Adaptive Subspace Methods for High-Dimensional Variable Selection

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen University zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Christian Staerk, M.Sc.

aus Aachen

Berichter: Universitätsprofessorin Dr. Maria Kateri
Professor Dr. Ioannis Ntzoufras
Universitätsprofessor Dr. Erhard Cramer

Tag der mündlichen Prüfung: 26. April 2018

Acknowledgements

First and foremost, I would like to thank my supervisor Professor Maria Kateri, who always supported me throughout my whole PhD. In particular, I want to thank her for her guidance as well as for the opportunity to pursue my research ideas. I am very grateful that she believed in my work even during challenging periods of research.

I want to thank my second supervisor Professor Ioannis Ntzoufras for the many valuable and fruitful discussions, particularly during my four research stays in Athens. I really enjoyed the open and supportive atmosphere in Athens and I miss these times a lot. Furthermore, I am very glad and honoured that he travelled to Aachen for my PhD defence.

I thank Professor Erhard Cramer for agreeing to be the third referee for my work. This is also the right place to thank him for his inspiring lectures and his support during my course of study at RWTH Aachen University, which initiated my interest in the field of statistics.

A great thanks goes to my colleagues from the Institute of Statistics! They all have been very supportive, especially through the numerous non-work related activities. In particular, I want to thank my former and current office mates, Panagiotis Bompotas and Anastasia Gaponik, for the great working atmosphere and for always having an open ear for any kinds of problems.

My deepest gratitude goes to my family and friends, who support me unconditionally. I want to thank my parents for everything. They have always been there for me, providing invaluable advice when needed and they have always been interested in my (often quite technical) work. A discussion with my father resulted in the football analogy which is included in this thesis as a motivation for the AdaSub method. I am so grateful and happy that my love Mareike stood by my side during the final stressful phase of my PhD. She cheered me up when I was struggling with my research and encouraged me to continue with new energy the next day. Without the support of all of my family, the completion of this thesis would not have been possible.

Contents

1. Introduction

> *Far better an approximate answer to the right question, which is often vague, than an exact answer to the wrong question, which can always be made precise.*
>
> — John Tukey, 1962

Rapid developments during the last decades in fields such as information technology and genomics have led to an enormous increase in the collection, storage and processing of large datasets. The new ability to collect huge amounts of data has called for a paradigm shift in the field of statistics: While classical statistics typically focused on scenarios with moderate sample sizes and a small number of variables (i.e. measurements per subject), nowadays one often faces huge sample sizes and/or a huge number of variables.

In this thesis we want to deal with the challenging scenario studied by the relatively new field named high-dimensional statistics, where many explanatory variables p are observed while the sample size n is relatively small. In this high-dimensional setting with p possibly much larger than n, we are particularly interested in variable selection, meaning that we want to find a sparse model including only a few of the numerous explanatory variables that fits and ideally explains the observed data well.

Typical areas of application for high-dimensional variable selection methods include genomics, chemometrics and text categorization. For example in genomics it is nowadays possible to "measure" thousands of genes, say $p = 10,000$ gene expression measurements per patient, while typically there is only a very limited number of patients available in a study, say $n = 100$. If we know that some of the patients have a certain disease, while the others do not, we might be interested in identifying those (often few) genes which are associated to that particular disease.

Although the variable selection problem occurs in all kinds of different statistical models and the developed variable selection methods are often more general, in this thesis we want

to focus on variable selection in generalized linear models (GLMs) with a univariate response variable. In a GLM the conditional expectation of the response variable given the explanatory variables is linked to a transformed affine combination of the explanatory variables. While many different methods have been proposed to solve the variable selection problem in a GLM setup, there are on the first sight two fundamentally different ways to attack this problem: the classical approach and the fully Bayesian approach. In this thesis, the first **Chapters 2–6** mainly focus on the classical approach, while **Chapter 7** addresses variable selection in a Bayesian framework.

1.1. Classical variable selection

A classical approach to the variable selection problem is to come up with a suitable selection criterion and to solve the resulting optimization problem. Selection criteria include, among many others, the Akaike Information Criterion (AIC, Akaike, 1974), the Bayesian Information Criterion (BIC, Schwarz, 1978) and the recently proposed Extended Bayesian Information Criterion (EBIC, Chen and Chen, 2008), which particularly aims at high-dimensional settings. The challenging problem with these ℓ_0-type selection criteria is that the resulting combinatorial optimization problems are very difficult to solve if there are many possible explanatory variables p, since there are generally 2^p possible models for which the criterion has to be evaluated. It has been shown that best subset selection in linear regression models with an ℓ_0-penalty is in general NP-hard (see e.g. Huo and Ni, 2007).

In the late 90's the focus shifted from solving discrete optimization problems to solving continuous, convex relaxations of the original problem. Tibshirani (1996) proposes the celebrated Lasso ("Least Absolute Shrinkage and Selection Operator"), which solves a convex optimization problem with an ℓ_1-penalty on the regression coefficients and then selects those variables whose corresponding regression coefficients are non-zero in the optimal solution. A drawback of ℓ_1-regularization methods like the Lasso is that they typically require quite strong conditions on the correlation structure between the explanatory variables in order to yield consistent variable selection (see e.g. Zhao and Yu, 2006 and Meinshausen and Bühlmann, 2006).

In **Chapter 2** of this thesis we provide a selective review of the properties of ℓ_0- and ℓ_1-type methods for high-dimensional variable selection. Furthermore, we briefly discuss

several other popular approaches to the variable selection problem, including non-convex regularization methods like the SCAD (Fan and Li, 2001), screening methods like Sure Independence Screening (SIS, Fan and Lv, 2008) and resampling methods like Stability Selection (Meinshausen and Bühlmann, 2010). While much recent work has focused on solving computationally efficient relaxations of the generally NP-hard ℓ_0-type problem, another quite obvious approach would be to directly attack the original ℓ_0-type problem by using approximate algorithms. Such algorithms might not be guaranteed to identify the exact solution to the optimization problem in all possible scenarios, but they have the potential to provide reasonable approximate solutions with good statistical properties. In fact, one of the main motives of the work presented in this thesis is the desire to solve optimization problems induced by ℓ_0-type selection criteria, which provide variable selection consistency under weaker conditions than their convex ℓ_1-type counterparts.

For this purpose, in **Chapter 3** we propose an Adaptive Subspace (AdaSub) method for high-dimensional variable selection which aims at identifying the best model with respect to a certain selection criterion (including ℓ_0-type criteria like the EBIC). AdaSub is a stochastic algorithm which is fundamentally based on the idea of adaptively solving several low-dimensional sub-problems of the original high-dimensional problem. Here, the low-dimensional problems to be solved consist of relatively small subspaces of all possible explanatory variables and are constructed (or more precisely sampled) in an adaptive way, so that, along the iterations of the algorithm, "important" explanatory variables are included in the sampled subspaces with increasing probability, while "irrelevant" explanatory variables are considered with decreasing probability. In the ideal situation, AdaSub "converges correctly" against the subspace which corresponds to the best model according to the considered selection criterion, meaning that the sampling probabilities of variables included in the best model converge to one, while the sampling probabilities of variables not included in the best model converge to zero.

In **Chapter 4** we theoretically investigate such limiting properties of AdaSub. In particular, we provide a sufficient condition (the so-called ordered importance property, OIP) for the "correct convergence" of AdaSub in the sense described above. Furthermore, we analyse the case in which AdaSub does not converge against the best model according to the considered criterion and we argue that in such a situation AdaSub can improve the "stability" of the

finally selected model, in the sense that the AdaSub model contains less "noise" variables (false positive selections) than the criterion optimal model. We also provide conditions under which the variable selection consistency of the models selected by AdaSub can be guaranteed.

In **Chapter 5** we consider different simulated and real data examples in the framework of normal linear models. In low-dimensional simulated examples we compare the models selected by AdaSub with the best model according to the used criterion; in particular, we empirically demonstrate that AdaSub can indeed provide more "stable" models (with less false positive selections), in situations where the employed criterion would yield a model that "overfits" the data. Furthermore, in various high-dimensional simulated examples we show that AdaSub performs very competitive in comparison to other popular variable selection methods such as Forward Stepwise Selection, the Lasso (Tibshirani, 1996), the SCAD (Fan and Li, 2001), the Adaptive Lasso (Zou, 2006) and Stability Selection (Meinshausen and Bühlmann, 2010). Finally, we illustrate the effectiveness of AdaSub via high-dimensional real data examples from the field of genomics.

The "vanilla" AdaSub method is based on the idea of solving the sampled sub-problems exactly by making use of a full model enumeration or of clever branch-and-bound algorithms (see e.g. Narendra and Fukunaga, 1977). However, for GLMs different than the normal linear model the evaluation of the criterion values for single models might already be quite computationally expensive since the respective maximum likelihood estimators of the regression coefficients are generally not available in closed analytic form. This motivates the introduction of certain modifications of AdaSub which make use of heuristic optimization methods for solving the low-dimensional sub-problems.

In **Chapter 6** we introduce two variants of AdaSub based on Forward and Backward Stepwise Selection, respectively. It turns out that the version based on Backward Stepwise Selection (called BackAdaSub) shows favourable properties in comparison to the version based on Forward Stepwise Selection (called FoAdaSub), both theoretically and empirically. In particular, we will argue that BackAdaSub can be viewed as a "good approximate algorithm" to the original AdaSub method. The effectiveness of BackAdaSub is demonstrated through simulated and real data examples in the framework of logistic regression models.

1.2. Bayesian variable selection

In a Bayesian approach to the variable selection problem one assigns prior probabilities to each of the considered models along with prior distributions for the respective model parameters. From the resulting posterior model distribution one can obtain posterior marginal inclusion probabilities as measures of the "importance" of the different explanatory variables. Furthermore, the median probability model (Barbieri and Berger, 2004) or Bayesian model averaging (Raftery et al., 1997) may be used for predictive inference (by appropriately accounting for model uncertainty).

Note that important ℓ_0-type criteria like the BIC or the EBIC can be viewed as asymptotic approximations to a fully Bayesian approach (compare e.g. Liang et al., 2013). Furthermore, the challenging problem in the Bayesian variable selection framework is basically the same as with ℓ_0-type information criteria: Computing (approximate) posterior model probabilities for all possible models is not feasible if the number of explanatory variables p is very large, since there are in general 2^p possible models which have to be considered. Although the fundamental problem is similar, the classical and the Bayesian focus is different: Classical methods aim at identifying the best model according to the employed criterion, i.e. they focus on optimization, while Bayesian methods intrinsically aim at deriving (an approximation to) the full posterior model distribution, i.e. they focus on sampling from the posterior distribution. Often, Markov Chain Monte Carlo (MCMC) methods based on Metropolis-Hastings steps (e.g. Madigan et al., 1995), Gibbs samplers (e.g. George and McCulloch, 1993) and "reversible jump" updates (e.g. Green, 1995) are used in order to sample from the posterior model distribution. However, the effectiveness of such MCMC methods depends heavily on a sensible choice of the proposal distributions being used.

It turns out that the basic idea which underlies the proposed AdaSub method can also be useful in the Bayesian setting for the construction of efficient algorithms that sample from posterior model distributions: If we have already sampled a number of models by using a certain MCMC algorithm, we may want to utilize the currently available information to construct better and better model proposals, so that the MCMC algorithm is able to explore the relevant part of the model space more quickly (leading to a faster mixing of the chain). In particular, if a certain explanatory variable is included in many of the already sampled models, this indicates that this variable is "important" (i.e. it has large posterior

marginal inclusion probability), so that we should propose to include this variable with large probability in the following iterations of the algorithm, as well. The general idea of updating the proposal distributions "on the fly" during a single run of the algorithm leads to adaptive MCMC algorithms (see e.g. Nott and Kohn, 2005 and Roberts and Rosenthal, 2007).

In **Chapter 7** we propose an adaptive MCMC algorithm — the Metropolized Adaptive Subspace (MAdaSub) method — for Bayesian variable selection. MAdaSub is based on an independent Metropolis-Hastings sampler, where the sampling probabilities of the explanatory variables are sequentially adapted after each iteration according to an updating scheme inspired by AdaSub. We show that the individual sampling probabilities of the variables finally converge against the true respective posterior marginal inclusion probabilities and that the proposed algorithm is ergodic despite its continuing adaptation, i.e. "in the limit" the algorithm samples correctly from the targeted posterior model distribution. Through simulated and real data examples we demonstrate that the algorithm provides an efficient and stable way for sampling from very high-dimensional and multimodal posterior model distributions.

In the concluding **Chapter 8** we summarize the main results presented in this thesis and discuss directions for future research.

2. Selective overview of classical variable selection methods

In this chapter we present a selective overview of variable selection methods in the framework of high-dimensional generalized linear models, where the focus will be primarily on classical and not on Bayesian methods (although we do provide Bayesian motivations for the methods when they give important further insights). The main purpose of this chapter is to provide a unified, compact and up-to-date summary of different methods along with their key properties. We refer to Guyon and Elisseeff (2003), Saeys et al. (2007), O'Hara and Sillanpää (2009), Fan and Lv (2010) and Mallick and Yi (2013) for comprehensive alternative overviews discussing high-dimensional variable selection from different points of view (note that in the Machine Learning community one tends to use the term "feature selection" in place of "variable selection").

In Section 2.1 we introduce the generalized linear model setting and the corresponding variable selection problem. While many recent articles and textbooks (like Bühlmann and van de Geer, 2011 and Hastie et al., 2015) mainly focus on convex ℓ_1-type regularization methods, in Section 2.2 we begin the discussion with ℓ_0-type information criteria which enforce sparse solutions in an arguably more "natural" way. Furthermore, in Section 2.3 we review different methods for solving the associated discrete optimization problems. In Section 2.4 we provide an overview of the methodology regarding convex ℓ_1-type relaxations of the original ℓ_0-type problem, including the Lasso and its many variants.

During the discussion, we exemplarily state important theoretical results concerning the consistency of ℓ_0-type (Section 2.2) and ℓ_1-type methods (Section 2.4.2). These results are obtained in certainly idealized theoretical frameworks, but they provide crucial insights concerning the different performances when applied on real datasets. They also give motivation and guidance for developing new variable selection methods that will be introduced and

discussed in the following chapters of this thesis.

In Section 2.5 we give a brief glimpse of the methodology regarding nonconvex regularization methods like the SCAD, while in Section 2.6 we discuss computationally efficient screening methods for ultra-high-dimensional scenarios where the number of possible explanatory variables largely exceeds the sample size. In Section 2.7 we conclude our discussion with resampling methods like Stability Selection, which aim at enhancing the stability of the models selected by different regularization methods.

2.1. The variable selection problem in a GLM

We briefly describe the variable selection problem in a *generalized linear model (GLM)*. In doing so, we also introduce relevant notation for the remainder of this thesis. We refer to McCullagh and Nelder (1989), Hardin and Hilbe (2007), Kateri (2014) and Agresti (2015) for comprehensive treatments of generalized linear models.

2.1.1. Generalized linear models (GLMs)

Notation 2.1. Let $X_1, \ldots, X_p$ denote *explanatory variables* (also called *predictors* or *covariates*) and let Y denote the *response variable*. Suppose that we observe some data of sample size n; note that in the high-dimensional setting it may be the case that the number of explanatory variables p is much larger than n (abbreviated by $p \gg n$).

Let $\boldsymbol{X} = (X_{i,j}) \in \mathbb{R}^{n \times p}$, $i = 1, \ldots, n,\ j = 1, \ldots, p$ denote the matrix of observed explanatory variables where it's i-th row $\boldsymbol{X}_{i,*}$ corresponds to the i-th observation and it's j-th column $\boldsymbol{X}_{*,j} \equiv \boldsymbol{X}_j$ corresponds to the values of the j-th predictor. For $I \subseteq \{1, \ldots, n\}$, $\boldsymbol{X}_{I,*} \in \mathbb{R}^{|I| \times p}$ denotes the matrix $\boldsymbol{X}$ restricted to the rows with indices in I and for $J \subseteq \{1, \ldots, p\}$, $\boldsymbol{X}_{*,J} \equiv \boldsymbol{X}_J \in \mathbb{R}^{n \times |J|}$ denotes the matrix $\boldsymbol{X}$ restricted to the columns with indices in J. Often, the matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ is extended to the *design matrix* $\tilde{\boldsymbol{X}} \in \mathbb{R}^{n \times (p+1)}$, where the first column of $\tilde{\boldsymbol{X}}$ is given by $\tilde{\boldsymbol{X}}_1 = (1, \ldots, 1)^T \in \mathbb{R}^n$ (in order to implicitly include an intercept in the model) and $\tilde{\boldsymbol{X}}_{\{2,\ldots,p\}} = \boldsymbol{X} \in \mathbb{R}^{n \times p}$. For simplicity, in the following both $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}}$ are referred to as design matrices, depending on the context. The response vector (as a random vector with values in $\mathbb{R}^n$) is denoted by $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^T$, while the observed response vector is denoted by $\boldsymbol{y} = (y_1, \ldots, y_n)^T \in \mathbb{R}^n$. Let furthermore $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y}) \in \mathfrak{D}_{n,p}$ summarize all the available data, where $\mathfrak{D}_{n,p}$ denotes an appropriate sample space.

Before we discuss generalized linear models (GLMs), we recall the definition of an exponential dispersion family, which is the fundamental building block for the stochastic component of a GLM. Details concerning exponential families and in particular exponential dispersion families can for example be found in Barndorff-Nielsen (2014) and Jørgensen (1987).

Definition 2.2. A family of (continuous or discrete) distributions $\mathscr{P} = \{P_{\theta,\psi};\ \theta \in \mathbb{R}, \psi > 0\}$ is a (univariate) *exponential dispersion family* if its density functions (or probability mass functions) can be written in the form

$$f(y;\, \theta, \psi) = \exp\left\{ \frac{y\theta - b(\theta)}{a(\psi)} + c(y, \psi) \right\},\ y \in S\,, \tag{2.1}$$

where $\theta \in \mathbb{R}$ and $\psi > 0$ are two parameters, a, b, c are given real functions with $a(\psi) > 0$ for all $\psi > 0$ and $S \subseteq \mathbb{R}$ is the support for all distributions in $\mathscr{P}$ (thus S does not depend on θ or ψ). The parameter θ is called the *natural parameter* or *canonical parameter* and ψ the *dispersion parameter* of the family. The parameter ψ might be known or unknown; if ψ is known, then $\mathscr{P}$ is a *one-parametric exponential family*.

Examples 2.1. (a) The family of univariate normal distributions $\{N(\theta, \sigma^2);\ \theta \in \mathbb{R}, \sigma^2 > 0\}$ forms an exponential dispersion family, where the mean θ is the natural parameter and the variance σ^2 is the dispersion parameter.

(b) The family of Bernoulli distributions $\{\text{Bernoulli}(\pi);\ \pi \in (0,1)\}$ with success probability π forms a one-parametric exponential family, where $\log\left(\frac{\pi}{1-\pi}\right)$ is the natural parameter (the so-called "log-odds").

(c) The family of Poisson distributions $\{\mathcal{P}o(\lambda);\ \lambda > 0\}$ with mean λ forms a one-parametric exponential family, where $\log(\lambda)$ is the natural parameter.

Definition 2.3 (Nelder and Wedderburn, 1972)**.** A *generalized linear model (GLM)* models the relationship between a response variable Y and explanatory variables $X_1, \ldots, X_p$ in the following way: The components of the response vector $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^T$ are assumed to be conditionally independent, given $\boldsymbol{X}_{1,*}, \ldots, \boldsymbol{X}_{n,*}$, with each component Y_i having a distribution from a fixed exponential dispersion family with

$$g(E(Y_i \,|\, \boldsymbol{X}_{i,*})) = \mu + \sum_{j=1}^{p} \beta_j X_{i,j} = f_{\mu,\beta}(\boldsymbol{X}_{i,*}), \quad i = 1, \ldots, n\,, \tag{2.2}$$

where g is a monotonic and differentiable *link function*, $\mu \in \mathbb{R}$ is the *intercept*, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T \in \mathbb{R}^p$ is the vector of regression coefficients and $f_{\mu,\boldsymbol{\beta}}(\boldsymbol{X}_{i,*})$ is the *linear predictor* for observation $\boldsymbol{X}_{i,*}$. If the link function g is chosen such that, for $i = 1, \ldots, n$, we have $g(E(Y_i \,|\, \boldsymbol{X}_{i,*})) = \theta_i$, where θ_i is the natural parameter of the exponential dispersion family corresponding to observation Y_i, then g is called a *canonical link function*.

Notation 2.4. Sometimes it is more convenient to combine the intercept μ and the vector of regression coefficients $\boldsymbol{\beta} \in \mathbb{R}^p$ into a single vector $\tilde{\boldsymbol{\beta}} = (\mu, \beta_1, \ldots, \beta_p)^T \in \mathbb{R}^{p+1}$. With Notation 2.1, equation (2.2) reads as

$$g(E(Y_i \,|\, \boldsymbol{X}_{i,*})) = (\tilde{\boldsymbol{X}}\tilde{\boldsymbol{\beta}})_i\,, \quad i = 1, \ldots, n\,. \tag{2.3}$$

Remark 2.1. Depending on the context, the explanatory variables can be treated as either fixed (*fixed design*) or random (*random design*). In the random design case a GLM is defined as in Definition 2.3 with the additional assumption that $(\boldsymbol{X}_{1,*}, Y_1), \ldots, (\boldsymbol{X}_{n,*}, Y_n)$ are independent identically distributed (i.i.d.). Methodologically, there is no difference between the random and fixed design setting if one assumes that the true model is of the form (2.2), since in that situation one can simply condition on the observed values of the explanatory variables (compare Bühlmann and van de Geer, 2011, Chapter 2.2). However, in the situation of a misspecified model the distinction between a fixed or random design is important and the theoretical analysis of the random design case is more involved (see e.g. Bühlmann et al., 2015).

Many popular models fall under the framework of GLMs, such as the normal linear regression model, the logistic regression model and the Poisson regression model. We briefly discuss these three important models, since in this thesis they will serve as main examples for the illustration of the methodology and the corresponding algorithms. In particular, special attention is paid to the linear regression model, which is, due to it's simplicity, probably the most used and certainly the most theoretically analysed statistical model in the high-dimensional setting.

