu}_i) - (y_i^{test}, y_i^{test}) \right],$$

where $\widehat{\mu}_i = h_0(\sum_{i=1}^{k} \widehat{\alpha}_k \phi_k(\widehat{\beta}_0 + \boldsymbol{x}_i^T \widehat{\boldsymbol{\beta}}))$, y_i^{test} is the ith observation of the test data set and the parameters $\widehat{\boldsymbol{\alpha}}$ and $\widehat{\boldsymbol{\beta}}_0$ are estimated on the training data set. The accuracy of $\widehat{\boldsymbol{\beta}}$ can not be given by the ordinary mean squared error $\sum_{j=1}^{p}(\widehat{\beta}_j - \beta_j)^2$, since a scaling of the estimate $\widehat{\boldsymbol{\beta}}$ can be absorbed in the response function. So we standardize $\widehat{\boldsymbol{\beta}}$ to Euclidean length 1 $\widetilde{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}/\|\widehat{\boldsymbol{\beta}}\|$ and give

$$\text{MSE}_{\boldsymbol{\beta}} = \sum_{j=1}^{p} (\widetilde{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_j)^2,$$

where $\boldsymbol{\beta}_j$ is the jth component of the true parameter vector, i.e. $\boldsymbol{\beta}_a$ or $\boldsymbol{\beta}_b$. It is remarkable that the true parameter vectors have Euclidean length 1. Additionally we give the mean of the hit rate (rate of influential covariates which are selected) and false positive rate (rate of non-influential covariates which are selected). The results are summed up in Table 7.1. In the binary case there is only weak information and the results are quite similar across

		SIPen	FlexLink	FlexLink (cut)	glmnet	mboost
Normal distribution						
Norm_a	$\text{MSE}_{\boldsymbol{\beta}}$	0.0294	0.0162	0.0174	0.1418	0.1498
	Dev(test)	1254.88	1954.13	1995.50	24277.33	24280.76
	hit	1.000	1.000	1.000	1.000	1.000
	false positve	0.795	0.000	0.000	0.344	0.375
Norm_b	$\text{MSE}_{\boldsymbol{\beta}}$	0.0253	0.0117	0.0145	0.0918	0.0980
	Dev(test)	1221.60	2019.21	2040.61	20614.36	20592.74
	hit	1.000	1.000	1.000	1.000	1.000
	false positve	0.895	0.000	0.000	0.350	0.403
Poisson distribution						
Pois_a	$\text{MSE}_{\boldsymbol{\beta}}$	0.0789	0.0492	0.0483	0.2471	0.2579
	Dev(test)	925.97	839.64	839.19	3353.96	3328.63
	hit	1.000	1.000	1.000	0.965	0.980
	false positve	0.605	0.016	0.006	0.266	0.251
Pois_b	$\text{MSE}_{\boldsymbol{\beta}}$	0.0744	0.0395	0.0434	0.1536	0.1745
	Dev(test)	959.30	935.42	936.75	3109.93	3099.30
	hit	1.000	1.000	1.000	1.000	1.000
	false positive	0.701	0.006	0.001	0.305	0.328
Binomial distribution						
Bin_a	$\text{MSE}_{\boldsymbol{\beta}}$	0.4165	0.4098	0.4049	0.4074	0.4230
	Dev(test)	1064.79	1071.01	1067.73	1138.14	1066.23
	hit	0.860	0.7950	0.810	0.855	0.865
	false positve	0.323	0.128	0.126	0.316	0.339
Bin_b	$\text{MSE}_{\boldsymbol{\beta}}$	0.3343	0.3762	0.3747	0.3391	0.3288
	Dev(test)	1135.00	1138.29	1140.91	1199.77	1137.23
	hit	1.000	0.995	0.995	1.000	1.000
	false positve	0.395	0.200	0.186	0.370	0.391

TABLE 7.1: *Medians of the model assessment measures for the settings of the simulation study.*

all procedures. We illustrate the results of the more interesting normal and Poisson cases Norm_a and Pois_a by boxplots in Figure 7.4.

The `SIPen` outperforms the estimating procedures with (fixed) canonical response function, namely `mboost` and `glmnet`, in each setting. In the normal and the binomial case the predictive performance of the `SIPen` is better than the predictive performance of both `FlexLink` procedures. In the normal case the predictive deviance of the `SIPen` is the best in comparison with the other methods but the accuracy of the parameter estimate degrades, i.e. the $\text{MSE}_{\boldsymbol{\beta}}$ is the worst. The $L1$-penalty combines shrinkage and variable selection. But the shrinkage effect can be partially equalized by the flexible response function. Hence many non-influential covariates are shrunken very much but not shrunken to zero. The predictive test deviance improves without hard variable selection. This effect is known

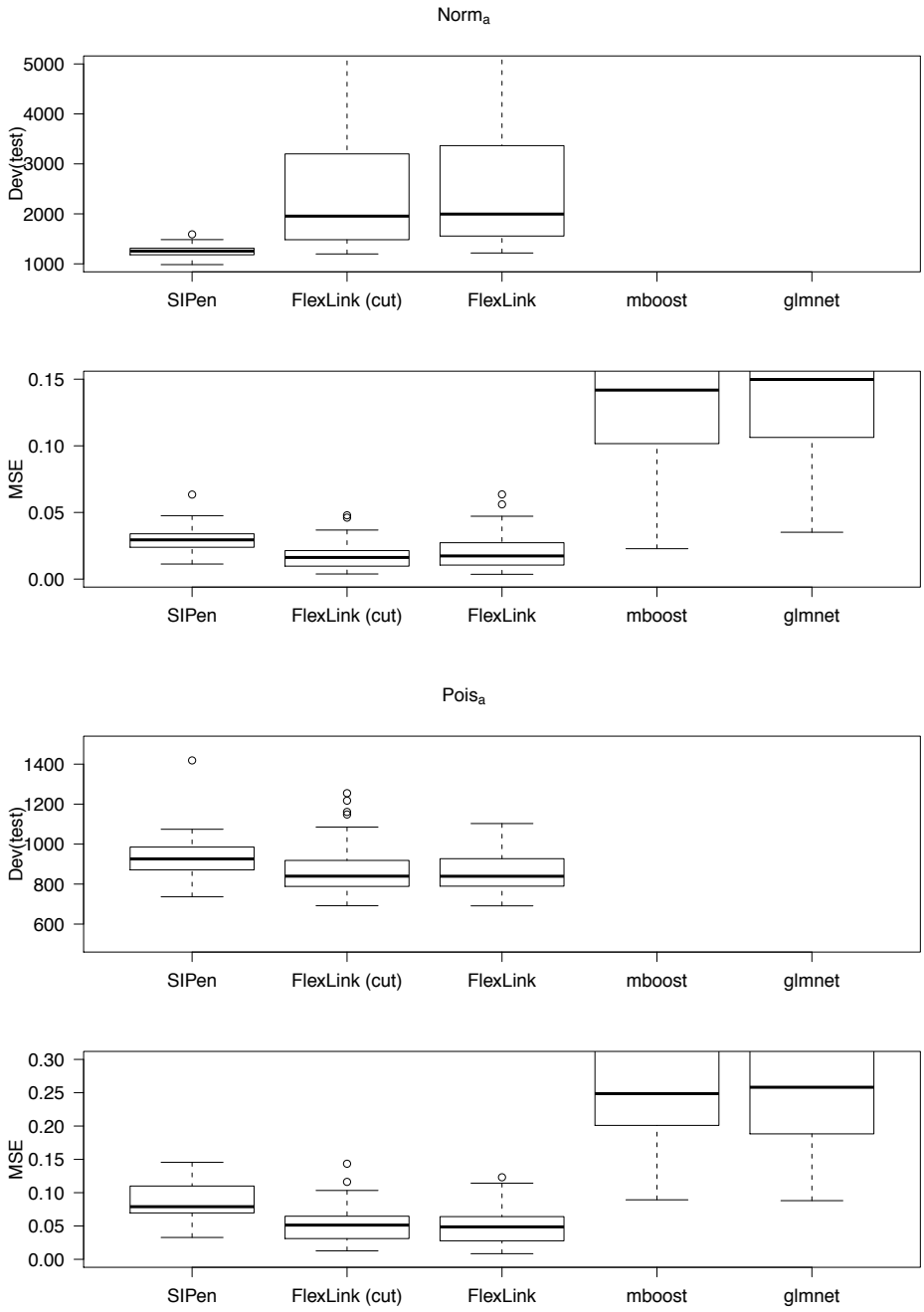

FIGURE 7.4: *Boxplots for* $\text{MSE}_{\boldsymbol{\beta}}$ *and* $\text{Dev}(test)$ *of the normal and the Poisson case settings. For* Norm_a *the predictive deviances of* `mboost` *and* `glmnet` *are not shown because they are much higher then for the other procedures.*

from the ridge regression (see Hoerl and Kennard, 1970) and so the hits and false positive rates of the `SIPen` are not very satisfying but nevertheless the predictive performance im-

proves. For the Poisson case the predictive performance of `FlexLink` and `FlexLink (cut)` is better than the predictive performance of `SIPen`. This is caused by the outer function chosen as canonical response function $h_0(.) = \exp(.)$. The inner function $h(.)$ can be very steep outside of the range of the linear predictor generated by the training data. The outer function exponentiated the inner function and so the predictive deviance worsens if many observations of the test data set are out of the range given by the training data.

7.5 Data Examples

7.5.1 Sao Paulo Air Pollution Data Set

The data set considered in the following concerns air pollution and its influence on the number of deaths caused by respiratory reasons. For the number of death a Poisson model is used. The data are recorded from January 1994 to December 1997 in the metropolian area for Sao Paulo. We consider only the first year of the recording. There are 5 missings and so we have 360 observations. The whole data set is available on the web at `http://www.ime.usp.br/~jmsinger/Polatm9497.zip`. We use a subset of covariates that was also used by Leitenstorfer and Tutz (2007). The response variable `RES65` is the number of deaths caused by respiratory reasons of people which are 65 years or older per day. The covariates are given in Table 7.2. The covariates `SO2ME.2`, `TMIN.2`, and `UMID`

Number	Label	Explanation
1	`TEMPO`	Time in days
2	`SEGUNDA`	Weekday of record: Monday.
3	`TERCA`	Weekday of record: Tuesday.
4	`QUARTA`	Weekday of record: Wednesday.
5	`QUINTA`	Weekday of record: Thursday.
6	`SEXTA`	Weekday of record: Friday.
7	`SABADO`	Weekday of record: Saturday.
8	`SO2ME.2`	The 24-hours mean of SO_2 concentration (in μ/m^3) over all monitoring measurement stations.
9	`TMIN.2`	The daily minimum temperature.
10	`UMID`	The daily relative humidity.
11	`CAR65`	Cardialogical caused deaths per day.
12	`OTH65`	Other (non respiratory or cardiologial) caused deaths per day.

TABLE 7.2: *Table of covariates and their labels of the Sao Paulo air pollution data set.*

describe the air quality. The covariates `SEGUNDA`, `TERCA`, `QUARTA`, `QUINTA`, `SEXTA`, `SABADO` describe effects within one week and `TEMPO` represents the effect over the record time. The covariates `CAR65` and `OTH65` represent a general health status. So there are representatives for time, general health, and air quality reasoned effects.

We split the complete data into a training and a test data set are repeat the procedure 50 times. The training data set contains 300 observation and is used to fit the model. The tuning parameters are determined by 5fold crossvalidation on the training data set. The test data sets are used for the calculation of the predictive test deviance. We present the boxplots of the predictive test deviance and the plots of the selected covariates across the 50 random splits in Figure 7.5. The tuning parameters for the SIPen are chosen from $\lambda_{\boldsymbol{\alpha}} \in \{1000, 100, 10, 1\}$ and the sequence of $\lambda_{\boldsymbol{\beta}}$ is determined as described above. The tuning parameter of both `FlexLink` procedures are $\lambda_h \in \{0.01, 0.1, 1, 10, 100\}$ and the maximal number of boosting iterations is $M = 1000$. The set of selected covariates across the random splits of the `glmnet` and `SIPen` are quite similar. The models estimated by the `FlexLink` and `FlexLink (cut)` are the sparsest. The selections property of `mboost` is instabil, either many or only very few are selected. In general the trend over the year, i.e. `TEMPO`, is not selected very often. In the mids of the week day-specific covariates are selected. Especially `glmnet` and `SIPen` select Tuesday and Thursday quite often. The covariates which describe the air quality `SO2ME.2`, `TMIN.2`, and `UMID` are selected by all procedures very often. `glmnet` and `SIPen` select these variables in each random split. The `mboost` includes the `UMID` only in combination with other covariates. The boosting procedures `FlexLink`, `FlexLink (cut)`, and `mboost` are not so stable by including the group of air quality covariates, namely `SO2ME.2`, `TMIN.2`, and `UMID`. Especially the `mbbost` includes `UMID` very instabil. The covariate `CAR65` is also selected quite often (always by `glmnet` and `SIPen`) in contrast to `OTH65`. In general the differences of the predictive test deviance are not very great. The `SIPen` outperforms the other response function estimating procedures `FlexLink` and `FlexLink (cut)` with respect to the median of predictive deviance.