Examples 2.2. (a) *Linear Regression Model*

In a linear regression model the conditional expectations of the components of a continuous response $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^T$ are modelled as affine combinations of the explanatory

variables $X_1, \dots, X_p$:

$$E(Y_i \,|\, \boldsymbol{X}_{i,*}) = \mu + \sum_{j=1}^{p} \beta_j X_{i,j}, \quad i = 1, \dots, n\,. \tag{2.4}$$

A usual additional assumption (though arguably a quite strong assumption, compare e.g. Wasserman, 2014) is that $Y_1, \dots, Y_n$ are conditionally independent, given $\boldsymbol{X}_{1,*}, \dots, \boldsymbol{X}_{n,*}$, and normally distributed with the same variance $\sigma^2 > 0$ (which can be known or unknown). Then clearly, this model, which we call the *normal linear model*, is part of the GLM family and the identity function on $\mathbb{R}$ is the canonical link function (compare Examples 2.1 (a)). With this additional assumption the model can equivalently be expressed as

$$Y_i = \mu + \sum_{j=1}^{p} \beta_j X_{i,j} + \epsilon_i, \quad i = 1, \dots, n\,, \tag{2.5}$$

where $\epsilon_1, \dots, \epsilon_n$ are i.i.d. with $\epsilon_i \sim N(0, \sigma^2)$ for $i = 1, \dots, n$; in the random design setting we additionally assume that the errors $\{\epsilon_i;\ i = 1, \dots, n\}$ are independent of the explanatory variables $\{\boldsymbol{X}_{i,*};\ i = 1, \dots, n\}$.

If we let $\boldsymbol{\epsilon} = (\epsilon_1, \dots, \epsilon_n)^T$, then we can write (2.5) in the common matrix notation form:

$$\boldsymbol{Y} = \tilde{\boldsymbol{X}}\tilde{\boldsymbol{\beta}} + \boldsymbol{\epsilon}\,. \tag{2.6}$$

(b) *Logistic Regression Model*

Suppose that we face a binary response variable with values in $\{0, 1\}$. Then, in a logistic regression model the components of $\boldsymbol{Y} = (Y_1, \dots, Y_n)^T$ are assumed to be conditionally independent Bernoulli random variables $Y_i \,|\, \boldsymbol{X}_{i,*} \sim \text{Bernoulli}(\pi_i)$ with

$$g(E(Y_i \,|\, \boldsymbol{X}_{i,*})) = \log\left(\frac{\pi_i}{1-\pi_i}\right) = \mu + \sum_{j=1}^{p} \beta_j X_{i,j}, \quad i = 1, \dots, n\,, \tag{2.7}$$

where the canonical link function $g : (0, 1) \to \mathbb{R}$ is the logistic link, i.e. $g(z) = \log\left(\frac{z}{1-z}\right)$ for $z \in (0, 1)$ (compare Examples 2.1 (b)).

(c) *Poisson Regression Model*

Suppose that we face a response variable which represents "counts", i.e. it takes values in $\mathbb{N}_0 = \{0, 1, 2, \dots\}$. Then, in a Poisson regression model the components of $\boldsymbol{Y} = (Y_1, \dots, Y_n)^T$ are assumed to be conditionally independent Poisson random variables

$Y_i \,|\, \boldsymbol{X}_{i,*} \sim \mathcal{P}o(\lambda_i)$ with

$$g(E(Y_i \,|\, \boldsymbol{X}_{i,*})) = \log(\lambda_i) = \mu + \sum_{j=1}^{p} \beta_j X_{i,j}, \quad i = 1, \ldots, n\,, \tag{2.8}$$

where the canonical link function $g : (0, \infty) \to \mathbb{R}$ is the log-link, i.e. $g(z) = \log(z)$ for $z \in (0, \infty)$ (compare Examples 2.1 (c)).

2.1.2. The variable selection problem

If high-dimensional data is observed with many possible explanatory variables $X_1, \ldots, X_p$, it is often reasonable to assume that only a small fraction of these variables contribute substantially to the response variable Y. Formally, this can be modelled by assuming that many components of the vector of regression coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T \in \mathbb{R}^p$ are equal to zero, with the interpretation that a particular variable X_j is not "relevant" if and only if $\beta_j = 0$. It is now desirable to identify the "relevant" explanatory variables, i.e. the spots of non-zero regression coefficients.

The Variable Selection Problem: Suppose that $\boldsymbol{\beta}_0 = (\beta_{0,1}, \ldots, \beta_{0,p})^T \in \mathbb{R}^p$ is the vector of true model parameters in a GLM given by equation (2.2) and that

$$S_0 = \{j \in \{1, \ldots, p\};\ \beta_{0,j} \neq 0\} \tag{2.9}$$

is the corresponding *active set*. Then the *variable selection problem* is to identify the true active set S_0, i.e. a variable selection procedure tries to "estimate" S_0 by some subset $\hat{S} \subseteq \{1, \ldots, p\}$. A usual assumption in the high-dimensional regime is that in fact only a few components of $\boldsymbol{\beta}_0$ are non-zero, so that the size of the active set is relatively small, i.e. $s_0 = |S_0| \ll p$. This property is referred to as *sparsity* (compare e.g. Bühlmann and van de Geer, 2011, Section 1.2).

Notation 2.5. Consider a particular GLM of the form (2.2) with explanatory variables $X_1, \ldots, X_p$. For a subset $S \subseteq \{1, \ldots, p\}$, the model induced by S is defined by the same GLM but with design matrix $\boldsymbol{X}_S \in \mathbb{R}^{n \times |S|}$ in place of $\boldsymbol{X} \in \mathbb{R}^{n \times p}$, i.e. the original design matrix is restricted to the columns with indices in S. For brevity, we often simply refer to the model S. Without further notice, we assume that we always include an intercept μ in the corresponding GLM with design matrix $\boldsymbol{X}_S$. We denote the set of labelled explanatory

variables by $\mathcal{P} = \{1, \dots, p\}$ and the full model space by

$$\mathcal{M}_{\text{full}} = \{S;\, S \subseteq \mathcal{P}\}\,. \tag{2.10}$$

Since we have $|\mathcal{M}_{\text{full}}| = 2^p$, the number of possible models is growing exponentially with the number of possible predictors p. This is one of the main reasons why the variable selection problem turns out to be very hard, especially in high-dimensional settings where the number of possible explanatory variables p is very large.

Note that the assumption that the true underlying model is of the form (2.2) and, furthermore, that this model is sparse cannot be expected to hold in practice (compare e.g. the critical position of Wasserman, 2014). Nevertheless, it is desirable to identify the "best" linear (where linearity should be interpreted in the sense of (2.2)), sparse approximation to the "truth" in order to find an interpretable model that avoids overfitting (see e.g. van de Geer et al., 2011 and Bühlmann et al., 2015). So regardless of whether the assumption of a sparse linear true model holds, it is of great interest to develop and analyse variable selection procedures, which we formally define next.

Definition 2.6. A *(high-dimensional) variable selection procedure* $E = \{E_{n,p}\}$ is a family of (measurable) functions $E_{n,p} : \mathfrak{D}_{n,p} \to \mathcal{M}_{\text{full}}$ which — for each sample size $n \in \mathbb{N}$ and number of explanatory variables $p \in \mathbb{N}$ — map the observed data $\mathcal{D} \in \mathfrak{D}_{n,p}$ to a subset

$$\hat{S} = \hat{S}^{(n)} = \hat{S}^{(n)}(\mathcal{D}) = E_{n,p}(\mathcal{D}) \subseteq \{1, \dots, p\}\,. \tag{2.11}$$

It is desirable that a variable selection procedure has the following frequentist properties: The probability $P(\hat{S} = S_0)$ of selecting the true model (provided that it is part of the model space) should be as large as possible and the procedure should be variable selection consistent in a sense which will be defined next. In order to do this, we first have to fix the asymptotic setting.

Notation 2.7. Consider GLMs with p_n explanatory variables in an asymptotic setting, where the number of variables p_n is allowed to grow with the sample size n. For $n \in \mathbb{N}$, let $\boldsymbol{X}^{(n)} \in \mathbb{R}^{n \times p_n}$ denote the corresponding design matrices and $\boldsymbol{\beta}_0^{(n)} = \left(\beta_{0,1}^{(n)}, \dots, \beta_{0,p}^{(n)}\right)^T \in \mathbb{R}^{p_n}$ the corresponding true vectors of coefficients. Here, for simplicity, we consider a fixed design setting (compare Remark 2.1). Let $\mathcal{P}^{(n)} = \{1, \dots, p_n\}$ denote the index set of explanatory

variables and let

$$S_0^{(n)} = \{j \in \{1, \dots, p_n\};\ \beta_{0,j}^{(n)} \neq 0\} \tag{2.12}$$

be the true underlying model which is allowed to change with the sample size n; in particular, the number of relevant explanatory variables $s_0^{(n)} = |S_0^{(n)}|$ might grow with n.

Definition 2.8. Consider a particular asymptotic setting as in Notation 2.7 with certain conditions on the growth rates of the number of variables p_n and the number of relevant variables $s_0^{(n)} = |S_0^{(n)}|$ with respect to the sample size n (to be specified per situation).
Then a variable selection procedure $\{E_{n,p}\}$ is called *variable selection consistent* (or just *consistent*, sometimes also called *sparsistent*) for the given asymptotic setting if

$$P\left(E_{n,p}(\mathcal{D}) = S_0^{(n)}\right) = P\left(\hat{S}^{(n)} = S_0^{(n)}\right) \to 1\,, \quad n \to \infty\,. \tag{2.13}$$

Remark 2.2. Variable selection is a special case of model selection: In the general model selection framework one may consider any collection of (maybe totally) different statistical models and tries to select the most appropriate among them (with respect to a certain criterion). For example, in the GLM framework one can compare different GLMs with different link functions (e.g. the logistic-link versus the probit-link for modelling binary data) or different underlying exponential families (e.g. the Poisson family versus the negative-binomial family for modelling count data). In this thesis we do not address these certainly important modelling issues and instead focus on the variable selection problem. Classical references discussing model selection in more generality include Kass and Raftery (1995), Wasserman (2000) and Claeskens and Hjort (2008).

2.1.3. Maximum-Likelihood estimation and identifiability

We need some further notation in order to introduce different variable selection criteria.

Notation 2.9. Given a particular GLM, we denote the likelihood of the response vector $\boldsymbol{y} = (y_1, \dots, y_n)^T$ given the design matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ under the full model (i.e. $S = \{1, \dots, p\}$) by

$$f(\boldsymbol{y} \,|\, \boldsymbol{X}, \mu, \boldsymbol{\beta}, \psi) = \prod_{i=1}^{n} f(y_i \,|\, \boldsymbol{X}_{i,*}, \mu, \boldsymbol{\beta}, \psi), \tag{2.14}$$

where, for each $i = 1, \dots, n$, $f(y_i \,|\, \boldsymbol{X}, \mu, \boldsymbol{\beta}, \psi)$ is of the form (2.1) and where the intercept μ and the coefficient vector $\boldsymbol{\beta}$ are implicitly linked to the conditional expectation $E(Y_i \,|\, \boldsymbol{X}_{i,*})$

according to (2.2). Note that for ease of presentation we use the same notation $f(\cdot)$ for different functions in (2.14), which should be determined by the context and the argument y_i (or $\boldsymbol{y}$). This notational convention is standard, especially in Bayesian statistics (see e.g. Gelman et al., 2014, p. 7), and will be used throughout this thesis.

For a particular subset $S \subseteq \{1, \dots, p\}$, the likelihood of $\boldsymbol{y} = (y_1, \dots, y_n)^T$ under the corresponding GLM is denoted by

$$f(\boldsymbol{y} \mid \boldsymbol{X}_S, \mu_S, \boldsymbol{\beta}_S, \psi_S) = \prod_{i=1}^{n} f(y_i \mid \boldsymbol{X}_{i,S}, \mu_S, \boldsymbol{\beta}_S, \psi_S), \tag{2.15}$$

where $\boldsymbol{X}_S$ is the design matrix restricted to the columns with indices in S (see Notation 2.1) and $\boldsymbol{X}_{i,S}$ denotes it's i-th row ($i = 1, \dots, n$).

Furthermore, let $\hat{\mu}_S, \hat{\boldsymbol{\beta}}_S$ and $\hat{\psi}_S$ denote maximum likelihood estimators (MLEs) of $\mu_S, \boldsymbol{\beta}_S$ and ψ_S under the model induced by $S \subseteq \mathcal{P}$, i.e. they satisfy

$$\left(\hat{\mu}_S, \hat{\boldsymbol{\beta}}_S, \hat{\psi}_S\right) \in \underset{(\mu_S, \boldsymbol{\beta}_S, \psi_S)}{\arg\max}\, f(\boldsymbol{y} \mid \boldsymbol{X}_S, \mu_S, \boldsymbol{\beta}_S, \psi_S)\,. \tag{2.16}$$

In particular, let $\hat{\mu}, \hat{\boldsymbol{\beta}}$ and $\hat{\psi}$ denote MLEs of $\mu, \boldsymbol{\beta}$ and ψ under the full model (i.e. $S = \{1, \dots, p\}$). For brevity, let $\boldsymbol{\theta}_S = (\mu_S, \boldsymbol{\beta}_S, \psi_S)^T$ denote the vector of all parameters in the model induced by $S \subseteq \{1, \dots, p\}$ (not to be confused with a natural parameter) and $\hat{\boldsymbol{\theta}}_S = (\hat{\mu}_S, \hat{\boldsymbol{\beta}}_S, \hat{\psi}_S)^T$ its MLE; a dispersion parameter ψ is included or excluded depending on the employed model.

Notation 2.10. For a subset $S \subseteq \mathcal{P}$, we define $\tilde{\boldsymbol{\beta}}_S = \left(\mu, \beta_S^T\right)^T \in \mathbb{R}^{|S|+1}$ as the vector of regression coefficients of model S including the intercept μ, and define $\tilde{\boldsymbol{X}}_S \in \mathbb{R}^{n \times (|S|+1)}$ as the matrix $\boldsymbol{X}_S$ extended by the first column just containing 1's (compare Notation 2.1).

Using this notation, the normal linear model equation for the model induced by a subset $S \subseteq \mathcal{P}$ can be written as

$$\boldsymbol{Y} = \tilde{\boldsymbol{X}}_S \tilde{\boldsymbol{\beta}}_S + \boldsymbol{\epsilon}\,. \tag{2.17}$$

Remark 2.3. In general, the MLEs in a GLM might not be unique. For example in a normal linear model induced by a subset $S \subseteq \{1, \dots, p\}$ with $\operatorname{rank}(\tilde{\boldsymbol{X}}_S) < |S| + 1$, the MLE of $\tilde{\boldsymbol{\beta}}_S$ is not unique. On the other hand, if $|S| \leq n - 2$ and $\operatorname{rank}(\tilde{\boldsymbol{X}}_S) = |S| + 1$, it is well-known that the unique MLEs of the vector of regression coefficients $\tilde{\boldsymbol{\beta}}_S \in \mathbb{R}^{p+1}$ (including the intercept

μ) and of the error variance $\sigma_S^2 > 0$, under the model S, are given in closed form by

$$\hat{\boldsymbol{\beta}}_S = (\tilde{\boldsymbol{X}}_S^T \tilde{\boldsymbol{X}}_S)^{-1} \tilde{\boldsymbol{X}}_S^T \boldsymbol{y}, \quad \hat{\sigma}_S^2 = \frac{1}{n} \boldsymbol{y}^T \left(\boldsymbol{I}_n - \boldsymbol{H}_S\right) \boldsymbol{y}\,, \tag{2.18}$$

where $\boldsymbol{I}_n \in \mathbb{R}^{n \times n}$ denotes the identity matrix of dimension n and

$$\boldsymbol{H}_S = \tilde{\boldsymbol{X}}_S (\tilde{\boldsymbol{X}}_S^T \tilde{\boldsymbol{X}}_S)^{-1} \tilde{\boldsymbol{X}}_S^T \in \mathbb{R}^{n \times n} \tag{2.19}$$

denotes the *hat matrix*, i.e. the orthogonal projection on the column space of $\tilde{\boldsymbol{X}}_S$. The estimated response vector under model S is given by $\hat{\boldsymbol{y}} = \boldsymbol{H}_S \boldsymbol{y} \in \mathbb{R}^n$.

Note that for GLMs different than the normal linear model (in particular for logistic and Poisson regression models), there is in general no closed form analytic expression for the MLEs of the model parameters. Therefore, in these cases one has to use numerical algorithms like iteratively reweighted least squares (IRLS) in order to compute approximations to the MLEs (see e.g. Hardin and Hilbe, 2007, Chapter 3). Furthermore, note that the MLEs of a GLM might not even exist in certain situations, as for example in a logistic regression model with complete or quasi-complete separation (see e.g. Albert and Anderson, 1984).

Remark 2.4. For the identifiability of the vector of regression coefficients β_S in a GLM induced by a subset $S \subseteq \{1, \ldots, p\}$, the columns $\{\boldsymbol{X}_j;\ j \in S\}$ have to be linearly independent, i.e. $\operatorname{rank}(\boldsymbol{X}_S) = |S|$ (see e.g. Christensen, 2011, Proposition 2.1.6). We have $\operatorname{rank}(\boldsymbol{X}_S) \leq \min\{n, |S|\}$ and therefore we necessarily need $|S| \leq n$ for the identifiability of β_S. Since we always include an intercept μ in the corresponding GLM, we need to have $|S| \leq n - 1$; while for the normal linear model with unknown variance σ^2 we even need to have $|S| \leq n - 2$. For identifiability reasons and in order to avoid obvious overfitting, in the following we will consider the restricted space of models

$$\mathcal{M} = \{S \subseteq \mathcal{P};\ |S| \leq n - p_0\}\,, \tag{2.20}$$

where p_0 denotes the number of parameters of the "null" model $S = \emptyset$, i.e. $p_0 = 1$ if only an intercept is used and $p_0 = 2$ if an additional dispersion parameter is fitted.

2.2. Information criteria

In this section we begin our discussion of different variable selection methods with classical ℓ_0-type information criteria which penalize the size of a model (i.e. the number of selected variables) in the most "natural" way.

2.2.1. AIC and BIC

As already mentioned in the introduction of this thesis, the classical approach to the variable selection problem is to come up with a selection criterion and then to solve the resulting optimization problem. The arguably most prominent selection criteria are the Akaike Information Criterion (AIC, Akaike, 1974) and the Bayesian Information Criterion (BIC, Schwarz, 1978), which we want to recall in the following definition.

Definition 2.11 (Akaike, 1974; Schwarz, 1978)**.** For a model $S \in \mathcal{M}$ the *Akaike Information Criterion (AIC)* is defined by

$$\mathrm{AIC}(S) = -2 \log \left(f(\boldsymbol{y} \,|\, \boldsymbol{X}_S, \hat{\boldsymbol{\theta}}_S) \right) + 2|S| \,, \tag{2.21}$$

while the *Bayesian Information Criterion (BIC)* is defined by

$$\mathrm{BIC}(S) = -2 \log \left(f(\boldsymbol{y} \,|\, \boldsymbol{X}_S, \hat{\boldsymbol{\theta}}_S) \right) + \log(n)|S| \,, \tag{2.22}$$

where $f(\boldsymbol{y} \,|\, \boldsymbol{X}_S, \hat{\boldsymbol{\theta}}_S)$ denotes the maximized likelihood under model S (compare Notation 2.9). According to AIC and BIC we "estimate" the true active set S_0 respectively by

$$\hat{S}_{\mathrm{AIC}} = \operatorname*{arg\,min}_{S \in \mathcal{M}} \mathrm{AIC}(S) \quad \text{and} \quad \hat{S}_{\mathrm{BIC}} = \operatorname*{arg\,min}_{S \in \mathcal{M}} \mathrm{BIC}(S) \,. \tag{2.23}$$

Remark 2.5. Note that both the AIC and the BIC penalize the number of regression coefficients $|S|$ that have to be estimated in the model $S \in \mathcal{M}$. Strictly speaking, the term $|S|$ should be replaced by the term $|S|+1$ or even $|S|+2$ since the intercept and possibly the dispersion parameter have to be estimated, too. But as they are always included (or excluded) in all models which we consider, we can omit the common additional constant leaving the solution of the optimization problem unchanged. Following the same argumentation we will similarly proceed with other selection criteria.

The AIC can asymptotically be derived by minimising the Kullback-Leibler divergence be-

tween the distribution from a candidate model and the true underlying distribution (Akaike, 1974; Bozdogan, 1987); the BIC can asymptotically be derived by maximising the posterior model probability in a Bayesian context (Schwarz, 1978). Here one has to distinguish between two different targets of variable selection: The first one — on which the AIC is based — is the aim of optimal prediction, while the second one — on which the BIC is based — is the aim of identifying the true underlying model. For normal linear models it can be shown that the model selected by the AIC asymptotically attains the minimal mean squared error among all models considered (see e.g. Shao, 1997), while the BIC is variable selection consistent (compare Definition 2.8) under mild conditions if the number of explanatory variables is bounded (see e.g. Nishii, 1984 and Shao, 1997). A nice philosophical and practical comparison of the AIC and the BIC can for example be found in Burnham and Anderson (2004).

It turns out, that the two described aims of variable selection are fundamentally different and that the desirable theoretical properties of the AIC and the BIC cannot be simultaneously shared by any selection criterion (Yang, 2005). In this thesis our methodology is primarily motivated by the second aim underlying the BIC, namely the identification of the true model, but we will also investigate and evaluate the predictive and estimative performances of the proposed methods.

2.2.2. Extended BIC

As already indicated above, the BIC can be obtained as an approximation to a fully Bayesian analysis with a uniform prior on the model space, i.e. $\pi(S) = \frac{1}{2^p}$ for all $S \subseteq \{1, \ldots, p\}$. The problem with this prior assumption in a high-dimensional situation is that the prior probability that the size of a model S is relatively large is much higher than that the model is sparse, i.e. we have

$$\pi(|S| = s_1) \ll \pi(|S| = s_2), \text{ if } s_1 < s_2 < p - s_1 \,. \tag{2.24}$$

So the model prior underlying the BIC is not acceptable in a high-dimensional framework where the truth is assumed to be sparse. Therefore, Chen and Chen (2008) propose a modified version of the BIC called the Extended Bayesian Information Criterion (EBIC) which is suited for high-dimensional situations. The difference to the original BIC is that

they use a different prior on the model space: For a fixed additional parameter $\gamma \in [0,1]$ and $S \subseteq \{1,\dots,p\}$ let

$$\pi(S) \propto \binom{p}{|S|}^{-\gamma}. \tag{2.25}$$

If $\gamma = 1$, then $\pi(S) = \frac{1}{p+1}\binom{p}{|S|}^{-1}$, so the prior gives equal probability to each model size, and to each model of the same size. If $\gamma = 0$, then we obtain the uniform prior used in the original BIC. Similar to the derivation of the BIC one asymptotically obtains the EBIC (see Chen and Chen, 2008 for details).