Procedure	`SIPen`	`FlexLink`	`FlexLink (cut)`	`glmnet`	`mboost`
med(Dev(test))	81.363	83.495	83.226	81.704	81.087

TABLE 7.3: *Medians of the predictive deviance of the different procedures across the 50 random splits of the Sao Paulo air pollution data set.*

7.5.2 Bodyfat Data Set

The body fat data set has already been analyzed by Penrose et al. (1985). The response is the percentage of body fat of 252 men. The different covariates are body characteristics and given in Table 7.4. The response has been calculated from a special equation introduced by Siri (1956) using the body density determined by underwater weighting. For this data set the normal distribution is used. The predictive performance of the different procedures is measured again by the predictive deviance across 50 random splits. We split the data 50 times into a training data set with 200 observations and a test data set with 51 observations. We proceed analogously to the Sao Paulo air pollution data and present the boxplots of the predictive test deviance and plots of the selected covariates in Figure 7.6.

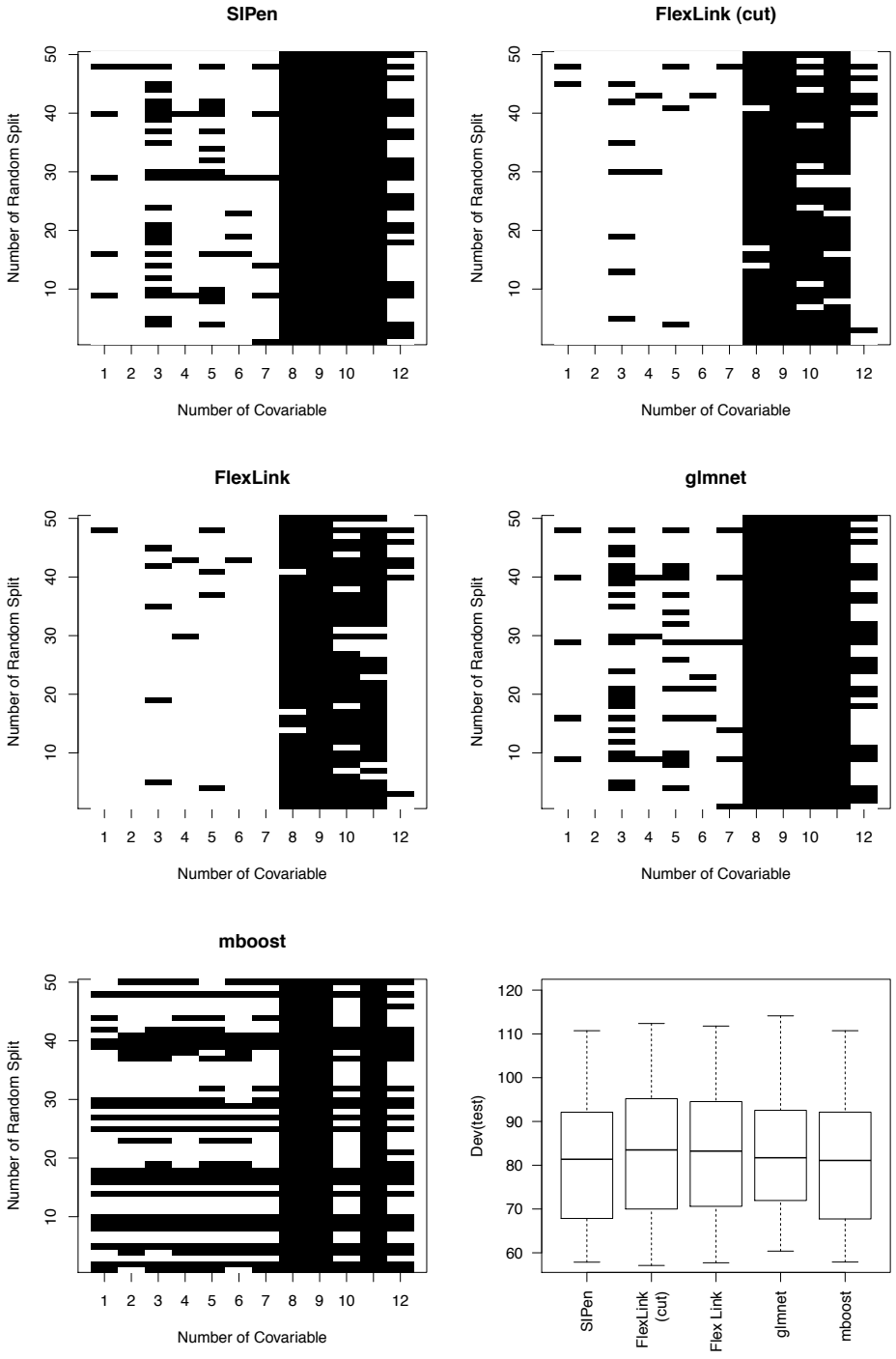

FIGURE 7.5: *Plots of selected covariates and the predictive deviance on the test data set across 50 random splits.*

Label	Explanaltion
1	age (in years)
2	weight (in lbs)
3	height (in inches)
4	neck circumference (in cm)
5	chest circumference (in cm)
6	abdomen 2 circumference (in cm)
7	hip circumference (in cm)
8	thigh circumference (in cm)
9	knee circumference (in cm)
10	ankle circumference (in cm)
11	biceps (extended) circumference (in cm)
12	forearm circumference (in cm)
13	wrist circumference (in cm)

TABLE 7.4: *Table of covariates and their number of the bodyfat data set.*

The `FlexLink` and the `FlexLink (cut)` procedures estimate the sparsest models. Only the covariates 3 (height), 4 (nech circumference), 6 (abdomen 2 circumference), and 13 (wrist circumference) are selected quite reliably. The variables 6 (abdomen 2 circumference) and 13 (wrist circumference) are selected by each procedure in each random split. The `SIPen`, `glmnet`, and the `mboost` select the covariate age in each random split. Additionally, these procedures include the covariate 12 (forearm circumference) quite often. The `mboost` selects the remaining covariates 5 and 7 to 11 only in few random splits. Covariate 9 (knee circumference) is never selected by the `mboost`. In general the penalized regression procedures `glmnet` and `SIPen` perform variable selection is not so strong as the boosting procedures for this data set. The `SIPen` outperforms the competitive procedures but gives not the sparsest models. The performance is quite stable in contrast to the more selective boosting procedures. We show the medians across the random splits in Table 7.5

Procedure	`SIPen`	`FlexLink`	`FlexLink (cut)`	`glmnet`	`mboost`
med(Dev(test))	1038.522	1160.323	1109.842	1157.516	1096.388

TABLE 7.5: *Medians of the predictive deviance of the different procedures across the 50 random splits of the bodyfat data set.*

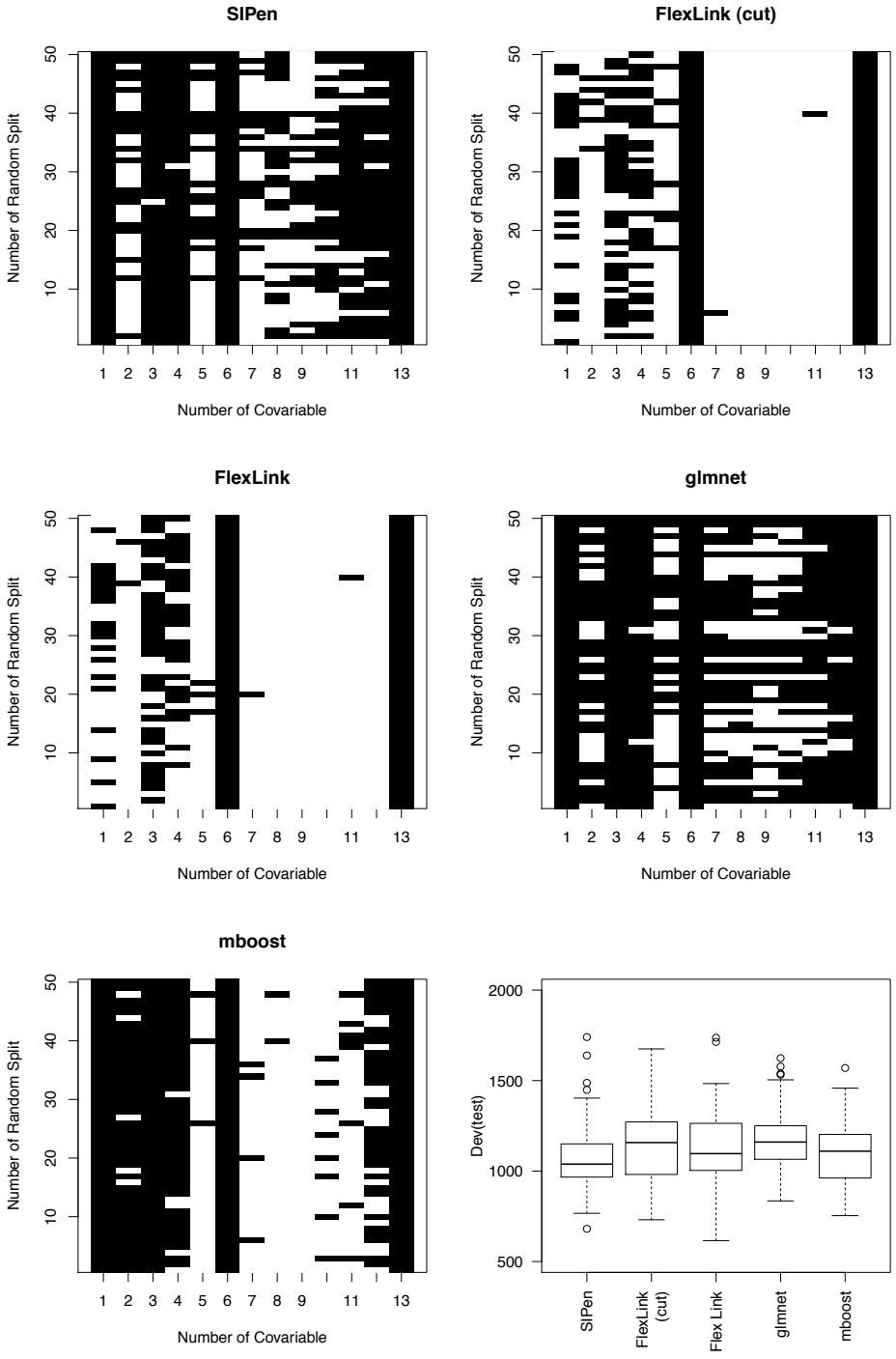

FIGURE 7.6: *Plot of selected covariates and the predictive deviance on the test data set across 50 random splits of the bodyfat data set.*

7.6 Conclusion and Remarks

A $L1$-penalization for SIMs is presented and it turns out that non-parametrical estimation of the response function improves the predictive performance. In the binomial case with a flexible modeling of the response function this improvement is quite small. All in all the presented `SIPen` is a strong competitor to the `FlexLink` and `FlexLink (cut)` which also do variable selection and estimate the response function non-parametrically. The parameter vector $\boldsymbol{\beta}$ and the response function $h(.)$ are connected by the linear predictor $\eta = \boldsymbol{x}^T\boldsymbol{\beta}$, hence $mu = h(\eta)$. So the shrinkage of the parameter vector can be partially compensated by the response function. This effect must be tackled by additional constraints. It is a challenge for the `SIPen` to achieve the same standard with respect to $\text{MSE}_{\boldsymbol{\beta}}$ as the `FlexLink` and `FlexLink (cut)` procedures. The same holds for the hits and false positive rates. The results are quite unsatisfying. In the real data examples the `SIPen` performs quite well. Especially for the body fat data the `SIPen` convinces.

By a small modification the algorithm can be used for $p > n$. In contrast to the Fisher scoring the gradient descent part of the algorithm needs no inverse. If only the gradient descent update is used the algorithm works also for the $p > n$ case.