Definition 2.12 (Chen and Chen, 2008)**.** For a model $S \in \mathcal{M}$, the *Extended Bayesian Information Criterion* (EBIC_γ) with parameter $\gamma \in [0,1]$ is defined as

$$\text{EBIC}_\gamma(S) = -2\log\Big(f(\boldsymbol{y} \mid \boldsymbol{X}_S, \hat{\boldsymbol{\theta}}_S)\Big) + \log(n)|S| + 2\gamma\log\binom{p}{|S|}. \tag{2.26}$$

So according to EBIC_γ we "estimate" the true active set S_0 by

$$\hat{S}_{\text{EBIC}_\gamma} = \underset{S\in\mathcal{M}}{\arg\min}\ \text{EBIC}_\gamma(S)\,. \tag{2.27}$$

Remark 2.6. In the literature one can find an alternative definition of the EBIC_γ (see e.g. Chen and Chen, 2012), which will also be used in the subsequent simulated and real data examples of this thesis. For a model $S \in \mathcal{M}$, the EBIC_γ with parameter $\gamma \in [0,1]$ is alternatively defined by

$$\text{EBIC}_\gamma(S) = -2\log\Big(f(\boldsymbol{y} \mid \boldsymbol{X}_S, \hat{\boldsymbol{\theta}}_S)\Big) + \Big(\log(n) + 2\gamma\log(p)\Big)|S|\,. \tag{2.28}$$

Note that asymptotically we have

$$\log\binom{p}{|S|} = \log\left(\frac{p^{|S|}}{|S|!}\left(1-\frac{1}{p}\right)\left(1-\frac{2}{p}\right)\dots\left(1-\frac{|S|-1}{p}\right)\right) \asymp |S|\log(p)\,, \tag{2.29}$$

for a fixed model $S \in \mathcal{M}$, where the notation $a_p \asymp b_p$ indicates that for sequences $(a_p)_{p\in\mathbb{N}}$ and $(b_p)_{p\in\mathbb{N}}$ it holds that $\lim_{p\to\infty}\frac{a_p}{b_p} = 1$. Thus the definitions of EBIC_γ in equation (2.26) and (2.28) are asymptotically equivalent if only models of bounded size are considered. However, the original EBIC_γ definition given in (2.26) might be preferred in an asymptotic setting where the size of the true active set is allowed to increase with the sample size (compare Foygel and Drton, 2010 and Luo and Chen, 2013).

Remark 2.7. Certain alternative modifications of the BIC have been proposed for the high-

dimensional setting, which are derived from different model prior assumptions: The modified BIC (mBIC, Bogdan et al., 2004) assumes an underlying binomial model prior, i.e. the prior probability of a model $S \subseteq \mathcal{P}$ is given by

$$\pi(S) = \left(\frac{q}{p}\right)^{|S|} \left(1 - \frac{q}{p}\right)^{p-|S|} , \tag{2.30}$$

where $q \in (0, p)$ is an additional parameter which is usually chosen to be small in order to reflect the expected sparsity of the truth. The mBIC2 (Frommlet et al., 2012) has been proposed as an adjustment of the mBIC motivated by a multiple testing correction. Żak-Szatkowska and Bogdan (2011) show that the EBIC_γ, the mBIC and the mBIC2 are asymptotically equivalent under certain conditions.

However, Żak-Szatkowska and Bogdan (2011) argue that the mBIC and the mBIC2 are more "natural" than the EBIC_γ in the following sense: If $\gamma > 0$, then the EBIC_γ becomes more liberal for models S with $|S| > \frac{p}{2}$ (compare (2.24)); additionally, the prior expected number of relevant explanatory variables is $\frac{p}{2}$ for all choices of $\gamma \in [0, 1]$ in the EBIC_γ. On the other hand, the corresponding prior expected number of relevant variables in the mBIC is given by q (which is chosen to be small) and the prior probability of the models of size greater than q decreases with growing dimension; so this prior reflects more naturally the assumed underlying sparsity of the high-dimensional problem. In defence of the EBIC_γ we want to stress that in a situation with $p \gg n$ only relatively low-dimensional models (at least with $|S| < n$ due to identifiability reasons, compare Remark 2.4) should be considered and therefore the undesirable property of EBIC_γ for models with $|S| > \frac{p}{2}$ is not relevant in practice. In the following discussions and simulations we focus on the EBIC_γ, which also seems to be more established in the literature than the mBIC or the mBIC2.

Remark 2.8. Scott and Berger (2010) discuss multiplicity corrections in empirical Bayesian and fully Bayesian frameworks, addressing the implicit multiple-testing issue inherent to the variable selection problem. In particular, in a fully Bayesian framework a multiplicity correction is naturally induced by the conjugate beta-binomial prior

$$\pi(S \,|\, \omega) = \omega^{|S|} (1 - \omega)^{p-|S|} \ , \ S \subseteq \mathcal{P} \,, \quad \omega \sim \mathcal{B}e(a_\omega, b_\omega) \,, \tag{2.31}$$

where $a_\omega, b_\omega > 0$ are parameters controlling the prior inclusion probability ω of each variable (see Kohn et al., 2001 for details). Note that the "default" choice $a_\omega = b_\omega = 1$ yields a

uniform prior on the inclusion probability ω and leads to the model prior $\pi(S) = \frac{1}{p+1}\binom{p}{|S|}^{-1}$ for $S \subseteq \mathcal{P}$, corresponding exactly to the prior underlying the EBIC_γ for $\gamma = 1$.

It has been shown by Chen and Chen (2008) that, under a mild asymptotic identifiability condition, the EBIC_γ is variable selection consistent for normal linear models. The detailed result is stated in Theorem 2.2 after the introduction and discussion of the sufficient condition.

Notation 2.13. Consider normal linear models with an asymptotic setting as in Notation 2.7, where we assume that the true model $S_0 = \{j \in \{1, \dots, p_n\};\ \beta_{0,j} \neq 0\}$ is fixed; in particular, the number of relevant explanatory variables $s_0 = |S_0|$ is not allowed to grow with n. In the following we make use of Notation 2.10 and let $\tilde{\boldsymbol{\beta}}_0 \in \mathbb{R}^{p+1}$ denote the true regression vector including the true intercept μ_0. For $n \in \mathbb{N}$ and a subset $S \subseteq \{1, \dots, p_n\}$, let

$$\boldsymbol{H}_S^{(n)} := \tilde{\boldsymbol{X}}_S^{(n)} \left(\left(\tilde{\boldsymbol{X}}_S^{(n)} \right)^T \tilde{\boldsymbol{X}}_S^{(n)} \right)^- \left(\tilde{\boldsymbol{X}}_S^{(n)} \right)^T \in \mathbb{R}^{n \times n} \tag{2.32}$$

be the *hat matrix*, where $\boldsymbol{A}^-$ denotes the generalized inverse of a matrix $\boldsymbol{A}$ (i.e. it satisfies $\boldsymbol{A}\boldsymbol{A}^-\boldsymbol{A} = \boldsymbol{A}$). Furthermore, let

$$\delta_S^{(n)} := \| \tilde{\boldsymbol{X}}^{(n)} \tilde{\boldsymbol{\beta}}_0^{(n)} - \boldsymbol{H}_S^{(n)} \tilde{\boldsymbol{X}}^{(n)} \tilde{\boldsymbol{\beta}}_0^{(n)} \|_2^2 \,, \tag{2.33}$$

where $\|.\|_2$ denotes the Euclidean norm.

Definition 2.14 (Chen and Chen, 2008)**.** Given the setting of Notation 2.13, the *asymptotic identifiability condition* holds if and only if

$$\lim_{n \to \infty} \min \left\{ \frac{\delta_S^{(n)}}{\log(n)};\ S \neq S_0, |S| \leq |S_0| \right\} = \infty \,. \tag{2.34}$$

The quantity $\delta_S^{(n)}$ is defined as the squared Euclidean distance between the true expected response $\boldsymbol{y}_0^{(n)} = \tilde{\boldsymbol{X}}^{(n)} \tilde{\boldsymbol{\beta}}_0^{(n)}$ and the predicted response $\hat{\boldsymbol{y}}_S^{(n)} = \boldsymbol{H}_S^{(n)} \tilde{\boldsymbol{X}}^{(n)} \tilde{\boldsymbol{\beta}}_0^{(n)}$ when observing the true expected response but projecting onto the subspace S. Thus, intuitively, $\delta_S^{(n)}$ measures how much predictive power is lost when we consider a model $S \neq S_0$ instead of the true model S_0. Therefore, the asymptotic identifiability condition can be interpreted as the assumption that asymptotically no model of size at most $s_0 = |S_0|$ can predict the response equally well as the true model S_0. Next, we state another condition which implies the asymptotic identifiability condition and gives some further insight (Chen and Chen, 2008).

Lemma 2.1 (Chen and Chen, 2008)**.** *If we have*

$$\lim_{n\to\infty} \min_{S\neq S, |S|\leq |S_0|} \max_{k\in S_0} \left[\log(n)^{-1} \left|\left|\left(\boldsymbol{I}_n - \boldsymbol{H}^{(n)}_{(S_0\setminus\{k\}\cup S)}\right)\boldsymbol{X}^{(n)}_k\right|\right|_2\right] = \infty\,, \tag{2.35}$$

then the asymptotic identifiability condition is satisfied.

Condition (2.35) is fulfilled if, for all $S \neq S_0$ with $|S| \leq |S_0|$, at least one column of the design $\boldsymbol{X}^{(n)}_{S_0}$ of the true model S_0 is linearly independent of the other columns in $\boldsymbol{X}^{(n)}_{S_0\cup S}$. In particular, collinearities among the columns in $\boldsymbol{X}^{(n)}_{\mathcal{P}^{(n)}\setminus S_0}$ (i.e. among "noise" variables) do not lead to failure of the asymptotic identifiability condition (Chen and Chen, 2008).

Theorem 2.2 (Chen and Chen, 2008)**.** *Consider normal linear models with an asymptotic setting as in Notation 2.7, where the true model S_0 is assumed to be fixed. Suppose that the asymptotic identifiability condition (2.34) is satisfied. Furthermore, assume that p_n is growing with the sample size n with the rate $p_n = \mathcal{O}(n^k)$ for some constant $k > 0$.*
If $\gamma > 1 - \frac{1}{2k}$ and if the maximal cardinality $K_{max} > |S_0|$ of the considered models (of which the $EBIC_\gamma$ is computed) does not depend on p_n, then the $EBIC_\gamma$ is variable selection consistent, i.e. we have

$$P\left(EBIC_\gamma(S_0) < \min\{EBIC_\gamma(S);\ S\subseteq\mathcal{P}^{(n)}, S\neq S_0, |S| = j\}\right) \to 1\,,\ n\to\infty\,, \tag{2.36}$$

for all $j = 0,\ldots,K_{max}$.

The proof of Theorem 2.2, which can be found in Chen and Chen (2008), also indicates that the usual BIC is generally not variable selection consistent if p_n grows with a rate faster than $\sqrt{n}$. The consistency result concerning the extended BIC has been extended by Luo and Chen (2013) and Foygel and Drton (2010) to the setting of a diverging number of relevant explanatory variables in normal linear models and in Gaussian graphical models, respectively. Chen and Chen (2012) provide conditions for the consistency of the extended BIC in GLMs (again for a fixed number of relevant explanatory variables and a fixed design setting). Barber and Drton (2015) show that the extended BIC is also consistent under certain conditions for logistic regression models with random designs. For the sake of brevity, we do not present the detailed results which involve somewhat more technical conditions than the result stated in Theorem 2.2. Although these conditions might be quite technical, they are usually much weaker than the conditions needed for similar consistency results of ℓ_1-type methods like the Lasso (see Section 2.4).

2.2.3. Generalized Information Criteria

The selection criteria that have been discussed so far are special instances of the family of Generalized Information Criteria (GIC) (compare Nishii, 1984 and Zhang et al., 2010) which we define by

$$\mathrm{GIC}_{\lambda_{n,p}}(S) = -2\log\left(f(\boldsymbol{y} \mid \boldsymbol{X}_S, \hat{\boldsymbol{\theta}}_S)\right) + \lambda_{n,p}|S|\,, \quad S \in \mathcal{M}\,, \tag{2.37}$$

for some regularization constants $\lambda_{n,p} > 0$ that might depend on the sample size n and the number of explanatory variables p. In particular, $\lambda_{n,p} = 2$ yields the AIC, $\lambda_{n,p} = \log(n)$ yields the BIC and $\lambda_{n,p} = \log(n) + 2\gamma\log(p)$ yields the EBIC_γ as given in equation (2.28). Many other well-known selection criteria also fall into the GIC family, such as Mallows C_p (Mallows, 1973; Mallows, 1995) or the risk inflation criterion (RIC, Foster and George, 1994). We refer to Nishii (1984) and Shao (1997) for details and general theoretical properties of the GIC depending on the choice of the regularization parameter $\lambda_{n,p}$ in the setting of normal linear models. Note that the GIC given in equation (2.37) does not coincide with the generalized information criterion proposed in Konishi and Kitagawa (1996).

All the criteria of the GIC family take a similar form which penalize the negative log-likelihood (as a measure of model fitting) with a constant times the number of non-zero regression coefficients (as a measure of model complexity). The form of the penalty is also referred to as an ℓ_0-type penalty, since the ℓ_0-norm of any coefficient vector $\boldsymbol{\beta} \in \mathbb{R}^p$ with non-zero entries only for indices in S is given by $\|\boldsymbol{\beta}\|_0 = \sum_{j=1}^p \mathbb{1}_{\{\beta_j \neq 0\}} = |S|$. Note that strictly speaking $\|.\|_0$ is not a norm, since it is not absolutely homogeneous, but it is common practice to speak of the "ℓ_0-norm" for simplicity.

The challenging problem with ℓ_0-type selection criteria is that the resulting combinatorial optimization problems are in general very difficult to solve if there are many possible explanatory variables p. Huo and Ni (2007) show that for normal linear models the problem of optimizing (2.37) for any fixed $\lambda_{n,p}$ is equivalent to an instance of the sparse approximate solution (SAS) problem, which has been shown to be NP-hard by Natarajan (1995). Therefore, even in the simple case of normal linear models, optimizing any GIC of the form (2.37) — which is also referred to as *best subset selection* — is in general an NP-hard problem. Nevertheless, different optimization algorithms have been proposed to attack the generally NP-hard problem, of which we want to give a short overview in the next section.

2.3. Discrete optimization algorithms

In this section we discuss selected algorithms for solving the discrete optimization problems induced by ℓ_0-type variable selection criteria. The algorithms can be divided into two categories: First, in Section 2.3.1 we consider exact algorithms which can guarantee that the global optimal solution is found; second, in Section 2.3.2 we discuss heuristic algorithms which try to trace "good" approximate solutions in a heuristic way, but in general there is no guarantee that one obtains the optimal solution. We refer to Hocking (1976), Thompson (1978) and Miller (1984) for detailed discussions of classical variable selection methods in the setting of linear regression models, including exact algorithms like best subset selection and heuristic algorithms like stepwise methods.

In order to simplify and unify the further presentation, we introduce a general notation that will be used throughout the remainder of this thesis.

Notation 2.15. In the following, let $C : \mathcal{M} \to \mathbb{R}$ denote a certain selection criterion with respect to maximization (as for example the negative AIC or the negative EBIC_γ). Here, without loss of generality we consider the maximization problem (instead of the minimization problem), i.e. the aim is to compute $S^* = \arg\max_{S \in \mathcal{M}} C(S)$.

2.3.1. Exact optimization algorithms

The most obvious approach to solve the optimization problem of minimizing (2.37) is a full enumeration in which the given selection criterion is evaluated for all possible models. With this approach it is feasible to compute the exact solution of the optimization problem, as long as the number of explanatory variables is not too large. Note that for p possible explanatory variables there are $|\mathcal{M}_{\text{full}}| = 2^p$ possible models in $\mathcal{M}_{\text{full}} = \{S \subseteq \{1, \ldots, p\}\}$ (compare Notation 2.5). For example, for $p = 50$ we have $|\mathcal{M}_{\text{full}}| \approx 1.13 \times 10^{15}$, i.e. there are approximately one quadrillion possible models. Even if we consider the restricted model space $\mathcal{M} = \{S \subseteq \{1, \ldots, p\};\ S \leq n - p_0\}$ (compare Remark 2.4) we still have

$$|\mathcal{M}| = \binom{p}{0} + \binom{p}{1} + \binom{p}{2} + \ldots + \binom{p}{n - p_0} \tag{2.38}$$

possible models for which the criterion has to be evaluated (e.g. for $p = 50$, $n = 25$ and $p_0 = 2$ we still have $|\mathcal{M}| \approx 3.78 \times 10^{14}$). So the full enumeration approach is certainly not

computationally feasible in a high-dimensional situation where p is very large. An additional potential issue is that the computation of the criterion value for a particular model can be very "expensive", since the MLEs of the model parameters might not be available in closed form and so for each model an additional (continuous) optimization problem has to be solved (this is for example the case for logistic or Poisson regression models).

However, in normal linear models the MLEs of the model parameters are given in closed form (see Remark 2.3). This fact and the "simpler" structure of normal linear models allow the development of certain alternative optimization approaches in order to identify the global solution of the generally NP-hard optimization problem for moderate sizes of p: Note that for normal linear models the GIC for a subset $S \subseteq \{1, \ldots, p\}$ can equivalently expressed as

$$\mathrm{GIC}_{\lambda_{n,p}}(S) = n \cdot \log\left(\frac{\mathrm{RSS}(S)}{n}\right) + \lambda_{n,p}|S|, \tag{2.39}$$

where $\mathrm{RSS}(S) = \|\boldsymbol{y} - (\hat{\mu}_S + \boldsymbol{X}_S\hat{\boldsymbol{\beta}}_S)\|_2^2$ denotes the residual sum of squares under the model S. The main idea underlying alternative algorithms is to use the fact that for subsets $S_1 \subseteq S_2 \subseteq \mathcal{P}$ we have $\mathrm{RSS}(S_1) \geq \mathrm{RSS}(S_2)$ and that the problem of minimizing (2.39) among all subsets $S \subseteq \mathcal{P}$ can be split into $p-1$ optimization problems in which, for each $k = 1, \ldots, p-1$, the residual sum of squares $\mathrm{RSS}(S)$ is minimized for subsets $S \subseteq \mathcal{P}$ with $|S| = k$. Based on this idea, Furnival and Wilson (1974) propose clever branch-and-bound strategies (also called "leaps-and-bounds") which significantly reduce the number of model evaluations in the variable selection framework for normal linear models; furthermore, they reduce the number of operations needed for computing the MLEs of $\boldsymbol{\beta}_S$ for models $S \subseteq \mathcal{P}$ (by avoiding the inversion of full matrices $\boldsymbol{X}_S^T\boldsymbol{X}_S$). Certain extensions and modifications of the leaps-and-bounds approach have been developed, for example in Narendra and Fukunaga (1977), Roberts (1984), Ridout (1988), Somol et al. (2004), Ni and Huo (2005) and Hofmann et al. (2007).

Very recently, a conceptually different approach has been proposed by Bertsimas et al. (2016) for minimizing ℓ_0-type criteria of the form (2.39). The approach combines discrete first-order methods (motivated from continuous first-order methods) and mixed integer optimization methods; by this, it becomes practically feasible to solve problems with $n \approx 1000$ and $p \approx 100$ exactly and to find approximate solutions for $n \approx 100$ and $p \approx 1000$. Although this new direction of applying modern discrete optimization techniques in order to solve the

original ℓ_0-type variable selection problem is very interesting and promising, due to brevity we do not give a detailed account of these methods here.

2.3.2. Heuristic optimization algorithms

We have seen that in a high-dimensional situation with many possible explanatory variables, a full enumeration is clearly not computationally feasible in order to solve ℓ_0-type optimization problems. Even though more efficient exact methods like leaps-and-bounds or mixed integer optimization approaches have been developed, these methods are only practical if the number of explanatory variables is not too large. Furthermore, they are only applicable in the framework of normal linear models (and not for different GLMs). Therefore, it is natural to consider heuristic algorithms for finding approximate solutions to the generally NP-hard optimization problems.

Algorithm 2.1 Forward Stepwise Selection (FS)

Input:

- Data $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$ with sample size n and p explanatory variables
- $C : \mathcal{M} \to \mathbb{R}$ model selection criterion with respect to maximization

Algorithm:

(1) Set $m := \min\{n - p_0, p\}$, where p_0 is the number of parameters of the "null" model. Set $S_F^{(0)} := \emptyset$.

(2) For $t = 1, \ldots, m$:

  (a) Compute $j^{(t)} := \arg\max_{j \in \mathcal{P} \setminus S_F^{(t-1)}} C\left(S_F^{(t-1)} \cup \{j\}\right)$.

  (b) Set $S_F^{(t)} := S_F^{(t-1)} \cup \{j^{(t)}\}$.

Output:

- Final model selected by FS: $\hat{S}_F = \arg\max\left\{C(S_F^{(0)}), \ldots, C(S_F^{(m)})\right\}$

Well-known heuristic algorithms for variable selection are Forward Stepwise Selection (FS) and Backward Stepwise Selection (BS) (see e.g. Miller, 1984 and Friedman et al., 2009). In Forward Stepwise Selection (given as Algorithm 2.1) one begins with the "null" model (i.e. $S_F^{(0)} = \emptyset$) including only an intercept (and possibly a dispersion parameter) and then

successively adds that explanatory variable which most improves the current fit, until the maximum number of explanatory variables is reached or the current model is saturated (resulting in a perfect fit of the data). By this we obtain an increasing sequence of nested subsets:

$$\emptyset = S_F^{(0)} \subseteq S_F^{(1)} \subseteq S_F^{(2)} \subseteq \ldots \subseteq S_F^{(m)}\,, \tag{2.40}$$

with $m := \min\{n - p_0, p\}$, where p_0 denotes the number of parameters of the "null" model.

Algorithm 2.2 Backward Stepwise Selection (BS)

Input:

- Data $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$ with sample size n and p explanatory variables with the restriction that $p + p_0 \leq n$, where p_0 is the number of parameters of the "null" model
- $C : \mathcal{M} \to \mathbb{R}$ model selection criterion with respect to maximization

Algorithm:

(1) Set $S_B^{(p)} := \{1, \ldots, p\}$.

(2) For $t = 1, \ldots, p$:

  (a) Compute $j^{(t)} := \arg\max_{j \in S_B^{(p-t+1)}} C\left(S_B^{(p-t+1)} \setminus \{j\}\right)$.

  (b) Set $S_B^{(p-t)} := S_B^{(p-t+1)} \setminus \{j^{(t)}\}$.

Output:

- Final model selected by BS: $\hat{S}_B = \arg\max\left\{C(S_B^{(0)}), \ldots, C(S_B^{(p)})\right\}$

In Backward Stepwise Selection (given as Algorithm 2.2) one starts with the full model including all available explanatory variables (i.e. $S_B^{(p)} = \{1, \ldots, p\}$) and then successively removes that explanatory variable leading to the lowest reduction in the goodness of fit, until the "null" model is reached. By this we obtain a decreasing sequence of nested subsets:

$$\{1, \ldots, p\} = S_B^{(m)} \supseteq S_B^{(p-1)} \supseteq S_B^{(p-2)} \supseteq \ldots \supseteq S_B^{(0)} = \emptyset\,, \tag{2.41}$$

with $m = p \leq n - p_0$.

The final model selected by FS (or BS) when using a certain selection criterion C with respect to maximization is given by

$$\hat{S}_i = \arg\max\left\{C(S_i^{(0)}), \ldots, C(S_i^{(m)})\right\}\,, \tag{2.42}$$

for $i = F$ (or $i = B$). It should be stressed that the sequences of nested subsets in FS and BS are usually independent of different selection criteria (e.g. for ℓ_0-type criteria), so that only a single run of the algorithms is needed and then the final model can be selected according to different criteria. Note that BS is typically only feasible if $p + p_0 \leq n$, since otherwise any model S with $|S| \geq n - p_0$ fits the data perfectly and thus in the "early steps" of the algorithms the identification of the variables with the smallest contribution to the fit is not possible. On the other hand, Forward Stepwise Selection can always be applied, even in the case of high-dimensional data with $p > n - p_0$.

Even though stepwise methods like FS and BS are very greedy and thus do not necessarily identify the best model according to the employed criterion, they have certain advantages over a full enumeration (i.e. best subset selection): They are computationally efficient even when there are many explanatory variables and, due to their greediness and lower variance in comparison to best subset selection, they might prevent from overfitting (see e.g. Friedman et al., 2009, p. 59).

Another remark in this direction concerns the stopping rule in Forward Stepwise Selection (FS): A variant of the algorithm described above (in the following referred to as FS2) is that one stops the algorithm as soon as the criterion value is not improved, i.e. return the model $\hat{S}_{F_2} := S_F^{(k)}$ if

$$C(S_F^{(0)}) \leq C(S_F^{(1)}) \leq \ldots \leq C(S_F^{(k)}) \geq C(S_F^{(k+1)})\,. \tag{2.43}$$

By this "early stopping" of the algorithm one might obtain a sparser model which turns out to be beneficial if the criterion used is too liberal and possibly "overfits" the data (with the additional effect of saving some computation time).

Other modifications and extensions of stepwise methods have been developed which for example can perform both "forward" and "backward" steps, depending on which direction yields a larger improvement of the current criterion value (see e.g. Zhang, 2011). Note that a version of this strategy is also implemented in the R-function `step`.

Stepwise methods have been and are still frequently employed in applied statistical work and one has to admit that their use has certainly led to a lot of misconceptions and misinterpretations of results, e.g. by using confidence intervals and p-values based only on the final model selected by stepwise methods, without taking the previous model selection step into account (see e.g. Miller, 1984 for classical criticism of such practices). However, due to their

simplicity and computational efficiency they are valuable tools for finding a single well-fitted and interpretable model. In fact, we will demonstrate in Chapters 5 and 6 that Forward Stepwise Selection can be quite competitive for variable selection in sparse high-dimensional situations (compare also Wang, 2009).

Therefore, we do not support the view which can be found in certain parts of the literature (as for example in Flom and Cassell, 2007) that stepwise methods should be completely abandoned and that ℓ_1-type methods like the Lasso (see Section 2.4) should be used instead. Furthermore, even though stepwise methods might be difficult to analyse from a theoretical point of view due to their algorithmic definition (compare Hastie et al., 2015, p. 86), some consistency results have also been derived for stepwise-like methods under (very restrictive) conditions which are surprisingly similar to those needed for the variable selection consistency of the Lasso (compare Theorem 2.4 in Section 2.4.2). Tropp and Gilbert (2007) and Zhang (2009) show that certain irrepresentable conditions (compare Definition 2.19 in Section 2.4.2) are sufficient and necessary for the variable selection consistency of Orthogonal Matching Pursuit (OMP), which is a close variant of Forward Stepwise Selection being more popular in the Signal Processing community. We do not want to present the theoretical details of such results here, but instead conclude this discussion with a quote, which shows that even proponents of convex regularization methods do recognize the benefits of stepwise methods: "Forward-stepwise methods are very efficient, and are hard to beat in terms of finding good, sparse subsets of variables" (Hastie et al., 2015, p. 86).

Apart from well-known stepwise methods, many other heuristic algorithms for discrete optimization problems have also been used in the variable selection context. Proposed methods include different variants of local searches like "shotgun stochastic search" (Hans et al., 2007), simulated annealing (see e.g. Brooks and Morgan, 1995 and Brooks et al., 2003) and tabu search (see e.g. Glover, 1990 and Pacheco et al., 2009). Furthermore, different population-based algorithms inspired from biological and physical processes have been developed such as genetic algorithms (see e.g. Holland, 1992, Yang and Honavar, 1998 and Kapetanios, 2007), particle swarm algorithms (see e.g. Eberhart and Kennedy, 1995 and Unler and Murat, 2010), ant colony algorithms (see e.g. Kennedy and Eberhart, 1997) and hybrid algorithms which combine different heuristics (e.g. "SAGA", Gheyas and Smith, 2010). A good overview of some of the mentioned heuristic optimization algorithms and their application on statistical

problems (including variable selection) can be found in Givens and Hoeting (2012, Chapter 3), while Xue et al. (2016) present an up-to-date survey of evolutionary algorithms for variable selection.

It should be noted that by the famous "no free lunch theorem" (Wolpert and Macready, 1997) any two different algorithms have the same average performance when averaging over all discrete optimization problems and assuming a uniform problem distribution (i.e. each objective function is assumed to be equally likely). Despite this negative result regarding the possible construction of a uniformly best optimization algorithm, the development of context-specific heuristics, which work well on certain subclasses of problems, remains an active research area. In particular, in the variable selection context the assumption that any objective function $C : \mathcal{M} \to \mathbb{R}$ (with respect to maximization) is equally "likely" seems not to be reasonable, since for example one might expect that for any $j \in S^* = \arg\max_{S \in \mathcal{M}} C(S)$, we will have $C(\{j\} \cup S') > C(S')$ for "many" $S' \in \mathcal{M}$ with $j \notin S'$. In other words, a variable selected to be in the best subset according to the criterion is "likely" to improve the criterion value when it is added to a set of other variables. A corresponding notion will be made more precise for the analysis of the Adaptive Subspace (AdaSub) method in Chapter 4.

2.4. Convex regularization methods

An alternative to the attempt of solving a very difficult optimization problem with heuristic methods is to consider a "simpler" relaxation of the original problem which can be solved efficiently. Indeed, with the advent of ever higher-dimensional data, in the 90's the focus shifted from solving discrete optimization problems to solving continuous, convex relaxations of the original ℓ_0-type problem. The following selective presentation of the corresponding methodology is mostly based on the two highly recommended books by Bühlmann and van de Geer (2011) and Hastie et al. (2015).

2.4.1. The Lasso and Bridge Regression

Tibshirani (1996) proposes the celebrated Lasso (Least Absolute Shrinkage and Selection Operator), which solves a convex optimization problem with an ℓ_1-penalty on the regression coefficients.

Definition 2.16 (Tibshirani, 1996)**.** The *Lasso* with penalty parameter $\lambda > 0$ is given by the optimization problem

$$\hat{\mu}(\lambda), \hat{\boldsymbol{\beta}}(\lambda) = \underset{\mu \in \mathbb{R},\, \boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \left(-\frac{2}{n} \log \left(f(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{\beta}, \mu) \right) + \lambda \|\boldsymbol{\beta}\|_1 \right), \tag{2.44}$$

where $\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p} |\beta_j|$ denotes the ℓ_1-norm of $\boldsymbol{\beta}$ and $f(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{\beta}, \mu)$ is the likelihood of the data under the full model.

Remark 2.9. (a) The factor $\frac{2}{n}$ in front of the log-likelihood in (2.44) is used only for mathematical convenience and can be replaced by $\frac{1}{n}$, by 2 or by 1, as is done in different parts of the literature. This does just correspond to a reparametrization of $\lambda > 0$ and makes no methodological difference (see e.g. Hastie et al., 2015, p. 9).

(b) For simplicity we assume that for GLMs with a dispersion parameter ψ, this parameter is known. Alternatively, in the Lasso optimization problem (2.44) one can also set the dispersion parameter to some arbitrary fixed value $\psi = \psi_0$ (so that $f(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{\beta}, \mu) = f(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{\beta}, \mu, \psi_0)$), leaving the Lasso estimator unchanged. Without further notice, these comments apply also to similar regularization methods that are discussed below. We refer to Fan et al. (2012) and Reid et al. (2016) for discussions of different estimators of the variance $\psi = \sigma^2$ in high-dimensional normal linear models. Further theoretical investigations of the unknown variance case can for example be found in Baraud et al. (2009), Giraud et al. (2012) and Chrétien and Darses (2014).

(c) Usually, the observed values of each explanatory variable X_j are assumed to have been standardized before the application of the Lasso, so that they have mean zero and standard deviation one (i.e. $\sum_{i=1}^{n} X_{i,j} = 0$ and $\frac{1}{n} \sum_{i=1}^{n} X_{i,j}^2 = 1$ for all $j \in \mathcal{P}$). This standardization ensures that the Lasso penalty is independent of the measurement scale for the individual explanatory variables.

The solutions $\hat{\boldsymbol{\beta}}(\lambda)$ of the Lasso are nonlinear in the data and there is in general no closed form analytic expression of the solutions even for normal linear models (see e.g. Friedman et al., 2009, p. 68). However, a main advantage of the Lasso optimization problem (2.44) in comparison to ℓ_0-type optimization problems is that it is a convex problem, since the ℓ_1-norm is convex and the negative log-likelihood in a GLM (with known dispersion parameter and canonical link function) is convex, too (see e.g. Park and Hastie, 2007). Therefore, very

efficient optimization algorithms can be used.

In normal linear models one can apply a modified version of the Least Angle Regression (LARS) algorithm proposed by Efron et al. (2004) to compute the Lasso estimator $\hat{\boldsymbol{\beta}}(\lambda)$ for all possible choices of the penalty parameter $\lambda > 0$, i.e. it is feasible to compute the whole regularization path, which is piecewise linear in this case. However, for different GLMs the solution path of the Lasso is not piecewise linear in general, so that one has to use alternative algorithms that compute the Lasso estimator $\hat{\boldsymbol{\beta}}(\lambda)$ for $\lambda \in \Lambda$, where Λ is a finite grid of pre-specified penalty parameter values. Coordinate descent algorithms (see e.g. Friedman et al., 2007 for the framework of GLMs) are very efficient first-order methods which update the current solution by sequentially minimizing along single coordinate axes. Even for normal linear models, these methods can be much faster than path-following algorithms when applied in sparse high-dimensional settings (see e.g. Bühlmann and van de Geer, 2011, p. 38). We refer to Hastie et al. (2015, Chapter 5) for a nice and detailed discussion of different convex optimization algorithms with a special focus on the Lasso.

Remark 2.10. By convex duality theory (see e.g. Bertsekas, 1999, Chapter 5.3), solving the Lasso with penalty parameter $\lambda > 0$ is equivalent to solving

$$\hat{\mu}(\lambda), \hat{\boldsymbol{\beta}}(\lambda) = \underset{\mu \in \mathbb{R},\, \boldsymbol{\beta} \in \mathbb{R}^p}{\arg\max} \log f(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{\beta}, \mu) \quad \text{w.r.t. } \|\boldsymbol{\beta}\|_1 \leq R(\lambda)\,, \tag{2.45}$$

with a (data dependent) one-to-one correspondence between λ and $R(\lambda)$.

The larger the penalty parameter $\lambda > 0$ is chosen (i.e. the less the radius $R(\lambda)$ of the corresponding ℓ_1-ball), the greater is the amount of shrinkage of the estimated Lasso coefficients $\hat{\boldsymbol{\beta}}(\lambda)$ towards zero. If we choose $\lambda \leq \lambda_0$ with $R(\lambda_0) = \sum_{j=1}^{p} |\hat{\beta}_j|$, where $\hat{\beta}_j$ denotes the j-th component of the MLE under the full model (provided that the MLE exists), then the Lasso estimator coincides with the usual MLE. Now, the characteristic feature of the Lasso is that if the penalty parameter λ is gradually increased from λ_0 to infinity, eventually more and more components of the Lasso estimator $\hat{\boldsymbol{\beta}}(\lambda)$ become **exactly** zero. This means that by solving the ℓ_1-type optimization problem, one encourages sparsity of the solution and therefore implicitly applies some kind of variable selection.

Definition 2.17. The Lasso with penalty parameter $\lambda > 0$ "estimates" the underlying true

active set $S_0 = \{j \in \mathcal{P};\ \beta_{0,j} \neq 0\}$ by

$$\hat{S}(\lambda) = \{j \in \mathcal{P};\ \hat{\beta}_j(\lambda) \neq 0\}. \tag{2.46}$$

Intuitively, the sparsity of the Lasso solution can be explained by considering the geometry of balls with equal ℓ_1-norm, which have a diamond shape with sharp edges on the coordinate axes; therefore, the contours of the negative log-likelihood often intersect with the ℓ_1-balls in some of the edges for the first time (see e.g. Tibshirani, 1996, Figure 2 for a two-dimensional illustration in normal linear models with elliptical contours of the negative log-likelihood). Furthermore, note that although the function $\boldsymbol{\beta} \mapsto \|\boldsymbol{\beta}\|_1$ is continuous and convex, it is not differentiable in points $\boldsymbol{\beta} \in \mathbb{R}^p$ with $\beta_j = 0$ for some $j \in \mathcal{P}$.

An important issue concerning the Lasso is the choice of the penalty parameter $\lambda > 0$, which controls the sparsity of the resulting Lasso estimator. One possible approach is to use cross-validation (CV) on a finite grid of penalty parameter values in order to choose the value λ which minimizes the CV error (see e.g. Hastie et al., 2015, p. 13 for details). By this most commonly used procedure one aims at finding the model which yields the best predictive performance. However, it turns out that with the optimal choice of λ with respect to prediction, the Lasso tends to select a non-negligible number of "noise" variables and thus does not yield variable selection consistency (see e.g. Bühlmann et al., 2010, p. 17).

Therefore, another popular approach is to use certain information criteria in order to tune the penalty parameter λ, for example generalized information criteria like the BIC or the extended BIC (compare Section 2.2). Wang et al. (2007), Wang et al. (2009), Zhang et al. (2010) and Fan and Tang (2013) consider different generalized information criteria for the selection of the penalty parameter in different regularization methods including the Lasso (and the SCAD, see Section 2.5) with the aim of obtaining variable selection consistent procedures. There is, however, no general guarantee that the (globally) best model according to the criterion used or even the true underlying model is among the models computed along the regularization path. In simulation studies one often observes a significant difference between the best criterion models and the models selected by the Lasso in this way (see also the simulation results in Section 5.1).

Turning to another issue, it is natural to ask the question, why the ℓ_1-norm should be used for regularization and not any other norm. And indeed, one may use any ℓ_q-norm with

$q \geq 0$ as a penalty term, which is generally referred to as Bridge Regression (see Frank and Friedman, 1993, Fu, 1998 and Knight and Fu, 2000).

Definition 2.18 (Frank and Friedman, 1993)**.** For $q \geq 0$ any optimization problem of the form

$$\hat{\mu}_q(\lambda), \hat{\boldsymbol{\beta}}_q(\lambda) = \underset{\mu \in \mathbb{R},\, \beta \in \mathbb{R}^p}{\arg\min} \left(-\frac{2}{n} \log\left(f(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{\beta}, \mu) \right) + \lambda \sum_{j=1}^{p} |\beta_j|^q \right) \tag{2.47}$$

with penalty parameter $\lambda > 0$ is called *Bridge Regression*.

For $q = 2$, (2.47) is called *Ridge Regression*. Note that for $q = 1$ Bridge Regression coincides with the Lasso (i.e. $\hat{\mu}_1(\lambda) \equiv \hat{\mu}(\lambda)$ and $\hat{\beta}_1(\lambda) \equiv \hat{\beta}(\lambda)$), and for $q = 0$ it coincides with best subset selection (where $|\beta_j|^0 := \mathbb{1}_{\{\beta_j \neq 0\}}$, since $|\beta_j|^q \to \mathbb{1}_{\{\beta_j \neq 0\}}$ as $q \to 0$).

The main reason why the ℓ_1-norm takes a special role in comparison to other possible forms of the penalty is the following: For $q < 1$ the optimization problem (2.47) is not convex, while for $q \geq 1$ it is convex. However, only for $q \leq 1$ does the optimization problem (2.47) encourage sparse solutions (see e.g. Knight and Fu, 2000). Therefore, the choice $q = 1$ for Bridge Selection is the only possibility to share the two desirable features of convexity and sparse solutions. In particular, the Lasso is "the closest convex relaxation of the best-subset selection problem" (Hastie et al., 2015, p. 23).

Remark 2.11. Bridge Regression can generally be interpreted as maximum a-posteriori (MAP) estimation in a Bayesian setting with an appropriate prior on the regression coefficients $\boldsymbol{\beta} \in \mathbb{R}^p$ (compare e.g. Fu, 1998, Section 7): Consider the improper prior $\pi(\mu) \propto 1$ for the intercept $\mu \in \mathbb{R}$ and suppose that we use an independent prior for each regression coefficient $\beta_j \in \mathbb{R}$, $j \in \mathcal{P}$ (independent of μ) with density

$$\pi_{\lambda,\, q}(\beta_j) = \frac{q \lambda^{1/q}}{2^{1+1/q} \cdot \Gamma(1/q)} \exp\left(-\frac{1}{2} \lambda \, |\beta_j|^q \right) \,, \;\; \beta_j \in \mathbb{R} \,, \tag{2.48}$$

where $q > 0$ and $\lambda > 0$ are assumed to be fixed and $\Gamma(x)$ denotes the value of the gamma function in $x > 0$. Furthermore, for simplicity we assume that the design matrix $\boldsymbol{X}$ is fixed (i.e. not random). Then the density of the posterior distribution of $(\mu, \boldsymbol{\beta})$ given the observed data $(\boldsymbol{X}, \boldsymbol{y})$ is proportional to

$$\begin{aligned} \pi(\mu, \boldsymbol{\beta} \,|\, \boldsymbol{X}, \boldsymbol{y}) &\propto f(\boldsymbol{y} \,|\, \boldsymbol{X}, \mu, \boldsymbol{\beta}) \pi(\mu) \prod_{j \in \mathcal{P}} \pi_{\lambda,\, q}(\beta_j) \\ &\propto \exp\left(\log(f(\boldsymbol{y} \,|\, \boldsymbol{X}, \mu, \boldsymbol{\beta})) - \frac{1}{2} \lambda \sum_{j=1}^{p} |\beta_j|^q \right) . \end{aligned}$$

Therefore, maximizing the posterior density $\pi(\mu, \boldsymbol{\beta} \,|\, \boldsymbol{X}, \boldsymbol{y})$ is equivalent to minimizing the corresponding (re-parametrized version of) Bridge Regression:

$$\begin{aligned}\underset{\mu\in\mathbb{R},\, \boldsymbol{\beta}\in\mathbb{R}^p}{\arg\max}\, \pi(\mu, \boldsymbol{\beta} \,|\, \boldsymbol{X}, \boldsymbol{y}) &= \underset{\mu\in\mathbb{R},\, \boldsymbol{\beta}\in\mathbb{R}^p}{\arg\max} \left(\log(f(\boldsymbol{y} \,|\, \boldsymbol{X}, \mu, \boldsymbol{\beta})) - \frac{1}{2}\lambda \sum_{j=1}^{p} |\beta_j|^q \right) \\ &= \underset{\mu\in\mathbb{R},\, \boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min} \left(-2\log(f(\boldsymbol{y} \,|\, \boldsymbol{X}, \mu, \boldsymbol{\beta})) + \lambda \sum_{j=1}^{p} |\beta_j|^q \right) .\end{aligned}$$

Note that this does correspond to the form of Bridge Regression given in (2.47) only if we consider the re-parametrization $\lambda \to \frac{\lambda}{n}$, i.e. consider $\pi_{\lambda/n,\, q}(\beta_j)$ instead of $\pi_{\lambda,\, q}(\beta_j)$. However, in the Bayesian context it is preferred to choose a prior distribution for $\boldsymbol{\beta}$ which does not depend on the sample size n of the observed data; therefore, in this setting we use the alternative parametrization (with the factor 2 in front of the log-likelihood).

Another remark is that the prior for Ridge Regression with $q = 2$ is simply a normal distribution with zero mean and variance λ^{-1}; the prior corresponding to the Lasso with $q = 1$ is a Laplace distribution (also called double exponential distribution) with zero mean and scale parameter $b = 2\lambda^{-1}$, i.e.

$$\pi_{\lambda,1}(\beta_j) = \frac{1}{2b} \exp\left(-\frac{|\beta_j|}{b} \right) , \ \beta_j \in \mathbb{R} . \tag{2.49}$$

Note that when imposing a Laplace prior, only the posterior mode of $\pi(\mu, \boldsymbol{\beta} \,|\, \boldsymbol{X}, \boldsymbol{y})$ coincides with the usual Lasso estimator, whereas the posterior mean and the posterior median (being more common estimators in the Bayesian setting) do not yield sparse estimators. We refer to Park and Casella (2008), Hans (2009), Hans (2010) and Lykou and Ntzoufras (2013) for fully Bayesian treatments of the Lasso.

Furthermore, note that best subset selection with the choice $q = 0$ for Bridge Regression in (2.47) does **not** correspond to the choice $q = 0$ in (2.48). In order to obtain a Bayesian analogue to best subset selection one has to use a prior which puts some point mass at 0 for each regression coefficient β_j, $j \in \mathcal{P}$. We refer to Liang et al. (2013) for a Bayesian subset modelling approach where finding the MAP model is asymptotically equivalent to minimizing the EBIC.