Appendix

The parameter space is the *Euclidean space* $\mathbb{R}^p$. With $(\mathbb{R}^p)^*$ we denote the *dual Euclidean space*. $\mathbb{R}^p$ represents the vector space of all column vectors of length p with real entries. $(\mathbb{R}^p)^*$ is the vector space of all linear functions $\mathbb{R}^p \to \mathbb{R}$ which are the row vectors of length p with real entries.

Definition A 1 (Hyperplane and Linear Halfspaces) *A subset $H \subset \mathbb{R}^p$ is called* hyperplane of $\mathbb{R}^p$*, if there is a linear functional $\mathbf{c} : \mathbb{R}^p \to \mathbb{R}$, $\mathbf{c} \in (\mathbb{R}^p)^* \setminus \{\mathbf{0}\}$, and a $t \in \mathbb{R}$ for which*

$$H = \{\mathbf{x} \in \mathbb{R}^p : \mathbf{cx} = t\}$$

holds.

A subset $H^- \subset \mathbb{R}^p$ is called lower linear halfspace (of $\mathbb{R}^p$)*, if there are a linear functional $\mathbf{c} : \mathbb{R}^p \to \mathbb{R}$, $\mathbf{c} \in (\mathbb{R}^p)^*$, and $t \in \mathbb{R}$ with*

$$H^-(\mathbf{c}, t) := \{\mathbf{x} \in \mathbb{R}^p : \mathbf{cx} \leq t\}.$$

Analogously a subset $H^+ \subset \mathbb{R}^p$ is called upper linear halfspace (of $\mathbb{R}^p$)*, if there are a linear functional $\mathbf{c} : \mathbb{R}^p \to \mathbb{R}$, $\mathbf{c} \in (\mathbb{R}^p)^*$, and $t \in \mathbb{R}$ with*

$$H^+(\mathbf{c}, t) := \{\mathbf{x} \in \mathbb{R}^p : \mathbf{cx} \geq t\}.$$

Definition A 2 (Supporting Hyperplane) *Let H a hyperplane $H \subset \mathbb{R}^p$ and $K \subset \mathbb{R}^p$ a convex set. Then H is called* supporting hyperplane *or* support *for K if $H \cap K \neq \emptyset$ and K is entirely contained in one of the both closed halfspaces H^+ or H^-.*

Definition A 3 (p-crosspolytope) *The set*

$$C_p^{\triangle} := \left\{ \mathbf{x} \in \mathbb{R}^p : \sum_{i=1}^{p} |x_i| \leq 1 \right\} = conv\{\mathbf{e}_1, -\mathbf{e}_1, ..., \mathbf{e}_p, -\mathbf{e}_p\}$$

is called p-crosspolytope. $\mathbf{e}_i$, $i = 1, ..., p$, terms the i^{th} base vector of $\mathbb{R}^p$.

Definition A 4 (p-cube) *The set*

$$C_p := \{\mathbf{x} \in \mathbb{R}^p : -1 \leq x_i \leq 1\} = conv\{\{+1, -1\}^p\}$$

is called p-cube. $\{\{+1, -1\}^p\}$ terms the set of all 2^p p-dimensional vectors whose components are $+1$ or -1.

Definition A 5 (Orthant) *An orthant of the* $\mathbb{R}^p$ *is the subspace*

$$\Omega = \{\boldsymbol{\beta} \in \mathbb{R}^p : 0 \leq e_i\beta_i,\, e_i \in \{-1,\, +1\},\, \forall i = 1,\, ...,\, p,\}\,.$$

So the sign constellation within an orthant does not change and is given by the vector $\mathbf{e} = (e_i)_{i=1,\,\ldots,\,p}$.

Example A 1 *The H-representation of the LASSO penalty region in* $\mathbb{R}^3$ *is*

$$P(\mathbf{L},\, \mathbf{t}) = \begin{pmatrix} 1 & 1 & 1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & -1 \\ -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} \leq \begin{pmatrix} t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \end{pmatrix}$$

The generalization for $p \in \mathbb{N}$ *follows immediately.*

Example A 2 *For* $p = 3$, $t > 0$, *and* $c > 0$ *the H-representation of the OSCAR penalty region is:*

$$\begin{pmatrix} 2c+1 & c+1 & 1 \\ 2c+1 & 1 & c+1 \\ c+1 & 2c+1 & 1 \\ c+1 & 1 & 2c+1 \\ 1 & c+1 & 2c+1 \\ 1 & 2c+1 & c+1 \\ -(2c+1) & c+1 & 1 \\ -(2c+1) & 1 & c+1 \\ -(c+1) & 2c+1 & 1 \\ -(c+1) & 1 & 2c+1 \\ -1 & c+1 & 2c+1 \\ -1 & 2c+1 & c+1 \\ 2c+1 & -(c+1) & 1 \\ 2c+1 & -1 & c+1 \\ c+1 & -(2c+1) & 1 \\ c+1 & -1 & 2c+1 \\ 1 & -(c+1) & 2c+1 \\ 1 & -(2c+1) & c+1 \\ 2c+1 & c+1 & -1 \\ 2c+1 & 1 & -(c+1) \\ c+1 & 2c+1 & -1 \\ c+1 & 1 & -(2c+1) \\ 1 & c+1 & -(2c+1) \\ 1 & 2c+1 & -(c+1) \\ -(2c+1) & -(c+1) & 1 \\ -(2c+1) & -1 & c+1 \\ -(c+1) & -(2c+1) & 1 \\ -(c+1) & -1 & 2c+1 \\ -1 & -(c+1) & 2c+1 \\ -1 & -(2c+1) & c+1 \\ -(2c+1) & c+1 & -1 \\ -(2c+1) & 1 & -(c+1) \\ -(c+1) & 2c+1 & -1 \\ -(c+1) & 1 & -(2c+1) \\ -1 & c+1 & -(2c+1) \\ -1 & 2c+1 & -(c+1) \\ 2c+1 & -(c+1) & -1 \\ 2c+1 & -1 & -(c+1) \\ c+1 & -(2c+1) & -1 \\ c+1 & -1 & -(2c+1) \\ 1 & -(c+1) & -(2c+1) \\ 1 & -(2c+1) & -(c+1) \\ -(2c+1) & -(c+1) & -1 \\ -(2c+1) & -1 & -(c+1) \\ -(c+1) & -(2c+1) & -1 \\ -(c+1) & -1 & -(2c+1) \\ -1 & -(c+1) & -(2c+1) \\ -1 & -(2c+1) & -(c+1) \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} \leq \begin{pmatrix} t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \\ t \end{pmatrix}$$