2.4.2. Theoretical properties of the Lasso in normal linear models

In this subsection we want to survey theoretical results for the Lasso which have been obtained for prediction, estimation and variable selection. Here, we state the results in detail for the important class of normal linear models, which allows one to focus on the essentials without imposing more technical conditions needed for the general GLM setting. For ease of presentation, in this subsection we suppose that the data has been standardized, so that we do not include an intercept in the model.

Even though the solutions to the Lasso are not given in closed form in general, they can be computed explicitly in the special case of an orthonormal design in normal linear models. It is insightful to compare the Lasso with best subset selection in this particular situation.

Remark 2.12. Consider a normal linear model with $p = n$ and $\frac{1}{n}\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{I}_p$. Then the MLE (i.e. the least-squares estimator) of $\boldsymbol{\beta} \in \mathbb{R}^p$ is given by $\hat{\boldsymbol{\beta}}_{\mathrm{ML}} = (\hat{\beta}_{\mathrm{ML},1}, \ldots, \hat{\beta}_{\mathrm{ML},p})^T = \frac{1}{n}\boldsymbol{X}^T\boldsymbol{Y}$. Furthermore, in this situation we obtain (compare e.g. Bühlmann and van de Geer, 2011, p. 10f. and Hastie et al., 2015, p. 22, Table 2.3):

(a) The Lasso estimator $\hat{\boldsymbol{\beta}}(\lambda) = (\hat{\beta}_1(\lambda), \ldots, \hat{\beta}_p(\lambda))^T \in \mathbb{R}^p$ is given by the *soft-threshold estimator*

$$\hat{\beta}_j(\lambda) = \mathrm{sgn}(\hat{\beta}_{\mathrm{ML},j}) \left(|\hat{\beta}_{\mathrm{ML},j}| - \frac{\lambda}{2} \right)_+, \quad j = 1, \ldots, p, \tag{2.50}$$

where $\mathrm{sgn}(x) = \mathbb{1}_{(0,\infty)}(x) - \mathbb{1}_{(-\infty,0)}(x)$ denotes the sign of $x \in \mathbb{R}$ and $(x)_+ = \max(x, 0)$ denotes the "positive part" of $x \in \mathbb{R}$.

(b) The estimator $\hat{\boldsymbol{\beta}}_0(\lambda) = (\hat{\beta}_{0,1}(\lambda), \ldots, \hat{\beta}_{0,p}(\lambda))^T \in \mathbb{R}^p$ corresponding to best subset selection (with $q = 0$ in (2.47)) is given by the *hard-threshold estimator*

$$\hat{\beta}_{0,j}(\lambda) = \hat{\beta}_{\mathrm{ML},j} \cdot \mathbb{1}\{|\hat{\beta}_{\mathrm{ML},j}| > \sqrt{\lambda}\}, \quad j = 1, \ldots, p. \tag{2.51}$$

By comparing the Lasso solution and the solution of best subset selection in the orthonormal design situation of Remark 2.12 it becomes clear that, in contrast to the ℓ_0-norm, the ℓ_1-norm shrinks even those coefficients towards zero which are estimated to be non-zero; therefore it produces biased estimates (with the bias being increasing in the penalty parameter λ). This comes with the benefit of a reduced variance of the estimators, which can be beneficial for prediction. However, Su et al. (2017) point out that the possibly large bias induced by the Lasso implies that a significant part of the effects of the true explanatory

variables ("signal variables") on the response may remain unexplained. Hence, when some of the coefficients corresponding to "signal variables" are estimated to be nonzero in the Lasso solution, the unexplained part which is left in the residuals due to shrinkage may lead to the undesirable inclusion of additional "noise variables" by the Lasso. This may serve as a first intuitive explanation for why the Lasso has certain problems for variable selection when the aim is the identification of the true underlying model. The qualitative reasoning will be made more precise in the following theorems.

For different consistency results concerning the Lasso we need to impose certain conditions, which we define next (compare Bühlmann and van de Geer, 2011, p. 22 and p. 106).

Definition 2.19 (Bühlmann and van de Geer, 2011)**.** Consider a given dataset $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$ with $\boldsymbol{X} \in \mathbb{R}^{n \times p}$. Let $\hat{\boldsymbol{\Sigma}} := \frac{1}{n}\boldsymbol{X}^T\boldsymbol{X}$ and w.l.o.g. suppose that the true active set is given by

$$S_0 = \{j \in \mathcal{P};\ \beta_{0,j} \neq 0\} = \{1, \ldots, s_0\}\,. \tag{2.52}$$

Furthermore, partition $\hat{\boldsymbol{\Sigma}} \in \mathbb{R}^{p \times p}$ such that $\hat{\boldsymbol{\Sigma}} = \begin{bmatrix} \hat{\boldsymbol{\Sigma}}_{1,1} & \hat{\boldsymbol{\Sigma}}_{1,2} \\ \hat{\boldsymbol{\Sigma}}_{2,1} & \hat{\boldsymbol{\Sigma}}_{2,2} \end{bmatrix}$, where $\hat{\boldsymbol{\Sigma}}_{1,1} \in \mathbb{R}^{s_0 \times s_0}$.

(a) The *compatibility condition* is satisfied for the set S_0 with *compatibility constant* $\phi_0^2 > 0$ iff for all $\boldsymbol{\beta} \in \mathbb{R}^p$ with $\|\boldsymbol{\beta}_{S_0^c}\|_1 \leq 3\|\boldsymbol{\beta}_{S_0}\|_1$, we have

$$\boldsymbol{\beta}^T\hat{\boldsymbol{\Sigma}}\boldsymbol{\beta} \geq \frac{\phi_0^2}{s_0}\|\boldsymbol{\beta}_{S_0}\|_1^2\,. \tag{2.53}$$

(b) The *strong irrepresentable condition* is satisfied iff $\hat{\boldsymbol{\Sigma}}_{1,1}$ is invertible and it holds that

$$\|\hat{\boldsymbol{\Sigma}}_{2,1}\hat{\boldsymbol{\Sigma}}_{1,1}^{-1}\mathrm{sgn}(\beta_{0,1}, \ldots, \beta_{0,s_0})\|_\infty \leq \theta \text{ for some } 0 < \theta < 1\,, \tag{2.54}$$

where $\|\boldsymbol{x}\|_\infty = \max_j |x_j|$ and $\mathrm{sgn}(x_1, \ldots, x_q) = (\mathrm{sgn}(x_1), \ldots, \mathrm{sgn}(x_q))^T \in \mathbb{R}^q$ for any $\boldsymbol{x} = (x_1, \ldots, x_q)^T \in \mathbb{R}^q$.

(c) The *weak irrepresentable condition* is satisfied iff $\hat{\boldsymbol{\Sigma}}_{1,1}$ is invertible and it holds that

$$\|\hat{\boldsymbol{\Sigma}}_{2,1}\hat{\boldsymbol{\Sigma}}_{1,1}^{-1}\mathrm{sgn}(\beta_{0,1}, \ldots, \beta_{0,s_0})\|_\infty \leq 1\,. \tag{2.55}$$

Before we proceed with the theoretical properties of the Lasso, we try to give some intuitive interpretations of these conditions. The compatibility condition requires that for possible estimated regression vectors $\boldsymbol{\beta} \in \mathbb{R}^p$ with much "ℓ_1-mass" concentrated on the true active set

S_0 (i.e. for $\boldsymbol{\beta}$ with $\|\boldsymbol{\beta}_{S_0^c}\|_1 \leq 3\|\boldsymbol{\beta}_{S_0}\|_1$), the ℓ_2-norm of the corresponding estimated response $\boldsymbol{\beta}^T\hat{\boldsymbol{\Sigma}}\boldsymbol{\beta} = \frac{1}{n}\|\boldsymbol{X}\boldsymbol{\beta}\|_2^2$ is sufficiently large, i.e. it is bounded from below by a constant times the (squared) ℓ_1-norm of $\boldsymbol{\beta}_{S_0}$. This has also motivated the name of this condition, since it addresses the compatibility between the ℓ_1- and ℓ_2-norm in a certain sense (van de Geer and Bühlmann, 2009, p. 1363). On the rough level, the compatibility condition is satisfied if there does not exist too large (absolute) correlations among the "signal" variables (i.e. variables in S_0).

The irrepresentable conditions require even more: The correlations between the "signal" and the "noise" variables should not be too large (in absolute value). In order to understand this interpretation, note that

$$\hat{\boldsymbol{\Sigma}}_{2,1}\hat{\boldsymbol{\Sigma}}_{1,1}^{-1} = \left(\boldsymbol{X}_{S_0^c}^T\boldsymbol{X}_{S_0}\right)\left(\boldsymbol{X}_{S_0}^T\boldsymbol{X}_{S_0}\right)^{-1}$$

and since (2.54) has to be satisfied for all possible signs of $\boldsymbol{\beta}_{0,S_0}$, we particularly require

$$\|\left(\boldsymbol{X}_{S_0}^T\boldsymbol{X}_{S_0}\right)^{-1}\boldsymbol{X}_{S_0}^T\boldsymbol{X}_j\|_1 < \theta \text{ , with } 0 < \theta < 1\,,$$

to hold for all $j \in \mathcal{P}\backslash S_0$ (compare Zhao and Yu, 2006). This means that the estimated regression coefficients when regressing a "noise" variable X_j on the "signal" variables $X_1, \ldots, X_{s_0}$ should be small, i.e. X_j should not be highly correlated with $X_1, \ldots, X_{s_0}$. The strong and weak irrepresentable conditions are almost identical with the only difference that the strong irrepresentable condition requires the components of $\hat{\boldsymbol{\Sigma}}_{2,1}\hat{\boldsymbol{\Sigma}}_{1,1}^{-1}\text{sgn}(\beta_{0,1}, \ldots, \beta_{0,s_0})$ to be bounded strictly away from 1 in the limit $n \to \infty$. This is basically a technicality needed for showing that the strong irrepresentable condition is sufficient for variable selection consistency of the Lasso (compare Theorem 2.4). It can be shown that the strong irrepresentable condition implies the compatibility condition (see van de Geer and Bühlmann, 2009). Finally, note that neither the compatibility condition nor the irrepresentable conditions can be checked in practice, since the true active set S_0 is unknown.

We now state a theorem with, on the first sight, quite "positive" results regarding the Lasso for (fixed design) prediction and estimation, which can for example be found in Bühlmann and van de Geer (2011, Section 2.4.2).

Theorem 2.3 (Bühlmann and van de Geer, 2011)**.** *Consider normal linear models for standardized data with an asymptotic setting as in Notation 2.7. Let* $\hat{\boldsymbol{\beta}}^{(n)}(\lambda_n)$ *denote the corre-*

sponding Lasso estimator with penalty parameter $\lambda_n > 0$ for data with sample size n. Suppose that we have $\|\boldsymbol{\beta}_0^{(n)}\|_1 = o\left(\sqrt{\frac{n}{\log(p_n)}}\right)$ and $\lambda_n \asymp \sqrt{\frac{\log(p_n)}{n}}$.

(a) We have

$$\frac{1}{n}\left\|\boldsymbol{X}^{(n)}\hat{\boldsymbol{\beta}}^{(n)}(\lambda_n) - \boldsymbol{X}^{(n)}\boldsymbol{\beta}_0^{(n)}\right\|_2^2 \xrightarrow{P} 0\,, \quad n \to \infty\,. \tag{2.56}$$

(b) If there exists an $N \in \mathbb{N}$ such that for all $n \geq N$ the compatibility condition is satisfied with compatibility constant ϕ_n^2, then we have

$$E\left[\frac{1}{n}\left\|\boldsymbol{X}^{(n)}\hat{\boldsymbol{\beta}}^{(n)}(\lambda_n) - \boldsymbol{X}^{(n)}\boldsymbol{\beta}_0^{(n)}\right\|_2^2\right] = \mathcal{O}\left(\frac{s_0^{(n)}}{\phi_n^2} \cdot \frac{\log(p_n)}{n}\right) \tag{2.57}$$

and we have

$$\left\|\hat{\boldsymbol{\beta}}^{(n)}(\lambda_n) - \boldsymbol{\beta}_0^{(n)}\right\|_1 \xrightarrow{P} 0\,, \quad n \to \infty\,. \tag{2.58}$$

Note that for ease of presentation we consider an asymptotic analysis in Theorem 2.3, but one can also obtain explicit finite sample results (see e.g. Bühlmann and van de Geer, 2011, Corollary 6.1 and Corollary 6.2 in Chapter 6).

The result in part (a) of Theorem 2.3 says that if the "sparsity measure" $\|\boldsymbol{\beta}_0^{(n)}\|_1$ grows "slower" than $a_n = \sqrt{\frac{n}{\log(p_n)}}$ and the penalty parameter λ_n is chosen in the order of a_n^{-1}, then the Lasso is consistent for prediction under no additional assumption on the design matrix $\boldsymbol{X}^{(n)}$ or on the true vector of coefficients $\boldsymbol{\beta}_0^{(n)}$. However, if we want to obtain a fast decay of the prediction error of order $\mathcal{O}(\log(p_n)/n)$, then we additionally have to assume that the compatibility condition is satisfied (see part (b) of Theorem 2.3).

Note that in a sparse high-dimensional linear model setting it can be shown that for ℓ_0-type estimators (compare equation (2.39) in Section 2.3.1) we obtain a fast decay of the prediction error of order $\mathcal{O}(\log(p_n)/n)$ without any restrictions on the design matrix (see e.g. Raskutti et al., 2011). In contrast, Zhang et al. (2017) show that in a sparse high-dimensional situation with $p > n$, without imposing assumptions like the compatibility condition, the prediction error for the Lasso decays with a rate slower than $o(\sqrt{\log(n)/n})$ for certain families of design matrices $\boldsymbol{X}^{(n)}$. This indicates that, even when the focus is solely on "optimal predictions", there can be a fundamental gap between the performance of ℓ_0-type and ℓ_1-type methods. Furthermore, we want to note that Zhang et al. (2017) have also derived similar quite negative results regarding the rate decay of the prediction error for other regularization methods like Ridge Regression as well as non-convex methods like the SCAD (compare Section 2.5).

Part (b) of Theorem 2.3 implies that, under the compatibility condition, we also obtain the consistency of the Lasso for estimating the unknown vector of coefficients $\boldsymbol{\beta}_0^{(n)}$ with respect to the ℓ_1-norm (see equation (2.58)). Under a slightly stronger assumption than the compatibility condition, the so-called *restricted eigenvalue condition*, one can derive the consistency of the Lasso for estimation with respect to the ℓ_2-norm (Meinshausen and Yu, 2009), i.e.

$$\left\| \hat{\boldsymbol{\beta}}^{(n)}(\lambda_n) - \boldsymbol{\beta}_0^{(n)} \right\|_2 \xrightarrow{\mathrm{P}} 0\,, \quad n \to \infty\,. \tag{2.59}$$

A drawback of ℓ_1-regularization methods like the Lasso is that they typically require quite strong conditions on the design matrix $\boldsymbol{X}$ to be variable selection consistent. For the Lasso in linear regression models it has been shown that the design matrix $\boldsymbol{X}$ has to satisfy restrictive irrepresentable conditions (compare Definition 2.19) to obtain variable selection consistency, even in the classical asymptotic setting with a fixed number of explanatory variables (Meinshausen and Bühlmann, 2006; Zhao and Yu, 2006). Note that the asymptotic identifiability assumption (see Definition 2.14) needed for the consistency of the EBIC is significantly weaker than the so-called *sparse Riesz condition* (see Chen and Chen, 2008) and, from a practical point of view, the sparse Riesz condition is in turn less restrictive than the irrepresentable conditions needed for the consistency of the Lasso (compare the discussion in Zhang and Huang, 2008, Section 6). This comparison between the EBIC and the Lasso is an important example for the general observation that ℓ_0-type methods provide variable selection consistency under weaker theoretical conditions than ℓ_1-type methods.

Due to its importance we want to state the result concerning the variable selection consistency of the Lasso in some more detail. For this, we consider an alternative notion of consistency, which is slightly stronger than usual variable selection consistency (see Definition 2.8) in the sense that we also require the signs of the non-zero estimated coefficients to coincide with the signs of the true underlying coefficients. This alternative notion is primarily used for technical reasons in the proof of the following theorem (the proof is not presented here, see Zhao and Yu, 2006). However, it is generally desirable that also this form of consistency holds since an estimated model with reversed signs might lead to highly misleading interpretations of the effects (compare Zhao and Yu, 2006, p. 2543).

Definition 2.20 (Zhao and Yu, 2006)**.** Consider an estimator $\hat{\boldsymbol{\beta}}^{(n)}$ of the vector of regression coefficients $\boldsymbol{\beta}_0^{(n)}$ in an asymptotic setting as in Notation 2.7. Then the estimator $\hat{\boldsymbol{\beta}}^{(n)}$ is called

sign consistent if

$$\lim_{n\to\infty} P\left(\operatorname{sgn}\left(\hat{\boldsymbol{\beta}}^{(n)}\right) = \operatorname{sgn}\left(\boldsymbol{\beta}_0^{(n)}\right)\right) = 1\,. \tag{2.60}$$

Theorem 2.4 (Zhao and Yu, 2006)**.** *Consider normal linear models for standardized data with an asymptotic setting as in Notation 2.7, but assume that the number of explanatory variables $p = p_n$, as well as the true coefficient vector $\boldsymbol{\beta}_0 = \boldsymbol{\beta}_0^{(n)}$ with active set $S_0 = S_0^{(n)}$ are fixed (i.e. they do not depend on the sample size n). Furthermore, assume that*

$$\boldsymbol{\Sigma}^{(n)} = \frac{1}{n}\left(\boldsymbol{X}^{(n)}\right)^T \boldsymbol{X}^{(n)} \to \boldsymbol{\Sigma},\ n \to \infty \tag{2.61}$$

for some positive definite matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{p\times p}$ (where the convergence should be understood componentwise) and that

$$\frac{1}{n}\max_{i=1,\ldots,n} \|\boldsymbol{X}_{i,*}^{(n)}\|_2^2 \to 0,\ n \to \infty\,. \tag{2.62}$$

Then we obtain:

(a) If there exists an $N \in \mathbb{N}$ such that the strong irrepresentable condition is satisfied for all design matrices $\boldsymbol{X}^{(n)}$ with $n \geq N$ and if the penalty parameter λ_n satisfies $\lambda_n/n \to 0$ and $\lambda_n/n^{\frac{1+c}{2}} \to \infty$ as $n \to \infty$ for some constant $0 \leq c < 1$, then it holds

$$P\left(sgn\left(\hat{\boldsymbol{\beta}}^{(n)}(\lambda_n)\right) = sgn\left(\boldsymbol{\beta}_0\right)\right) = 1 - o(\exp(-n^c))\,. \tag{2.63}$$

In particular, under the strong irrepresentable condition the Lasso with penalty parameter λ_n is sign consistent.

(b) If the Lasso is sign consistent, then there exists an $N \in \mathbb{N}$ such that the weak irrepresentable condition holds for all design matrices $\boldsymbol{X}^{(n)}$ with $n \geq N$.

Theorem 2.4 shows that in order to obtain variable selection consistency (or more precisely sign consistency) with the Lasso, the weak irrepresentable condition has to hold. This means that the strong irrepresentable condition is sufficient and "almost" necessary for variable selection consistency, even in the classical asymptotic setting. Similar results also hold in an asymptotic setting with a diverging number of explanatory variables p_n and relevant explanatory variables $s_0^{(n)}$ (Zhao and Yu, 2006) as well as for random Gaussian designs (Meinshausen and Bühlmann, 2006).

We briefly summarize and interpret the theoretical results for the Lasso stated in Theorems 2.3 and 2.4 (compare the discussion in Bühlmann, 2013b, p. 4): The Lasso yields consistent

prediction under essentially no additional assumptions (Theorem 2.3 (a)). If one wants to achieve a fast decay of the prediction error or consistency for estimation, one has to impose the compatibility condition on the design (Theorem 2.3 (b)). However, for variable selection consistency one has to assume certain irrepresentable conditions (Theorem 2.4), which are very "unlikely" to hold in a high-dimensional practical situation, since typically some explanatory variables have high empirical correlations (compare e.g. Fan et al., 2012). Therefore, the overall message is that ℓ_1-type methods should be used with caution if the aim is the identification of the true underlying model and ℓ_0-type methods are likely to be preferred since they provide variable selection consistency under weaker conditions (compare e.g. Theorem 2.2 for the EBIC).

Remark 2.13. Similar consistency results as in Theorem 2.3 and Theorem 2.4 have also been obtained for the Lasso in the high-dimensional GLM framework, see e.g. van de Geer (2008) and van de Geer and Müller (2012). Even though the involved conditions are mathematically more technical, the main "messages" remain essentially the same as in the simpler case of normal linear models. In particular, van de Geer and Müller (2012) show that the Lasso is variable selection consistent under similar restrictive conditions which involve a generalized version of the irrepresentable condition.

Remark 2.14. Finally, we want to note that in this subsection — as in most parts of the classical literature on the Lasso — the focus has been on the theoretical properties of the Lasso from the perspective of point estimation. It turns out to be a very challenging problem to construct valid confidence intervals for the unknown regression coefficients as well as p-values for conducting hypothesis tests in high-dimensional models. However, quite a few different approaches have recently been proposed for valid inference with regularization methods like the Lasso. Important references include Wasserman and Roeder (2009), Meinshausen et al. (2009), Bühlmann (2013b), Lockhart et al. (2014), van de Geer et al. (2014), Javanmard and Montanari (2014) and Lee et al. (2016). A recent review and comparison of different high-dimensional inference methods can be found in Dezeure et al. (2015).

2.4.3. Variants of the Lasso

Many modifications of the Lasso have been proposed which try to compensate for different possible deficiencies of the "vanilla Lasso" given in Definition 2.16. One such issue is that

if there exist groups of highly correlated explanatory variables which contribute similarly to the response, then the Lasso tends to select only one variable from each of these groups. In contrast, the ℓ_2-type regularization in Ridge Regression leads to estimated coefficients of the highly-correlated variables that are all non-zero. The Elastic Net, proposed by Zou (2006), makes use of this property by combining ℓ_1-type and ℓ_2-type penalties.

Definition 2.21 (Zou, 2006)**.** The *Elastic Net* with penalty parameter $\lambda > 0$ and mixing parameter $\alpha \in [0, 1]$ is given by the optimization problem

$$\hat{\mu}(\lambda, \alpha), \hat{\boldsymbol{\beta}}(\lambda, \alpha) = \underset{\mu \in \mathbb{R},\, \boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \left(-\frac{2}{n} \log \left(f(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\beta}, \mu) \right) + \lambda \left(\alpha \|\boldsymbol{\beta}\|_1 + \frac{1}{2}(1 - \alpha) \|\boldsymbol{\beta}\|_2^2 \right) \right) . \tag{2.64}$$

If the mixing parameter is chosen to be $\alpha = 1$, then the Elastic Net reduces to the usual Lasso, while for the choice $\alpha = 0$ it reduces to (a rescaled version of) Ridge Regression. The factor $\frac{1}{2}$ in front of the squared ℓ_2-norm in (2.64) is typically chosen just for mathematical convenience (see e.g. Hastie et al., 2015, p. 57) and so the penalty of the Elastic Net can in principle be viewed as a convex combination of the ℓ_1- and ℓ_2-norm. For mixing parameters $\alpha \in (0, 1)$, the penalty induced by the Elastic Net shares desirable features of ℓ_1- and ℓ_2-type penalties so that highly correlated variables tend to be selected in groups while still enforcing sparsity of the solution. This important property distinguishes the Elastic Net from Bridge Regression with $q \in (1, 2)$, which does not yield sparse solutions. The additional mixing parameter $\alpha \in [0, 1]$ of the Elastic Net can either be chosen on a subjective basis or by cross-validation using a finite grid of mixing parameter values (see e.g. Hastie et al., 2015, p. 57).

If a particular group structure between the explanatory variables is known a-priori, then it is desirable to make explicit use of this structure in the estimation process. If for example a categorical variable (factor) with more than two levels is modelled with several binary "dummy" variables, then it might be desirable that all the corresponding estimated coefficients should either be zero or all of them non-zero. Yuan and Lin (2006) propose the Group Lasso which accounts for such group structures by using an ℓ_2-type penalty for the coefficients inside the groups and combining these penalties with an ℓ_1-type penalty between the groups.

Definition 2.22 (Yuan and Lin, 2006)**.** Suppose that $G_1, \ldots, G_b$ is a known partition of

$\mathcal{P} = \{1, \ldots, p\}$, i.e. $\cup_{r=1}^{b} G_r = \mathcal{P}$ and $G_r \cap G_{r'} = \emptyset$ for $r \neq r'$. Then the corresponding *Group Lasso* with penalty parameter $\lambda > 0$ is given by the optimization problem

$$\hat{\mu}(\lambda), \hat{\boldsymbol{\beta}}(\lambda) = \underset{\mu \in \mathbb{R},\, \boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \left(-\frac{2}{n} \log\left(f(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\beta}, \mu)\right) + \lambda \sum_{r=1}^{b} \sqrt{|G_r|} \cdot \|\boldsymbol{\beta}_{G_r}\|_2 \right), \tag{2.65}$$

where $\boldsymbol{\beta}_{G_r} = (\beta_j;\ j \in G_r) \in \mathbb{R}^{|G_r|}$ for $r = 1, \ldots, b$.

It is obvious that if all groups consist of only one variable, i.e. if we have $G_1 = \{1\}, \ldots, G_p = \{p\}$ in Definition 2.22, then the Group Lasso reduces to the usual Lasso. Furthermore, depending on the choice of the penalty parameter $\lambda > 0$, for each $r = 1, \ldots, b$ we either have $\hat{\beta}_j(\lambda) = 0$ for all $j \in G_r$ or we have $\hat{\beta}_j(\lambda) \neq 0$ for all $j \in G_r$ (see Yuan and Lin, 2006); thus the Group Lasso has the desirable property that variables in the same group are either all selected to be in the model or not.

Another problem of the "vanilla" version of the Lasso is that it requires very strong conditions for variable selection consistency (compare Theorem 2.4). The so-called Adaptive Lasso has been proposed by Zou (2006) in order to provide variable selection consistency under weaker conditions.

Definition 2.23 (Zou, 2006)**.** Let $\hat{\boldsymbol{\beta}}_{\text{Init}} \in \mathbb{R}^p$ be some initial estimate of the coefficient vector $\boldsymbol{\beta} \in \mathbb{R}^p$. Then the *Adaptive Lasso* with penalty parameter $\lambda > 0$ is given by the optimization problem

$$\hat{\mu}(\lambda), \hat{\boldsymbol{\beta}}(\lambda) = \underset{\mu \in \mathbb{R},\, \boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \left(-\frac{2}{n} \log\left(f(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\beta}, \mu)\right) + \lambda \sum_{j=1}^{p} \omega_j |\beta_j| \right), \tag{2.66}$$

where the weights are given by $\omega_j = |\hat{\beta}_{\text{Init},j}|^{-1}$ for $j \in \mathcal{P}$ with the convention that $\omega_j = \infty$ if $\hat{\beta}_{\text{Init},j} = 0$ (corresponding to the equality constraint $\hat{\beta}_j(\lambda) = 0$).

The initial estimate $\hat{\boldsymbol{\beta}}_{\text{Init}}$ of $\boldsymbol{\beta} \in \mathbb{R}^p$ can for example be obtained from Ridge Regression or the usual Lasso. If some of the components of the initial estimate are exactly equal to zero (as in the Lasso), then, by construction, the Adaptive Lasso has the property that $\hat{\beta}_{\text{Init},j} = 0$ implies $\hat{\beta}(\lambda) = 0$ for any choice of $\lambda > 0$ (compare e.g. Bühlmann and van de Geer, 2011, p. 25). This means that the Adaptive Lasso produces a sparser model than the model corresponding to the initial estimator, i.e. we have

$$\hat{S}_{\text{Ada}}(\lambda) := \{j \in \mathcal{P};\ \hat{\beta}_{\text{Ada,j}}(\lambda) \neq 0\} \subseteq \{j \in \mathcal{P};\ \hat{\beta}_{\text{Init,j}} \neq 0\} =: \hat{S}_{\text{Init}}\,. \tag{2.67}$$

It can be shown that, under certain conditions and in contrast to the usual Lasso, the

Adaptive Lasso enjoys the so-called *oracle property* (Fan and Li, 2001), which means that the method is variable selection consistent in the sense of Definition 2.8 and that the estimators of the non-zero regression coefficients are asymptotically normal distributed having the same mean and covariance as in a situation where the true active set was known (Zou, 2006; Huang et al., 2008).

Definition 2.24 (Fan and Li, 2001)**.** Consider normal linear models with an asymptotic setting as in Notation 2.7, i.e. the true underlying model (corresponding to the sample size n) is denoted by $S_0^{(n)} = \{j \in \{1, \dots, p_n\};\ \beta_{0,j}^{(n)} \neq 0\}$ with size $s_0^{(n)} = |S_0^{(n)}|$. For $n \in \mathbb{N}$, let $\hat{\boldsymbol{\beta}}^{(n)} \in \mathbb{R}^{p_n}$ be an estimator of $\boldsymbol{\beta}_0^n \in \mathbb{R}^{p_n}$ and let $\hat{S}^{(n)} = \{j \in \{1, \dots, p_n\};\ \hat{\beta}_j^{(n)} \neq 0\}$ denote the corresponding estimated active set.
Then the estimator $\hat{\boldsymbol{\beta}}^{(n)}$ enjoys the *oracle property* for the given asymptotic setting if the following two properties are satisfied:

(i) The method is variable selection consistent (compare Definition 2.8) in the sense that

$$\lim_{n\to\infty} P\left(\hat{S}^{(n)} = S_0^{(n)}\right) = 1\,. \tag{2.68}$$

(ii) We have

$$\sqrt{n}\left(\hat{\boldsymbol{\beta}}^{(n)}_{S_0^{(n)}} - \boldsymbol{\beta}^{(n)}_{0,S_0^{(n)}}\right) \xrightarrow{\mathrm{d}} \mathcal{N}\left(\mathbf{0}\,,\, \sigma^2\left(\boldsymbol{X}^T_{S_0^{(n)}}\boldsymbol{X}_{S_0^{(n)}}\right)^{-1}\right),\ n \to \infty\,. \tag{2.69}$$

Theorem 2.5 (Zou, 2006)**.** *Consider normal linear models for standardized data with an asymptotic setting as in Notation 2.7, but assume that the number of explanatory variables $p = p_n$, as well as the true coefficient vector $\boldsymbol{\beta}_0 = \boldsymbol{\beta}_0^{(n)}$ with active set $S_0 = S_0^{(n)}$ are fixed (i.e. they do not depend on the sample size n). Then we obtain:*

(a) The usual Lasso does not enjoy the oracle property in the sense of Definition 2.24.

(b) Suppose that the initial estimator $\hat{\boldsymbol{\beta}}^{(n)}_{Init}$ used for the Adaptive Lasso is an $\sqrt{n}$-consistent estimator of $\boldsymbol{\beta}_0$, i.e. we have

$$\|\hat{\boldsymbol{\beta}}^{(n)}_{Init} - \boldsymbol{\beta}_0\|_2 = \mathcal{O}_P\left(n^{-\frac{1}{2}}\right)\,. \tag{2.70}$$

If $\frac{\lambda_n}{\sqrt{n}} \to 0$ and $\lambda_n \to \infty$ as $n \to \infty$, then the Adaptive Lasso with penalty parameter $\lambda_n > 0$ enjoys the oracle property in the sense of Definition 2.24.

Theorem 2.5 shows that even in the classical asymptotic scenario with a finite number of possible explanatory variables, the Lasso fails to enjoy the oracle property. The reason for this is that in order to satisfy the desirable limiting property (2.69) concerning the estimation of $\boldsymbol{\beta}_0$, the penalty parameter for the Lasso has to be chosen asymptotically like $\lambda_n \asymp \sqrt{n}$; but this rate leads to the selection of the wrong model with positive probability and therefore to generally inconsistent variable selection (compare Zou, 2006).

Zou (2006) has extended the result of Theorem 2.5 to the framework of generalized linear models, where some mild regularity conditions on the Fisher information matrix have to be imposed. In addition, the oracle property of the Adaptive Lasso has also been established, under some additional conditions, in an asymptotic setting where the number of explanatory variables p_n and the number of "signal variables" $s_0 = |S_0|$ is allowed to grow with the sample size n (Huang et al., 2008). However, Pötscher and Schneider (2009) give a warning concerning the interpretation of such "oracle" results and note that the sampling distribution of the Adaptive Lasso estimator can be highly non-normal even in large sample regimes.

An important assumption for all these "oracle" results is that the initial estimator which is used in the Adaptive Lasso is $\sqrt{n}$-consistent. In the low-dimensional setting considered in Theorem 2.5 one can use the ordinary least-squares estimator which is $\sqrt{n}$-consistent. However, in a high-dimensional asymptotic setting with $p_n > n$, the least-squares estimator cannot be used and one might use the usual Lasso instead (which is in general not $\sqrt{n}$-consistent). It can be shown that if the compatibility condition (see Definition 2.19) is satisfied and if the non-zero regression coefficients are sufficiently large, i.e. if a so-called "beta-min condition"

$$\min_{j \in S_0^{(n)}} |\beta_{0,j}^{(n)}| \geq C |S_0^{(n)}| \sqrt{\log(n)/p_n}\,, \tag{2.71}$$

holds for some constant $C > 0$ (independent of n), then the Adaptive Lasso with the usual Lasso as the initial estimator is variable selection consistent for appropriate choices of the tuning parameters λ_{Init} and λ (see Bühlmann et al., 2010, Section 2.8.3 and van de Geer et al., 2011, Section 3.6). Since the compatibility condition is weaker than the irrepresentable condition (see van de Geer and Bühlmann, 2009), this implies that the Adaptive Lasso requires arguably less restrictive assumptions than the usual Lasso for variable selection consistency.

Another variant similar to the Adaptive Lasso is the so-called *Thresholded Lasso* (van de Geer et al., 2011).

Definition 2.25 (van de Geer et al., 2011)**.** Let $\hat{\boldsymbol{\beta}}_{\text{Init}}(\lambda_{\text{Init}})$ be the usual Lasso estimator with penalty parameter $\lambda_{\text{Init}} > 0$. For some threshold $\lambda_{\text{Thres}} > 0$, let

$$\hat{S}_{\text{Thres}} = \left\{ j \in \mathcal{P};\ \left| \hat{\boldsymbol{\beta}}_{\text{Init},j}(\lambda_{\text{Init}}) \right| > \lambda_{\text{Thres}} \right\} . \tag{2.72}$$

Then the final estimator of the *Thresholded Lasso* is given by the ordinary maximum likelihood estimator when restricting to the model induced by $\hat{S}_{\text{Thres}} \subseteq \mathcal{P}$ with $q := |\hat{S}_{\text{Thres}}|$, i.e.

$$\hat{\mu}_{\text{Thres}}(\lambda_{\text{Init}}, \lambda_{\text{Thres}}),\, \hat{\boldsymbol{\beta}}_{\text{Thres}}(\lambda_{\text{Init}}, \lambda_{\text{Thres}}) = \underset{\mu \in \mathbb{R},\, \boldsymbol{\beta}_{\hat{S}_{\text{Thres}}} \in \mathbb{R}^q}{\arg\max} f(\boldsymbol{y} \,|\, \boldsymbol{X}_{\hat{S}_{\text{Thres}}}, \boldsymbol{\beta}_{\hat{S}_{\text{Thres}}}, \mu) . \tag{2.73}$$

Even though this procedure seems relatively simple, Bühlmann and van de Geer (2011) show that the Thresholded Lasso enjoys, on the rough level, very similar theoretical properties as the Adaptive Lasso. In particular, theoretical results for the linear model indicate that the Thresholded Lasso and the Adaptive Lasso may be preferred over the *Lasso-OLS hybrid* (compare Bühlmann and van de Geer, 2011, p. 33f) which is defined as the Thresholded Lasso in equation (2.73) but with the set $\hat{S}_{\text{Init}} = \{j \in \mathcal{P};\ \hat{\boldsymbol{\beta}}_{\text{Init}}(\lambda_{\text{Init}}) \neq 0\}$ in place of the set $\hat{S}_{\text{Thres}}$; i.e. the Lasso-OLS hybrid does not include an additional "thresholding step".

Many other variants of the Lasso have been proposed, such as the Fused Lasso (Tibshirani et al., 2005), the Dantzig Selector (Candes and Tao, 2007), the Relaxed Lasso (Meinshausen, 2007), the Scout Method (Witten and Tibshirani, 2009) or the SLOPE (Bogdan et al., 2015). We refer to Tibshirani (2011) for a compact summary of further developments and extensions of the Lasso methodology between the years 1996 and 2011.

2.5. Nonconvex methods

In the preceding sections we have seen that convex ℓ_1-type regularization methods like the Lasso typically require more restrictive assumptions than ℓ_0-type selection criteria in order to obtain variable selection consistency and we have argued that the bias induced by ℓ_1-type methods might be a possible reason for this undesirable behaviour. This has been one of the main motivations for developing alternative nonconvex relaxations that try to reduce the bias

inherent in ℓ_1-type methods and try to mimic the original ℓ_0-type penalization in a "closer" way. A natural candidate for this would be Bridge Regression (see Definition 2.18) with a choice of $q \in (0,1)$, but it turns out that certain alternative nonconvex penalties have better computational and statistical properties since they lead to continuous estimators reducing the instability of the corresponding procedures (compare the discussion in Fan and Li, 2001, Section 2).

The most prominent examples for nonconvex methods are the *SCAD* (smoothly clipped absolute deviation penalty, Fan and Li, 2001) and the *MC+* (*MC*: minimax concave penalty, *+*(*PLUS*): penalized linear unbiased selection algorithm, Zhang, 2010). Recently, Song and Liang (2015b) have proposed a quite unusual reciprocal ℓ_1-type regularization which employs a penalty function that is not continuous at zero. We do not want to discuss these methods in detail here, but provide at least the definition of the SCAD as an illustrative example for such a nonconvex method.

Definition 2.26 (Fan and Li, 2001)**.** The *SCAD* with penalty parameters $\lambda > 0$ and $a > 2$ is given by the optimization problem

$$\hat{\mu}(\lambda, a),\, \hat{\boldsymbol{\beta}}(\lambda, a) = \underset{\mu \in \mathbb{R},\, \boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \left(-\frac{2}{n} \log\left(f(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\beta}, \mu) \right) + \sum_{j \in \mathcal{P}} h_{\lambda,a}(\beta_j) \right), \tag{2.74}$$

where the penalty function $h_{\lambda,a} : \mathbb{R} \to (0, \infty)$ is, for $x \in \mathbb{R}$, defined by

$$h_{\lambda,a}(x) = \begin{cases} \lambda |x| & , \text{if } |x| \leq \lambda\,, \\ -\frac{x^2 - 2a\lambda|x| + \lambda^2}{2a-2} & , \text{if } \lambda < |x| \leq a\lambda\,, \\ \frac{(a+1)\lambda^2}{2} & , \text{if } |x| > a\lambda\,. \end{cases} \tag{2.75}$$

The SCAD penalty function $h_{\lambda,a}$ is a quadratic spline function with knots at λ and $a\lambda$. Note that for small regression coefficients (i.e. for $|x| \leq \lambda$), the SCAD penalty coincides with the ℓ_1-penalty of the Lasso. For "medium sized" coefficients (i.e. for $\lambda < |x| < a\lambda$), the imposed penalty still increases in $|x|$ but at a slower and slower pace until, for $|x| \geq a\lambda$, we have $h'_{\lambda,a}(x) = 0$, implying that the SCAD imposes a constant penalty for large regression coefficients which resembles the behaviour of the ℓ_0-penalty. The additional tuning parameter $a > 2$ is typically chosen to the fixed value of $a = 3.7$ (compare the discussion in Fan and Li, 2001, Section 2.1).

It has been shown that, in contrast to the Lasso, the SCAD enjoys the oracle property (see Definition 2.24) under suitable conditions for GLMs (Fan and Li, 2001; Fan and Peng, 2004; Fan and Lv, 2011). However, since the corresponding optimization problems are nonconvex, one cannot generally guarantee that one identifies the global optimal solution of the SCAD.

2.6. Screening methods

Nowadays one often faces very high-dimensional data where the number of explanatory variables p largely exceeds the sample size n. In these (ultra-)high-dimensional scenarios one might be first interested in reducing the number of variables with a computationally efficient screening method and then afterwards proceed with the analysis using only a significantly smaller number of variables. It is desirable that in such a screening step we do not "loose" true important variables, i.e. the so-called *sure screening property* (Fan and Lv, 2008) should be satisfied.

Definition 2.27 (Fan and Lv, 2008)**.** Consider a particular asymptotic setting as in Notation 2.7. Let $\{E_{n,p}\}$ be a variable selection procedure in the sense of Definition 2.6. Then the procedure $\{E_{n,p}\}$ satisfies the *sure screening property* for the given asymptotic setting if

$$P\left(E_{n,p}(\mathcal{D}) \supseteq S_0^{(n)}\right) = P\left(\hat{S}^{(n)} \supseteq S_0^{(n)}\right) \to 1\,, \quad n \to \infty\,. \tag{2.76}$$

In the setting of normal linear models, Fan and Lv (2008) propose a very simple, yet efficient method called *Sure Independence Screening* which is based on ranking the univariate marginal correlations between the response Y and the individual explanatory variables X_j, $j \in \mathcal{P}$. The approach has been extended by Fan and Song (2010) to the general setting of GLMs by considering the ranking of maximum marginal likelihood estimators.

Definition 2.28 (Fan and Song, 2010)**.** Suppose that the observed values of each explanatory variable X_j have been standardized such that they have mean zero and standard deviation one. For each $j \in \mathcal{P}$, consider maximum marginal likelihood estimators given by

$$\hat{\mu}_j, \hat{\beta}_j = \underset{\mu \in \mathbb{R},\, \beta_j \in \mathbb{R}}{\arg\max}\, f(\boldsymbol{y} \,|\, X_j, \beta_j, \mu)\,, \tag{2.77}$$

where $f(\boldsymbol{y} \,|\, X_j, \beta_j, \mu)$ denotes the likelihood under the GLM including only an intercept μ and variable X_j with corresponding coefficient β_j. Then the model selected by *Sure Independence*

Screening (SIS) with threshold $\gamma \geq 0$ is defined by

$$\hat{S}_{\text{SIS}}(\gamma) = \{j \in \mathcal{P};\, |\hat{\beta}_j| \geq \gamma\}\,. \tag{2.78}$$

Note that for normal linear models, ranking the maximum marginal likelihood estimators of the coefficients is equivalent to ranking the marginal correlations between Y and X_j for $j \in \mathcal{P}$, so that Definition 2.28 coincides with the original SIS proposed by Fan and Lv (2008) for normal linear models. Under certain conditions which particularly rule out marginal independence between individual "signal" variables and the response, it has been shown that for appropriate choices of the threshold $\gamma_n \geq 0$, SIS satisfies the sure screening property even in ultra-high-dimensional asymptotic settings where the number of variables p_n might grow exponentially with the sample size n, i.e. $\log(p_n) = \mathcal{O}(n^c)$ for some $c > 0$ (Fan and Lv, 2008; Fan and Song, 2010). Note that if there are many "noise" variables which are strongly correlated with "signal" variables, then many of these "noise" variables will also be marginally associated with the response, so that the size of the set $\hat{S}_{\text{SIS}}(\gamma_n)$ selected by SIS will typically be quite large in order to ensure the sure screening property in such multicollinear situations.

SIS can naturally be used in certain two-step procedures, where in the first step the number of possible explanatory variables is (maybe drastically) reduced by SIS and then in the second step the final model is selected by applying another more refined regularization method like the Lasso, the Adaptive Lasso or the SCAD based on the already reduced model obtained by SIS (compare Fan and Lv, 2008, Section 3). These two-stage procedures are referred to as *SIS-Lasso*, *SIS-AdaLasso* and *SIS-SCAD*, respectively.

The main feature of SIS is that the screening is based only on the univariate contributions of single explanatory variables to the response and that interdependencies between the variables are not taken into account, which might not be optimal in a situation with high underlying correlation between the variables. Therefore, Fan and Lv (2008) propose an extension of SIS, called *Iterative Sure Independence Screening (ISIS)*, in which a two-step procedure with SIS (like SIS-SCAD) is applied multiple times in some iterative procedure where the current residuals are used as new responses for the next iteration. For details we refer to Fan and Lv (2008, Section 4).