The generalization for $p \in \mathbb{N}$ *follows immediately.*

Example A 3 *Given the H-representation of an OSCAR penalty region in* $\mathbb{R}^3$ *as in Example A 2 the set of vertices of this penalty region is:*

$$\begin{aligned}\mathcal{O} \;=\; \Bigg\{ &\begin{pmatrix} \frac{\pm t}{2c+1} \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \frac{\pm t}{2c+1} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ \frac{\pm t}{2c+1} \end{pmatrix}, \\ &\begin{pmatrix} \frac{\pm t}{3c+2} \\ \frac{\pm t}{3c+2} \\ 0 \end{pmatrix}, \begin{pmatrix} \frac{\pm t}{3c+2} \\ 0 \\ \frac{\pm t}{3c+2} \end{pmatrix}, \begin{pmatrix} 0 \\ \frac{\pm t}{3c+2} \\ \frac{\pm t}{3c+2} \end{pmatrix}, \begin{pmatrix} \frac{\pm t}{3c+2} \\ \frac{\mp t}{3c+2} \\ 0 \end{pmatrix}, \begin{pmatrix} \frac{\pm t}{3c+2} \\ 0 \\ \frac{\mp t}{3c+2} \end{pmatrix}, \begin{pmatrix} 0 \\ \frac{\pm t}{3c+2} \\ \frac{\mp t}{3c+2} \end{pmatrix}, \\ &\begin{pmatrix} \frac{\pm t}{3c+3} \\ \frac{\pm t}{3c+3} \\ \frac{\pm t}{3c+3} \end{pmatrix}, \begin{pmatrix} \frac{\mp t}{3c+3} \\ \frac{\pm t}{3c+3} \\ \frac{\pm t}{3c+3} \end{pmatrix}, \begin{pmatrix} \frac{\pm t}{3c+3} \\ \frac{\mp t}{3c+3} \\ \frac{\pm t}{3c+3} \end{pmatrix}, \begin{pmatrix} \frac{\pm t}{3c+3} \\ \frac{\pm t}{3c+3} \\ \frac{\mp t}{3c+3} \end{pmatrix} \Bigg\}. \end{aligned}$$

Proof A 1 (Propostion 1) *We consider a p-dimensional OSCAR penalty region for fixed tuning parameters* $t > 0$ *and* $c > 0$*. Let* $\mathcal{O}$ *denote the set of all vertices of this OSCAR penalty region. As remarked every row of the system of inequalities depends on the order of* $|\beta_i|$ *and one special orthant. For every facet determined by row of the system of inequalities one can find exactly p elements of* $\mathcal{O}$ *which confirm to the row by meanings of the order of* $|\beta_i|$ *and the signs. Consider the orthant with only positive values and the order* $|\beta_1| \geq |\beta_2| \geq \cdots \geq |\beta_p|$ *then only the following p vertices are elements of the corresponding row:*

$$\widetilde{\mathcal{O}} = \left\{ \begin{pmatrix} v(1) \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} v(2) \\ v(2) \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} v(3) \\ v(3) \\ v(3) \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} v(p) \\ v(p) \\ v(p) \\ v(p) \\ \vdots \\ v(p) \end{pmatrix} \right\}$$

By changing the signs and permuting the rows of the vertices of $\widetilde{\mathcal{O}}$ *we get the other orders of* $|\beta_i|$ *in every orthant.*

Hence every facet is defined by a p-elementic subset of $\mathcal{O}$ *and one row of the inequation system. The fact that no hyperplanes is ignored by a set of the kind* $\widetilde{\mathcal{O}}$ *and all elements of* $\mathcal{O}$ *are used completes the proof.*

Proof A 2 (Corollary 1) *If m of the p components of a vertex are nonzero then there are* $\binom{p}{m}$ *permutations of this m components. Further there are* 2^m *different sign combinations which are convenient. Its well known that* $\sum_{m=0}^{p} \binom{p}{m} a^{p-m} b^m = (a+b)^p$*. Now choose* $a = 1$ *and* $b = 2$*. Further* $0 < m \leq p$ *and* $\binom{p}{0} 1^p 2^0 = 1$ *holds and immediately* $\sum_{m=1}^{p} \binom{p}{m} 2^m = 3^p - 1$ *follows.*

The second statement follows directly from Proof A 1.

Example A 4 *The set of vertices the V8 penalty region is the union of the LASSO vertices* $\mathcal{L}$ *and vertices on the bisecting lines in every* β_i*-*β_j*-plane* $\mathcal{B} = \bigcup_{i<j} \mathcal{B}_{ij}$*.*

In $\mathbb{R}^4$ *the LASSO vertices are:*

$$\mathcal{L} = \left\{ \begin{pmatrix} \pm t \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \pm t \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ \pm t \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ \pm t \end{pmatrix} \right\}.$$

The remaining set $\mathcal{B} = \bigcup_{i<j} \mathcal{B}_{ij}$ *is*

$$\mathcal{B} = \left\{ \begin{pmatrix} \pm\frac{t}{2-c_{12}} \\ \pm\frac{t}{2-c_{12}} \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \pm\frac{t}{2-c_{12}} \\ \mp\frac{t}{2-c_{12}} \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \pm\frac{t}{2-c_{13}} \\ 0 \\ \pm\frac{t}{2-c_{13}} \\ 0 \end{pmatrix}, \begin{pmatrix} \pm\frac{t}{2-c_{13}} \\ 0 \\ \mp\frac{t}{2-c_{13}} \\ 0 \end{pmatrix}, \right.$$
$$\begin{pmatrix} \pm\frac{t}{2-c_{14}} \\ 0 \\ 0 \\ \pm\frac{t}{2-c_{14}} \end{pmatrix}, \begin{pmatrix} \pm\frac{t}{2-c_{14}} \\ 0 \\ 0 \\ \mp\frac{t}{2-c_{14}} \end{pmatrix}, \begin{pmatrix} 0 \\ \pm\frac{t}{2-c_{23}} \\ \pm\frac{t}{2-c_{23}} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \pm\frac{t}{2-c_{23}} \\ \mp\frac{t}{2-c_{23}} \\ 0 \end{pmatrix},$$
$$\left. \begin{pmatrix} 0 \\ \pm\frac{t}{2-c_{24}} \\ 0 \\ \pm\frac{t}{2-c_{24}} \end{pmatrix}, \begin{pmatrix} 0 \\ \pm\frac{t}{2-c_{24}} \\ 0 \\ \mp\frac{t}{2-c_{24}} \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ \pm\frac{t}{2-c_{34}} \\ \pm\frac{t}{2-c_{34}} \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ \pm\frac{t}{2-c_{34}} \\ \mp\frac{t}{2-c_{34}} \end{pmatrix} \right\}.$$

The generalization to any finite $p \in \mathbb{N}$ *follows immediately. So The V8 penalty region is* $\mathcal{P} = conv(\mathcal{L} \cup \mathcal{B})$.

Bibliography

Anbari, M. E. and A. Mkhadri (2008). Penalized regression combining the l1 norm and a correlation based penalty. *INRIA Research Report 6746*.

Antoniadis, A., G. Gregoire, and I. W. McKeague (2004). Bayesian estimation in single-index models. *Statistica Sinica 14*, 1147–1164.

Avalos, M., Y. Grandvalet, and C. Ambroise (2007). Parsimonious additive models. *Computational Statistics and Data Analysis 51*(6), 2851–2870.

Bondell, H. D. and B. J. Reich (2008). Simultaneous regression shrinkage, variable selction and clustering of predictors with oscar. *Biometrics 64*, 115–123.

Bondell, H. D. and B. J. Reich (2009). Simultaneous factor selection and collapsing levels in anova. *Biometrics 65*, 169–177.

Breiman, L. (1996). Heuristics of instability and stabilisation in model selection. *The Annals of Statistics 24*, 2350–2383.

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

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

Candes, E. and T. Tao (2007, DEC). The Dantzig selector: Statistical estimation when p is much larger than n. *Annals of Statistics 35*(6), 2313–2351.

Carroll, R. J., J. Fan, I. Gijbels, and M. P. Wand (1997). Generalized partially linear single-index models. *Journal of the American Statistical Association 92*, 477–489.

Conceicao, G. M. S., S. G. E. K. Miraglia, H. S. Kishi, P. H. N. Saldiva, and J. M. Singer (2001). Air pollution and children mortlity: a time series study in são paulo, brazil. *Environmental Health Perspectives 109*, 347–350.

Cui, X., W. K. Härdle, and L. Zhu (2009). Generalized single index models: The efm approach. Discussion Paper 50, SFB 649, Humboldt University Berlin, Economic Risk.

Czado, C. (1992). On link selection in generalized linear models. In S. L. N. in Statistics (Ed.), *Advances in GLIM and Statistical Modelling*. Springer-Verlag. 78, 60–65.

Czado, Y. and A. Munk (2000). Noncanonical links in generalized linear models - when is the effort justifed? *Journal of statistical planning and inference 87*, 317–345.

Czado, Y. and T. Santner (1992). The effect of link misspecification on binary regression inference. *Journal of statistical planning and inference 33*, 213–231.

De Boor, C. (1978). *A Practical Guide to Splines*. New York: Springer-Verlag.

Dep, P. and P. K. Trivedi (1997). Demand for medical care by the elderly: A finite mixture approach. *Journal of Applied Econometrics 12*, 313–336.

Dierckx, P. (1993). *Curve and Surface Fitting with Splines*. Oxford Science Publications.

Doppler, S. (2008). *Alters-, Aktivitäts- und Krankheitsmerkmale in der menschlichen-Knochenmikrostruktur: Eine vergleichende Studie einer individualaltersbekannten historischenPopulation mit rezenten Menschen*. Dissertation, Ludwig-Maximilians-Universität, München.

Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani (2004). Least angle regression. *The Annals of Statistics 32*, 407–499.

Eilers, P. H. C. and B. D. Marx (1996). Flexible smoothing with B-splines and Penalties. *Statistical Science 11*, 89–121.

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

Freund, Y. and R. E. Shapire (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences 55*, 119–139.

Friedman, J., T. Hastie, and R. Tibshirani (2008). *glmnet: Lasso and elastic-net regularized generalized linear models*. R package version 1.1.

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

Friedman, J. H., T. Hastie, and R. Tibshirani (2010b). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software I 33*.

Friedman, J. H. and W. Stützle (1981). Projection pursuit regression. *J. Amer. Statist. Assoc. 76*, 817–823.

Gaiffas, S. and G. Lecue (2007). Optimal rates and adaptations in the single-index model using aggregation. *Electronic Journal of Statistics 1*, 538–573.

Gawrilow, E. and M. Joswig (2000). polymake: a framework for analyzing convex polytopes. In G. Kalai and G. M. Ziegler (Eds.), *Polytopes — Combinatorics and Computation*, pp. 43–74. Birkhäuser.

Genz, A., F. Bretz, T. Miwa, X. Mi, F. Leisch, F. Scheipl, and T. Hothorn (2011). *mvtnorm: Multivariate Normal and t Distributions*. R package version 0.9-999.

Gertheiss, J., S. Hogger, C. Oberhauser, and G. Tutz (2009). Selection of ordinally scaled independent variables. Technical Report 62, Department of Statistics LMU Munich.

Gertheiss, J. and G. Tutz (2010). Sparse modeling of categorial explanatory variables. *The Annals of Applied Statistics 136*, 100–107.

Goeman, J. (2010a). L1 penalized estimation in the cox proportional hazards model. *Biometrical Journal 52*, 70–84.

Goeman, J. (2010b). *penalized: L1 (lasso) and L2 (ridge) penalized estimation in GLMs and in the Cox model*. R package version 0.9-32.

Härdle, W., P. Hall, and H. Ichimura (1993). Optimal smoothing in single-index models. *The Annals of Statistics 21*, 157–178.

Hastie, T. (2007). Comment: Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science 22*(4), 513–515.

Hastie, T. and R. Tibshirani (1990). *Generalized Additive Models*. London: Chapman & Hall.

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

Hofner, B., T. Hothorn, T. Kneib, and M. Schmid (2009). A framework for unbiased model selection based on boosting. Technical Report 72, Department of Statistics LMU Munich.

Hofner, B., T. Hothorn, T. Kneib, and M. Schmid (2011). A framework for unbiased model selection based on boosting. *Journal of Computational and Graphical Statistics (to appear) 39*(5), 1–13.

Hothorn, T., P. Buehlmann, T. Kneib, M. Schmid, and B. Hofner (2010). Model-based boosting 2.0. *Journal of Machine Learning Research 11*, 2109–2113.

Hothorn, T., P. Bühlmann, T. Kneib, M. Schmid, and B. Hofner (2009). *mboost: Model-Based Boosting*. R package version 2.0-0.