Finally, we want to note that even though the Lasso requires very restrictive assumptions

for variable selection consistency, Meinshausen and Yu (2009) show that one can obtain the sure screening property of the Lasso under the significantly weaker restricted eigenvalue condition on the fixed design matrix $\boldsymbol{X}$ and a "beta-min condition" similar to equation (2.71). Therefore, Bühlmann suggests that "the original translation of Lasso (Least Absolute Shrinkage and Selection Operator) may be better re-translated as Least Absolute Shrinkage and *Screening* Operator" (Bühlmann, 2013b, p. 4). Similarly, Wang (2009) shows that a version of Forward Stepwise Selection in combination with a stopping criterion like the EBIC enjoys the sure screening property under similar conditions as the Lasso.

2.7. Resampling methods

A general problem of both ℓ_0- and ℓ_1-type regularization methods is that their optimal solution is not very "stable" with respect to small changes in the data. Here, the concept of "stability" refers to the desirable property that the subsets of variables selected by a particular method do not largely vary if different samples from the same population are considered (compare Meinshausen and Bühlmann, 2010), meaning that the selected models do not include many false positive selections. In particular, it has been observed that the discrete nature of the ℓ_0-penalty can lead to "overfitting" of the criterion, if the optimization is carried out among all possible 2^p models (see e.g. Breiman, 1996; Loughrey and Cunningham, 2005). Another problem of ℓ_1-type criteria is that they do not provide any information about the uncertainty concerning the best model, per se.

Different resampling methods have been proposed in order to address these issues and in order to increase the stability of the selected models. While the aim of screening methods, discussed in the previous section, is to avoid false negative selections, the aim of such resampling methods is basically to guard against a large number of false positives. Meinshausen and Bühlmann (2010) propose a procedure called *Stability Selection* which is based on the idea of applying a given variable selection procedure $\hat{S}^{(\lambda)}$ with additional tuning parameter $\lambda > 0$ (e.g. the Lasso with penalty parameter λ) multiple times on subsamples of the data. The subsampling scheme is to draw subsets I of a given size (for example of size $\lfloor \frac{n}{2} \rfloor$) without replacement from $\{1, \ldots, n\}$ and then repeatedly apply the given variable selection procedure on data with observations corresponding to indices $i \in I$ only. At the end, one selects those explanatory variables whose relative selection frequencies exceed a certain threshold

$\rho \in (0, 1)$. The detailed Stability Selection method is given as Algorithm 2.3.

Algorithm 2.3 Stability Selection (Meinshausen and Bühlmann, 2010)

Input:

- Data $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$
- Variable selection procedure $\hat{S}^{(\lambda)}$ with tuning parameter $\lambda > 0$
- Grid $\Lambda \subseteq (0, \infty)$ of tuning parameter values $\lambda \in \Lambda$
- Number of iterations $T \in \mathbb{N}$

Algorithm:

(1) Initialize absolute selection frequencies $f_j^{(\lambda)} = 0$ for $j \in \mathcal{P}$ and $\lambda \in \Lambda$.

(2) For $\lambda \in \Lambda$ and $t = 1, \ldots, T$:

  (a) Draw subsample I_t of size $\lfloor \frac{n}{2} \rfloor$ without replacement from $\{1, \ldots, n\}$.

  (b) Apply the selection procedure $\hat{S}^{(\lambda)}$ on data with observations corresponding to indices in I_t only, i.e. compute $\hat{S}^{(\lambda)}(\mathcal{D}_{I_t})$ where $\mathcal{D}_{I_t} = (\boldsymbol{X}_{I_t,*}, \boldsymbol{y}_{I_t})$.

  (c) For $j \in \hat{S}^{(\lambda)}(\mathcal{D}_{I_t})$ update $f_j^{(\lambda)} \leftarrow f_j^{(\lambda)} + 1$.

(3) Compute relative selection frequencies $r_j^{(\lambda)} = \frac{f_j^{(\lambda)}}{T}$ for $j \in \mathcal{P}$ and $\lambda \in \Lambda$.

Output (Final subset selected by Stability Selection):

- $\hat{S}_{\text{Stab},\rho} = \{j \in \mathcal{P};\ \max_{\lambda \in \Lambda} r_j^{(\lambda)} \geq \rho\}$ for some threshold $\rho \in (0, 1)$

The main feature of Stability Selection is that the threshold ρ can be selected in such a way that, under some theoretical conditions, we can control the expected number of false positive selections.

Theorem 2.6 (Meinshausen and Bühlmann, 2010)**.** *Consider Stability Selection (Algorithm 2.3) with a given variable selection procedure $\hat{S}^{(\lambda)}$ and tuning parameter $\lambda \in \Lambda \subseteq (0, \infty)$. Let $\hat{S}^{\Lambda} = \cup_{\lambda \in \Lambda} \hat{S}^{(\lambda)}$ be the union of all possible selected subsets and let $q_\Lambda = E\left[|\hat{S}^{\Lambda}(\mathcal{D}_I)|\right]$ denote the corresponding expected number of selected variables, where the expectation is taken with respect to the random subsampling scheme ($I \subseteq \{1, \ldots, n\}$).*

Assume that the distribution of $\{\mathbb{1}_{\{k \in \hat{S}^{(\lambda)}\}};\ k \in \mathcal{P} \setminus S_0\}$ is exchangeable for all $\lambda \in \Lambda$ and that the variable selection procedure $\hat{S}^{(\lambda)}$ is not worse than random guessing in the sense that

$$\frac{E\left[\left|S_0 \cap \hat{S}^{\Lambda}\right|\right]}{E\left[\left|(\mathcal{P} \setminus S_0) \cap \hat{S}^{\Lambda}\right|\right]} \geq \frac{|S_0|}{|\mathcal{P} \setminus S_0|}\,. \tag{2.79}$$

Then, for $\rho \in (0.5, 1)$, the expected number of false positives in $\hat{S}_{Stab,\rho}$ is bounded by

$$E\left[\left|\hat{S}_{Stab,\rho} \cap (\mathcal{P} \setminus S_0)\right|\right] \leq \frac{q_\Lambda^2}{(2\rho - 1)p}\,. \tag{2.80}$$

The theoretical bound (2.80) on the expected number of false positives can be used for the choice of the threshold ρ given a sensible choice of the grid Λ, or vice versa. For details we refer to Meinshausen and Bühlmann (2010). The exchangeability assumption in Theorem 2.6, needed for the error control, is arguably quite strong and can be dropped if one uses a simple extension of Stability Selection, called Complementary Pairs Stability Selection (Shah and Samworth, 2013). Instead of considering only the selected subset $\hat{S}^{(\lambda)}(\mathcal{D}_I)$ for some subsample $I \subseteq \{1, \ldots, n\}$ of size $|I| = \lfloor n/2 \rfloor$ at each iteration of the algorithm, Shah and Samworth (2013) also consider the selected subset $\hat{S}^{(\lambda)}(\mathcal{D}_{I'})$ with complementary indices $I' = \{1, \ldots, n\} \setminus I$.

Another approach closely related to Stability Selection is the so-called Bolasso (Bach, 2008), in which the Lasso is applied multiple times on bootstrapped data of sample size n (by sampling **with** replacement from $\{1, \ldots, n\}$). Further extensions of Stability Selection have been proposed in Beinrucker et al. (2016), where a baseline method like the Lasso is iteratively applied to random submatrices of the original design matrix $\boldsymbol{X}$ (i.e. sampling without replacement from $\{1, \ldots, n\}$ **and** $\{1, \ldots, p\}$).

Even though these different variants and extensions of Stability Section have been developed in its own right, one might view them as attempts to correct for one particular deficiency of Stability Selection: In a high-dimensional situation with $p \gg n$, Stability Selection successively applies a possibly inconsistent selection procedure like the Lasso (compare Thereom 2.4) on even more severe high-dimensional problems with $p \ggg \lfloor \frac{n}{2} \rfloor$. This observation serves as one important motivation for the development of the so-called Adaptive Subspace (AdaSub) method, which we introduce in the next chapter of this thesis.

3. Adaptive Subspace (AdaSub) method

In the previous chapter we have contrasted ℓ_0- and ℓ_1-type methods for high-dimensional variable selection and discussed their advantages and disadvantages, respectively: While ℓ_0-type methods like the EBIC are variable selection consistent under weak conditions (compare Theorem 2.2), the resulting combinatorial optimization problems are generally NP-hard. On the other hand, convex ℓ_1-type methods like the Lasso are computationally efficient, but they require much stronger conditions for variable selection consistency (compare Theorem 2.4). We have also briefly discussed different resampling methods (see Section 2.7) which aim at enhancing the stability of the selected models by different variable selection methods. Even though the Stability Selection procedure (see Algorithm 2.3) has nice theoretical properties (compare Theorem 2.6) and also seems to be used more and more in practice, in Section 2.7 we have noted that in a high-dimensional situation with $p \gg n$, Stability Selection (in combination with the Lasso) successively applies a possibly inconsistent selection procedure on even more severe high-dimensional problems with $p \ggg \lfloor \frac{n}{2} \rfloor$.

Motivated from these theoretical insights, in this chapter we introduce the Adaptive Subspace (AdaSub) method which is fundamentally based on the idea of successively applying a consistent variable selection procedure (ℓ_0-type criteria like EBIC) on data with the original sample size n and only a few $q < n$ out of the p covariates. So the ideology behind AdaSub can be summarized as: "Solve a high-dimensional problem by solving several low-dimensional sub-problems."

Two issues naturally arise in this regime: Which low-dimensional problems should be solved? And how can the information from the solved low-dimensional problems be combined in order to solve the original problem? AdaSub links the answers to those questions using a certain form of adaptive learning: In each iteration of the algorithm, the solutions from the already solved low-dimensional problems are used to construct (or more precisely "sample" in a stochastic way) a new low-dimensional problem of potentially higher interest, where the

construction is based on the principle that, when repeatedly restricting to lower-dimensional subspaces of the full model space, a relevant explanatory variable should also be important in many cases.

This chapter is structured as follows: We start with a short motivating analogy in Section 3.1, which aims at giving some intuitive understanding of the main methodological differences of AdaSub in comparison to other variable selection procedures that have been discussed in the previous chapter. In Section 3.2 we present the generic Adaptive Subspace (AdaSub) method. In Section 3.3 we show that AdaSub can be viewed as a Markov chain and thus has similarities to Markov Chain Monte Carlo (MCMC) algorithms. Furthermore, in Section 3.4 we interpret the updating scheme of AdaSub as a basic form of Bayesian learning. In Section 3.5 we discuss the choice of the tuning parameters of AdaSub and introduce important "diagnostic" plots for assessing the convergence of the algorithm. In Section 3.6 the computational cost of AdaSub is examined. Finally, in Section 3.7 we compare AdaSub with different other related methods that have recently been proposed in the literature.

Note that an early version of the material of Sections 3.2, 3.3 and 3.4 has been published in the conference proceedings paper Staerk et al. (2016). The material presented in Section 3.5 is partially included in the conference proceedings paper Staerk and Kateri (2017). Further parts of this chapter (Sections 3.2, 3.3, 3.4 and 3.7) are contained in Staerk et al. (2018), which has been submitted for publication in a journal.

3.1. A motivating analogy

The following analogy aims at providing an intuitive motivation for AdaSub and should not be understood as a one-to-one correspondence. Suppose that you are the new manager of FC Liverpool and you want to find the best football team in the world. This task might be thought of as a variable selection problem where each variable represents a player (say there is a total number of $p = 10^6$ players). You might apply one of the following "scouting strategies":

1. Visit each player at home and assess their individual ball skills. Screen out those with the worst skills (corresponds to Sure Independence Screening, SIS).

2. Invite all players to Liverpool, put them on one football field and watch what hap-

pens. Select those which seem to have performed best (corresponds to any stand-alone method like the Lasso or EBIC).

3. Invite all players to Liverpool, put them on one football field and let them successively play with different obstacles (e.g. first blindfolded, then on only one leg,...). Select those which seem to have performed best on average (corresponds to Stability Selection).

4. Over a (long) time period observe the players successively in different teams of only a few players. Your effort on observing a certain player should be proportional to his current estimated skills (corresponds to the proposed AdaSub method).

While this is only a loose pedagogical analogy, it may provide some basic intuition that attacking the high-dimensional variable selection problem "directly" by simply solving an ℓ_0- or ℓ_1-type optimization problem (strategy 2) might not be the best strategy since the dependencies between the numerous variables (i.e. the interactions between the different players) can be very difficult to control. Solving many of such high-dimensional optimization problems (strategy 3) can improve the stability of the results, but does not reduce the problematic high-dimensionality of the data. However, by solving several low-dimensional sub-problems of the original high-dimensional problem (strategy 4) one might gain a better understanding of the contributions of (generally small) groups of variables to the response.

3.2. The AdaSub algorithm

In order to describe the AdaSub algorithm, we first introduce some notation in a setting with a criterion-based variable selection procedure. Here, we present the AdaSub algorithm in the general setting of variable selection in a (fixed) GLM framework. An important special case is the search for the "best" subset among a possibly large set of explanatory variables in normal linear models.

Notation 3.1. Let $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$ denote the observed data, as defined in Notation 2.1, which we want model with a particular GLM from $\mathcal{M}$, where $\mathcal{M}$ denotes the restricted space of models (due to identifiability and overfitting reasons, compare Remark 2.4), as defined in equation (2.20). Let $C_{\mathcal{D}} : \mathcal{M} \to \mathbb{R}$ be a certain model selection criterion. In the following we will write $C \equiv C_{\mathcal{D}}$ for brevity, but one should always recall that the function C depends on

the observed data $\mathcal{D}$. We assume w.l.o.g. that we want to find the model that maximizes the given criterion C. Examples for C include posterior model probabilities (within the Bayesian setup) or the negative EBIC (within the ℓ_0-penalized criteria framework). We define

$$f_C : \mathfrak{P}(\{1,\ldots,p\}) \to \mathcal{M},\ f_C(V) := \underset{S\subseteq V,\, S\in\mathcal{M}}{\arg\max}\, C(S)\,, \tag{3.1}$$

where $\mathfrak{P}(\{1,\ldots,p\}) = \{V \subseteq \{1,\ldots,p\}\}$ denotes the power set of $\mathcal{P} = \{1,\ldots,p\}$. So for a given $V \subseteq \mathcal{P}$, $f_C(V)$ is the best model according to criterion C among all models included in V. For simplicity, in this chapter we will make the assumption that

$$C(S) \neq C(S') \text{ for all } S, S' \in \mathcal{M} \text{ with } S \neq S'\,, \tag{3.2}$$

so that f_C is a well-defined function which maps a subset $V \subseteq \mathcal{P}$ to a single model $f_C(V) \in \mathcal{M}$. In the ℓ_0-penalized likelihood framework this assumption is almost surely satisfied for normal linear models if the values of the explanatory variables are generated from an absolutely continuous distribution with respect to the Lebesgue measure (see e.g. Nikolova, 2013). The AdaSub method can easily be generalized to the situation of competing models with the same criterion value and the convergence properties of AdaSub still hold with slight modifications (see Remark 4.4 in Section 4.1). Let $S^* := \arg\max_{S\in\mathcal{M}} C(S)$ with $s^* = |S^*|$ denote the best model according to criterion C which is unique under the adopted assumptions. We also say that S^* is a C-optimal set or a C-optimal model.

We will now describe the generic Adaptive Subspace (AdaSub) method, which is given as Algorithm 3.1. If we have observed some data $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$, the aim is to identify the C-optimal model $S^* = \arg\max_{S\in\mathcal{M}} C(S)$ according to criterion C. As explained above, the basic idea of AdaSub is to solve many low-dimensional sub-problems (i.e. compute $f_C(V)$ for many $V \subseteq \mathcal{P}$ with $|V|$ relatively small) in order to solve the given high-dimensional problem (i.e. compute $S^* = f_C(\mathcal{P})$).

AdaSub is a stochastic algorithm which in each iteration t, for $t = 1,\ldots,T$, samples a subset $V^{(t)} \subseteq \mathcal{P}$ of the set of all possible explanatory variables $\mathcal{P} = \{1,\ldots,p\}$ and then computes $S^{(t)} = f_C(V^{(t)})$. The probability that $j \in \mathcal{P}$ is included in $V^{(t)}$ at iteration t is given by $r_j^{(t-1)}$. The selection probabilities $r_j^{(t)}$ are automatically adjusted after each iteration t in the following way:

$$r_j^{(t)} = \frac{q + K\sum_{i=1}^{t} \mathbb{1}_{S^{(i)}}(j)}{p + K\sum_{i=1}^{t} \mathbb{1}_{V^{(i)}}(j)}\,, \tag{3.3}$$

where $q \in (0, p)$ and $K > 0$ are tuning parameters of the algorithm and $\mathbb{1}_A$ denotes the indicator function of a set A. Note that $r_j^{(t)}$ depends on the whole history (from iteration 1 up to iteration t) of the number of times variable X_j has been considered in the search ($j \in V^{(i)}$) and the number of times it has been included in the best subset ($j \in S^{(i)}$).

If $j \in V^{(t)}$ but $j \notin S^{(t)} = f_C(V^{(t)})$, then $r_j^{(t)} < r_j^{(t-1)}$, so the selection probability of variable X_j decreases in the next iteration. If $j \in V^{(t)}$ and also $j \in S^{(t)}$, then $r_j^{(t)} > r_j^{(t-1)}$, so the selection probability increases. If $j \notin V^{(t)}$, then obviously $j \notin S^{(t)}$, so the selection probability does not change in the next iteration. Clearly we have $0 < r_j^{(t)} < 1$ for all $t = 1, \ldots, T$ and $j \in \mathcal{P}$. So at each iteration t each variable X_j has positive probability $r_j^{(t)}$ of being considered in the model search ($j \in V^{(t)}$) and also has positive probability $1 - r_j^{(t)}$ of being not considered ($j \notin V^{(t)}$).

Algorithm 3.1 Adaptive Subspace (AdaSub) method

Input:

- Data $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$
- $C : \mathcal{M} \to \mathbb{R}$ model selection criterion ($C \equiv C_{\mathcal{D}}$)
- Initial expected search size $q \in (0, p)$
- Learning rate $K > 0$
- Number of iterations $T \in \mathbb{N}$

Algorithm:

(1) For $j \in \mathcal{P}$ initialize selection probability of variable X_j as $r_j^{(0)} := \frac{q}{p}$.

(2) For $t = 1, \ldots, T$:

  (a) Draw $b_j^{(t)} \sim \text{Bernoulli}(r_j^{(t-1)})$ independently for $j \in \mathcal{P}$.

  (b) Set $V^{(t)} = \{j \in \mathcal{P};\ b_j^{(t)} = 1\}$.

  (c) Compute $S^{(t)} = f_C(V^{(t)})$.

  (d) For $j \in \mathcal{P}$ update $r_j^{(t)} = \frac{q + K \sum_{i=1}^{t} \mathbb{1}_{S^{(i)}}(j)}{p + K \sum_{i=1}^{t} \mathbb{1}_{V^{(i)}}(j)}$.

Output (Final model selected by AdaSub):

(i) "Best" sampled model: $\hat{S}_{\text{b}} = \arg\max\{C(S^{(1)}), \ldots, C(S^{(T)})\}$

(ii) Thresholded model: $\hat{S}_\rho = \{j \in \mathcal{P};\ r_j^{(T)} > \rho\}$ for some threshold $\rho \in (0, 1)$

As the final subset selected by AdaSub one can either (i) choose the "best" sampled model $\hat{S}_\mathrm{b}$ for which $C(\hat{S}_\mathrm{b}) = \max\{C(S^{(1)}), \ldots, C(S^{(T)})\}$, or (ii) consider the thresholded model $\hat{S}_\rho = \{j \in \mathcal{P};\ r_j^{(T)} > \rho\}$ with some threshold $\rho \in (0, 1)$. While $\hat{S}_\mathrm{b}$ is obviously more likely to coincide with the C-optimal model S^*, it can be beneficial in terms of variable selection "stability" to consider the thresholded model $\hat{S}_\rho$ instead (with ρ relatively large). A detailed discussion is given in Chapter 4.

The Adaptive Subspace method requires that we initialize three parameters q, K and T. Here $q \in (0, p)$ is the initial expected search size, which should be relatively small (e.g. $q = 10$). The initial expected search size q reflects our prior belief about the sparsity of the problem, i.e. q should be a first rough "estimate" of the size of the C-optimal model. We have

$$E\left(|V^{(1)}|\right) = E\left(\sum_{j=1}^{p} b_j^{(1)}\right) = \sum_{j=1}^{p} E b_j^{(1)} = \sum_{j=1}^{p} r_j^{(0)} = \sum_{j=1}^{p} \frac{q}{p} = q, \tag{3.4}$$

so the expected search size in the first iteration is indeed q. In the following iterations the expected search size is automatically adapted depending on the sizes of the previously selected models $S^{(i)}$. The parameter $K > 0$ controls the learning rate (or adaptation rate) of the algorithm. The larger K is chosen, the faster the selection probabilities $r_j^{(t)}$ of the variables X_j are adapted. A more precise interpretation and discussion of the choice of the parameters q and K is given in Sections 3.4 and 3.5, where we interpret the proposed algorithm as a form of Bayesian learning and illustrate the performance of the algorithm for different choices of the tuning parameters, respectively. The number of iterations $T \in \mathbb{N}$ can be specified in advance. Alternatively one might impose an automatic stopping criterion for the algorithm, but we strongly advise to inspect the output of AdaSub by appropriate plots and assess the convergence of the algorithm manually (see e.g. Section 3.5 for examples of diagnostic plots).

Note that we implicitly assume that it is computationally feasible to compute $S^{(t)} = f_C(V^{(t)})$ in each iteration t. In fact, if the underlying "truth" is sparse and the criterion used enforces sparsity, $|V^{(t)}|$ is expected to be relatively small. Otherwise one might use heuristic algorithms in place of a full enumeration (see Chapter 6). Alternatively, if $|V^{(t)}|$ is bigger than some computational bound $U_C \in \mathbb{N}$, one might replace $V^{(t)}$ by a subsample of $V^{(t)}$ of size U_C. In the case of variable selection in linear regression with $C(S) = -\mathrm{EBIC}(S)$ using the fast branch-and-bound algorithm (Lumley and Miller, 2009) one might set $U_C \leq 40$.

Furthermore, note that for GLMs different than the normal linear model the MLEs of the regression coefficients are generally not available in closed form (compare Remark 2.3), so that the solution of the relatively low-dimensional sub-problems can be computationally expensive. We refer to Section 3.6 and Chapter 6 for a more detailed discussion and computationally efficient modifications of AdaSub, respectively. For the remainder of this chapter, we will assume that the original AdaSub method (Algorithm 3.1) is used.