Hristache, M., A. Juditsky, and V. Spokoiny (2001). Direct estimation of the index coefficient in a single-index model. *Annals of Statistics 29*, 595–623.

James, G. M. and P. Radchenko (2008). A Generalized Dantzig selector with Shrinkage Tuning. *Biometrika* , 127–142.

James, W. and C. Stein (1961). Estimation with quadratic loss. *Proc. 4th Berkeley Symp. Math. Statist. Prob., University of California Press, Berkeley 1*, 361–380.

Klein, R. L. and R. H. Spady (1993). An efficient semiparametric estimator for binary response models. *Econometrica 61*, 387–421.

Leitenstorfer, F. and G. Tutz (2007). Generalized monotonic regression based on b-splines with an application to air pollution data. *Biostatistics 8*, 654–673.

Lokhorst, J., B. Venables, B. Turlach, and M. Maechler (2007). *lasso2: L1 constrained estimation aka 'lasso'*. R package version 1.2-6.

Luenberger, D. G. (1969). *Optimazition by Vector Space Methods*. New York: John Wiley & Sons.

Maron, M. (2007). Threshold effect of eucalypt density on an aggressive avian competitor. *Biological Conservation 136*, 100–107.

McCullagh, P. and J. A. Nelder (1983). *Generalized Linear Models*. New York: Chapman & Hall.

Muggeo, V. M. R. and G. Ferrara (2008). Fitting generalized linear models withunspecified link function: A p-spline approach. *Computational Statistics & DataAnalysis 52*(5), 2529–2537.

Naik, P. A. and C.-L. Tsai (2001). Single-index model selection. *Biometrika Trust 88*, 821–832.

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

Osborne, M., B. Presnell, and B. Turlach (2000). On the lasso and its dual. *Journal of Computational and Graphical Statistics*, 319–337.

Park, M. Y. and T. Hastie (2007a). *glmpath: L1 Regularization Path for Generalized Linear Models and Cox Proportional Hazards Model*. R package version 0.94.

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

Penrose, K. W., A. G. Nelson, and A. G. Fisher (1985). Generalized body composition prediction equation for men using simple measurement techniques. *Medicine and Science in Sports and Exercise 17*, 189.

Petry, S., C. Flexeder, and G. Tutz (2010). Pairwise fused lasso. Technical Report 102, Department of Statistics LMU Munich.

Petry, S. and G. Tutz (2011a). The oscar for generalized linear models. Technical Report 112, Department of Statistics LMU Munich.

Petry, S. and G. Tutz (2011b). Shrinkage and variable selection by polytopes. *Journal of Statistical Planning and Inference (to appear)*.

Powell, J. L., J. H. Stock, and T. M. Stoker (1989). Semiparametric estimation of index coefficients. *Econometrica 57*, 1403–1430.

R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Ramsay, J. O., H. Wickham, S. Graves, and G. Hooker (2010). *fda: Functional Data Analysis*. R package version 2.2.5.

Ramsey, J. O. and B. W. Silverman (2005). *Functional Data Analysis*. New York: Springer.

Ruckstuhl, A. and A. Welsh (1999). Reference bands for nonparametrically estimatedlink functions. *Journal of Computational and Graphical Statistics 8*(4), 699–714.

Schäfer, J., R. Opgen-Rhein, and K. Strimmer (2009). *Efficient estimation of covariance and (partial) correlation*. R package version 1.5.3.

Schmid, M. and T. Hothorn (2008). Boosting additive models using component-wise p-splines. *Computational Statistics & Data Analysis 53*(2), 298–311.

Shapire, R. E. (1990). The strength of weak learnability. *Machine Learning 5*, 197–227.

Simon, N., J. Friedman, T. Hastie, and R. Tibshirani (2011). Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software 39*(5), 1–13.

Siri, W. B. (1956). *Advances in Biological and Medical Physics*, Volume 4, Chapter The gross composition of the body, pp. 239–280. Academic Press New York.

Sjöstrand, K. (2005, jun). Matlab implementation of LASSO, LARS, the elastic net and SPCA. Version 2.0.

Stoker, T. M. (1986). Consistent estimation of scaled coefficients. *Econometrica 54*, 1461–1481.

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

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

Turlach, B. A. (2009). *quadprog: Functions to solve Quadratic ProgrammingProblems*. R package version 1.4-11, S original by Berwin A. Turlach, R portby Andreas Weingessel.

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

Tutz, G. and F. Leitenstorfer (2011). Estimation of single-index models based on boosting techniques. *Statistical Modelling, to appear*, 183–197.

Tutz, G. and S. Petry (2011). Nonparametric estimation of the link function including variable selection. *Statistics and Computing (to appear) 21*.

Tutz, G. and J. Ulbricht (2009). Penalized regression with correlation based penalty. *Statistics and Computing 19*, 239–253.

Ulbricht, J. (2010a). *lqa: Local quadratic approximation*. R package version 1.0-2.

Ulbricht, J. (2010b). *Variable selection in generalized linear models*. Dissertation, Ludwig-Maximilians-Universität, München.

Venables, W. N. and B. D. Ripley (2002). *Modern Applied Statistics with S. Fourth edition*. Springer.

Weisberg, S. (2005). *Applied Linear Regression* (Third ed.). Hoboken NJ: Wiley.

Weisberg, S. (2011). *alr3: Data to accompany Applied Linear Regression 3rd edition*. R package version 2.0.3.

Weisberg, S. and A. H. Welsh (1994). Adapting for the missing link. *Annals of Statistics 22*, 1674–1700.

Whittaker, J. (1990). *Graphical Models in Applied Multivariate Statistics*. Chichester: Wiley.

Wood, S. (2011). *mgcv: GAMs with GCV/AIC/REML smoothness estimation and GAMMs by PQL*. R package version 1.7-2.

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

Yu, Y. and D. Ruppert (2002). Penalized spline estimation for partially linear single-index models. *Journal of the American Statistical Association 97*, 1042–1054.

Zeileis, A. (2006). Object-oriented computation of sandwich estimator. *Journal of Statistical Software 16*(9).

Ziegler, G. M. (1994). *Lectures on Polytopes*. New York, Berlin, Heidelberg, London, Paris Tokyo, Hong Kong, Barcelona, Budapest: Springer Verlag (in Graduates Texts in Mathematics).

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association 101*(476), 1418–1429.

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