Finally, we want to note that in the case of categorical explanatory variables, the AdaSub algorithm can be used without any fundamental methodological changes: If a qualitative variable X_j, for some $j \in \mathcal{P} = \{1, \dots, p\}$, has potentially $c > 2$ different categories, one can replace X_j with $c-1$ binary "dummy" variables $X_{j,1}, \dots, X_{j,c-1}$ in each fit of a GLM induced by a subset $S \in \mathcal{M}$ with $j \in S$, i.e. the only change in AdaSub concerns the evaluations of the criterion C in step 2 (c) of Algorithm 3.1. By this way, AdaSub can lead to a natural way of grouping, in the sense that either all "dummy" variables $X_{j,1}, \dots, X_{j,c-1}$ associated with the categorical variable X_j are included in the model (i.e. variable X_j is "important") or that all of them are not included in the model (i.e. variable X_j is "unimportant"). Recall that ℓ_1-type methods like the Lasso do not enforce such grouping effects, per se, unless the penalty function is modified as for example in the Group Lasso (see Definition 2.22 of Section 2.4.3)

3.3. AdaSub as a Markov chain

The evolution of the AdaSub algorithm can be described by a Markov chain $(\boldsymbol{Q}^{(t)})_{t \in \mathbb{N}_0}$ with state space $S = (\mathbb{N}_0^2)^p$, where

$$\boldsymbol{Q}^{(t)} = \left(\left(A_1^{(t)}, B_1^{(t)} \right), \left(A_2^{(t)}, B_2^{(t)} \right), \dots, \left(A_p^{(t)}, B_p^{(t)} \right) \right) \in \left(\mathbb{N}_0^2 \right)^p \tag{3.5}$$

with

$$A_j^{(t)} = \sum_{i=1}^{t} \mathbb{1}_{S^{(i)}}(j) \quad \text{and} \quad B_j^{(t)} = \sum_{i=1}^{t} \mathbb{1}_{V^{(i)}}(j) \tag{3.6}$$

for $j \in \mathcal{P}$, $t \in \mathbb{N}$. Additionally, we set $\boldsymbol{Q}^{(0)} = ((0,0), \dots, (0,0))$ as the initial state of the Markov chain. So the tuple $\left(A_j^{(t)}, B_j^{(t)} \right) \in \mathbb{N}_0^2$ describes the relevant information available about variable X_j after iteration t of the algorithm, i.e. $B_j^{(t)}$ is the number of times that variable X_j has been considered in a subspace and $A_j^{(t)}$ is the number of times X_j has been

selected to be in the "best" submodel. Obviously we have $A_j^{(t)} \leq B_j^{(t)}$ for all $j \in \mathcal{P}$ and $t \in \mathbb{N}$. In order to obtain the transition kernel of the Markov chain we follow the steps of the AdaSub method (Algorithm 3.1):
For $\boldsymbol{Q}^{(t-1)} \in (\mathbb{N}_0^2)^p$, $t \in \mathbb{N}$, we generate $\boldsymbol{b}^{(t)} = \left(b_1^{(t)}, \ldots, b_p^{(t)}\right) \in \{0,1\}^p$ according to an independent Bernoulli distribution with probability

$$P^{\boldsymbol{b}^{(t)}}(\boldsymbol{b}^{(t)}) = \prod_{j=1}^{p} \left(r_j^{(t-1)}\right)^{b_j^{(t)}} \left(1 - r_j^{(t-1)}\right)^{1-b_j^{(t)}} , \tag{3.7}$$

where

$$r_j^{(t-1)} = \frac{q + KA_j^{(t-1)}}{p + KB_j^{(t-1)}} \quad , j \in \mathcal{P}\,.$$

We make use of the bijective map

$$g : \{0,1\}^p \to \mathfrak{P}(\{1, \ldots, p\}), \quad g(\boldsymbol{b}^{(t)}) := \{j \in \mathcal{P};\ b_j^{(t)} = 1\} \tag{3.8}$$

in order to obtain $V^{(t)} = g(\boldsymbol{b}^{(t)}) \subseteq \mathcal{P}$. Finally, we apply the map f_C and obtain $S^{(t)} = f_C(V^{(t)}) \in \mathcal{M}$. Now, for the given state $\boldsymbol{Q}^{(t-1)} \in (\mathbb{N}_0^2)^p$ and $\boldsymbol{b}^{(t)}, V^{(t)}, S^{(t)}$ as above, we are able to define the next state

$$\boldsymbol{Q}_{\boldsymbol{b}^{(t)}}^{(t)} = \left(\left(A_1^{(t-1)} + \mathbb{1}_{S^{(t)}}(1), B_1^{(t-1)} + \mathbb{1}_{V^{(t)}}(1)\right), \ldots, \left(A_p^{(t-1)} + \mathbb{1}_{S^{(t)}}(p), B_p^{(t-1)} + \mathbb{1}_{V^{(t)}}(p)\right)\right) . \tag{3.9}$$

Note that $\boldsymbol{Q}_{\boldsymbol{b}^{(t)}}^{(t)}$ depends solely on $\boldsymbol{b}^{(t)}$, since $\boldsymbol{b}^{(t)}$ is the only random part of the algorithm and after $\boldsymbol{b}^{(t)}$ has been drawn, $V^{(t)}$ and $S^{(t)}$ are obtained deterministically. So the transition kernel T of the Markov chain $(\boldsymbol{Q}^{(t)})_{t \in \mathbb{N}_0}$ is given by

$$T\left(\boldsymbol{Q}^{(t)} \,|\, \boldsymbol{Q}^{(t-1)}\right) = \begin{cases} P^{\boldsymbol{b}^{(t)}}(\boldsymbol{b}^{(t)}) & \text{, if } \boldsymbol{Q}^{(t)} = \boldsymbol{Q}_{\boldsymbol{b}^{(t)}}^{(t)} \text{ for some } \boldsymbol{b}^{(t)} \in \{0,1\}^p\,, \\ 0 & \text{, otherwise}\,. \end{cases} \tag{3.10}$$

Thus AdaSub has analogies to Markov Chain Monte Carlo (MCMC) methods. However, instead of sampling from a certain target distribution as standard MCMC algorithms do, AdaSub constructs a Markov chain which does not have a limiting distribution in the classical sense, but which "converges" (in a sense and under conditions that need to be specified, see Chapter 4) to the (unknown) solution $S^* = \arg\max_{S \in \mathcal{M}} C(S)$ of the given optimization problem.

3.4. Bayesian motivation

The Adaptive Subspace method can also be interpreted as a basic form of Bayesian learning: Let π_j denote the subjective belief that variable X_j is in the best model S^* with respect to the given criterion C. Note that under knowledge of $S^* = \arg\max_{S \in \mathcal{M}} C(S)$, in fact π_j has a Dirac distribution concentrated at 1 if $j \in S^*$ or concentrated at 0 if $j \notin S^*$. But since we cannot solve the high-dimensional optimization problem exactly with full enumeration, we sequentially solve low-dimensional problems of the form $S^{(t)} = \arg\max_{S \in \mathfrak{S}^{(t)}} C(S)$, with $\mathfrak{S}^{(t)} = \{S \in \mathcal{M};\ S \subseteq V^{(t)}\}$, and sequentially update our belief π_j about variable X_j.

This process can be viewed as an adaptive experimental design where in each step we set up the design with explanatory variables given by $V^{(t)}$ (i.e. with associated design matrix $\boldsymbol{X}_{V^{(t)}}$, restricted to the columns with indices in $V^{(t)}$) and observe "new" data $\mathcal{D}^{(t)} = (\boldsymbol{X}_{V^{(t)}}, \boldsymbol{y})$. Clearly, $\mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(T)}$ are not independent, since we actually use the current information $\pi_1, \ldots, \pi_p \,|\, \mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(t-1)}$ to sample (or construct) $V^{(t)}$ and then "observe" the corresponding data $\mathcal{D}^{(t)} = (\boldsymbol{X}_{V^{(t)}}, \boldsymbol{y})$. The algorithm is constructed in such a way that we only make use of the marginal current information $\pi_j \,|\, \mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(t-1)}$ about each variable X_j.

At the beginning of the algorithm we assume that no specific information about the different variables is available, but we expect approximately q variables to be in the best model. So we put an independent beta prior on π_j for $j \in \mathcal{P}$:

$$\pi_j \sim \mathcal{B}e\left(\frac{q}{K}, \frac{p-q}{K}\right) \text{ with } E(\pi_j) = \frac{q}{p} = r_j^{(0)}\,. \tag{3.11}$$

Now in iteration t of AdaSub, $S^{(t)} = f_C(V^{(t)})$ is the "best" subset corresponding to the "new" observed data $\mathcal{D}^{(t)} = (\boldsymbol{X}_{V^{(t)}}, \boldsymbol{y})$. If $j \notin V^{(t)}$, then we are not learning anything about variable X_j in iteration t. But if $j \in V^{(t)}$, then the new information about variable X_j obtained in iteration t can be interpreted as the outcome of a Bernoulli trial where $j \in S^{(t)}$ corresponds to "success" and $j \notin S^{(t)}$ to "failure". Ignoring the dependence of $\pi_1, \ldots, \pi_p$ and $\mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(t)}$, we obtain the (pseudo) posterior distribution of π_j given $\mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(t)}$ as in the conjugate beta-binomial case:

$$\pi_j \,|\, \mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(t)} \sim \mathcal{B}e\left(\frac{q}{K} + \sum_{i=1}^{t} \mathbb{1}_{S^{(i)}}(j), \frac{p-q}{K} + \sum_{i=1}^{t} \mathbb{1}_{V^{(i)} \setminus S^{(i)}}(j)\right) \tag{3.12}$$

with posterior expectation

$$E\left[\pi_j \,|\, \mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(t)}\right] = \frac{q + K \sum_{i=1}^{t} \mathbb{1}_{S^{(i)}}(j)}{p + K \sum_{i=1}^{t} \mathbb{1}_{V^{(i)}}(j)} = r_j^{(t)} \tag{3.13}$$

and posterior variance

$$\mathrm{Var}\left[\pi_j \,|\, \mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(t)}\right] = C_j^{(t+1)} r_j^{(t)} \left(1 - r_j^{(t)}\right), \tag{3.14}$$

where

$$C_j^{(t+1)} = \frac{K}{p + K \sum_{i=1}^{t} \mathbb{1}_{V^{(i)}}(j) + K}. \tag{3.15}$$

The question remains as to how to choose the "prior variance parameter" $K > 0$. The prior variance of $\pi_j \sim \mathcal{B}e\left(\frac{q}{K}, \frac{p-q}{K}\right)$ is given by

$$\mathrm{Var}(\pi_j) = C_j^{(1)} r_j^{(0)} \left(1 - r_j^{(0)}\right) = \frac{K}{p+K} \times \frac{q}{p} \times \frac{p-q}{p} \tag{3.16}$$

and as limiting cases we obtain

$$\mathrm{Var}(\pi_j) \to \begin{cases} \frac{q}{p} \times \frac{p-q}{p} & , \text{if } K \to \infty, \\ 0 & , \text{if } K \to 0. \end{cases} \tag{3.17}$$

For $K \to \infty$ we are in the limiting case of a Bernoulli distribution with success probability $\frac{q}{p}$ and corresponding variance $\frac{q}{p} \times \frac{p-q}{p}$, which is the upper bound on the variance of a beta distribution with mean $\frac{q}{p}$. So choosing K very large corresponds to the most diffuse prior. If we choose K very small then we have low prior variance, meaning that we impose a large bias towards our prior belief that $\pi_j \approx \frac{q}{p}$.

In simulation studies the choice $K = n$ seems favourable (which is also supported by the sensitivity analysis presented in Section 3.5). Indeed, for $K = n$ we have

$$E\left[\pi_j \mid \mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(t)}\right] = \frac{q + \sum_{i=1}^{t} n \times \mathbb{1}_{S^{(i)}}(j)}{p + \sum_{i=1}^{t} n \times \mathbb{1}_{V^{(i)}}(j)}, \tag{3.18}$$

so that each single "success" ($j \in S^{(i)}$) and each single "trial" ($j \in V^{(i)}$) actually account for n "successes" and n "trials", respectively. This can be interpreted as a correction for the fact that each (pseudo) data $\mathcal{D}^{(i)} = (\boldsymbol{X}_{V^{(i)}}, \boldsymbol{y})$ considered at iteration i consists of sample size n.

3.5. Choice of tuning parameters in AdaSub

Extensive simulation studies for the investigation of the performance of AdaSub will be considered in Chapter 5 of this thesis. Here, in order to illustrate the performance of AdaSub in a high-dimensional setup and how it is effected by the choice of the tuning parameters K and q, we consider a normal linear model example with $p = 1000$ explanatory variables and sample size $n = 100$.

For this we generate data $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{Y})$ by simulating $\boldsymbol{X} = (X_{ij}) \in \mathbb{R}^{n \times p}$ with independent rows $\boldsymbol{X}_{i,*} \sim \mathcal{N}_p(\boldsymbol{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} \in \mathbb{R}^{p \times p}$ is a covariance matrix with entries $\Sigma_{k,l} = 0.3$ for $k \neq l$ and $\Sigma_{k,k} = 1$. Furthermore let $\boldsymbol{\beta}_0 = (1, -1, 1, 2, -2, 2, 0, \ldots, 0)^T \in \mathbb{R}^p$ be the true vector of coefficients with active set $S_0 = \{1, \ldots, 6\}$. The response $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^T$ is simulated via $Y_i \overset{\text{ind.}}{\sim} N(\boldsymbol{X}_{i,*}\boldsymbol{\beta}_0, 1)$, $i = 1, \ldots, n$.

The criterion C we adopt is the (negative) Extended BIC (EBIC_γ, as given in equation (2.28) of Section 2.2.2) with parameter $\gamma = 0.5$, which is especially suited for high-dimensional situations (Chen and Chen, 2008). The R-package `leaps` (Lumley and Miller, 2009), based on the efficient branch-and-bound algorithm, is used to compute the solutions of the low-dimensional sub-problems $f_C(V)$ for $V \subseteq \mathcal{P}$.

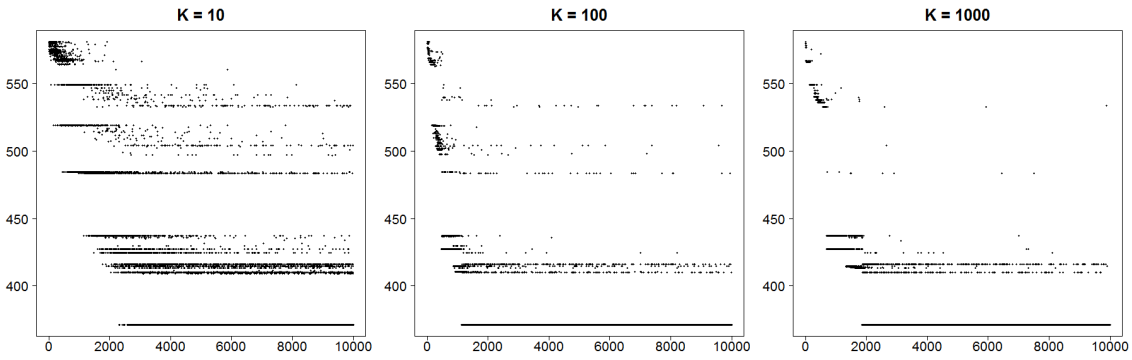


Figure 3.1.: AdaSub for high-dimensional example: Plot of the evolution of $\text{EBIC}_{0.5}(S^{(t)})$ along the iterations (t) for different values of K ($q = 5$ fixed).

We apply AdaSub with $T = 10,000$ iterations on a dataset simulated as above and fix $q = 5$ as the initial expected search size. We present some typical "diagnostic" plots for the particular simulated data example, which are generally very helpful for examining the convergence of the AdaSub algorithm. Figure 3.1 and Figure 3.2 show the evolution of $\text{EBIC}_{0.5}(S^{(t)})$ and $r_j^{(t)}$ along the iterations t for different values of the learning rate $K \in \{10, 100, 1000\}$. In all three cases the algorithm identifies the correct model $S_0 = \{1, \ldots, 6\}$

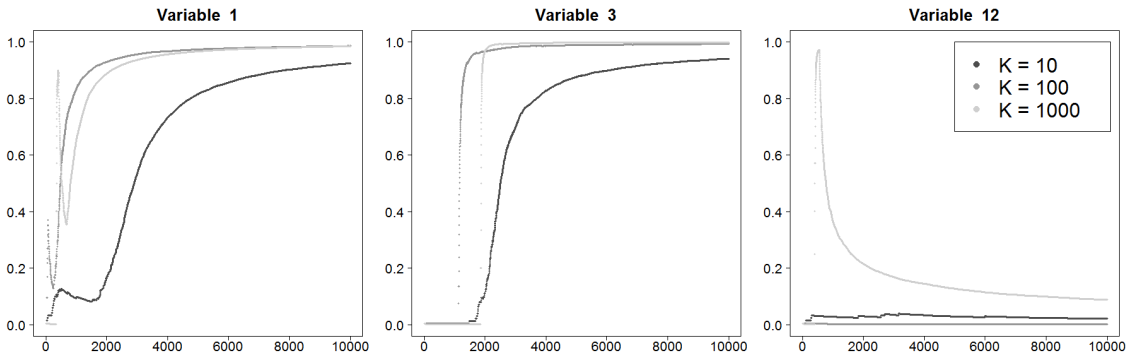


Figure 3.2.: AdaSub for high-dimensional example: Plot of the evolution of $r_j^{(t)}$ for $j \in \{1, 3, 12\}$ along the iterations (t) for different values of K ($q = 5$ fixed).

with $\text{EBIC}_{0.5}(S_0) \approx 371$ based on the "best" sampled model $\hat{S}_\text{b}$ and the thresholded model $\hat{S}_\rho$ for thresholds $\rho \in [0.2, 0.9]$, i.e. we have $S_0 = \hat{S}_\text{b} = \hat{S}_\rho$.

However, there is a trade-off in choosing $K > 0$: If K is small ($K = 10$), then AdaSub adapts slowly to the information learned about the variables and hence a very diverse range of models is considered. If instead K is large ($K = 1000$), then the algorithm might actually adapt the selection probabilities too quickly. Suppose for example that a variable X_j is first considered in the model search at the t-th iteration but not chosen to be in the "best" submodel (i.e. $j \in V^{(t)}$ but $j \notin S^{(t)}$), then $r_j^{(t)} = \frac{q}{p+K} \approx 0$ for K very large, so variable X_j will probably not be considered in the model search for many subsequent iterations. In our case, the choice $K = 100 = n$ seems favourable, for which AdaSub only needed 1123 iterations to find S_0, while it took 2313 iterations for $K = 10$ and 1854 iterations for $K = 1000$ in order to identify S_0 (compare Figure 3.1). Nevertheless, an important observation is that the selected model by AdaSub is stable with respect to changes of K, as long as the number of iterations is large enough.

We now apply AdaSub with $T = 10,000$ iterations on the same dataset as above for different values of $q \in \{2, 5, 10\}$, while $K = 100$ is fixed. Figure 3.3 shows the sizes of the sampled sets $V^{(t)}$ and the sizes of the "best" subsets $S^{(t)} = f_C(V^{(t)})$ along the iterations t, while Figure 3.4 depicts the evolution of the expected sizes of the sets $V^{(t)}$ which are given by $E[|V^{(t)}|] = \sum_{j \in \mathcal{P}} r_j^{(t-1)}$ for $t = 1, \ldots, T$. We can see that the AdaSub algorithm automatically and quickly adjusts the expected search sizes $E[|V^{(t)}|]$ and that the algorithm "converges" against the true underlying model S_0 with six variables, no matter which initial expected search size q is used. Ideally, the tuning parameter q should be chosen in a way

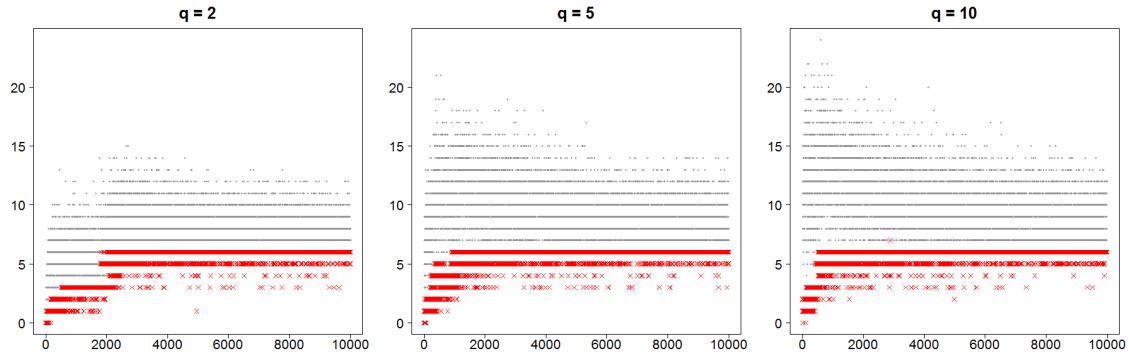


Figure 3.3.: AdaSub for high-dimensional example: Plot of the sizes of the sampled sets $V^{(t)}$ (grey dots) and the sizes of the "best" subsets $S^{(t)}$ (red crosses) along the iterations (t) for different values of q ($K = 100$ fixed).

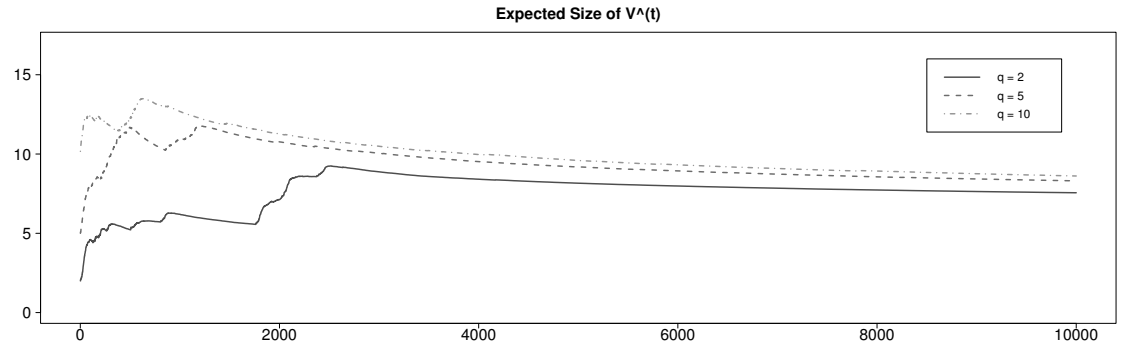


Figure 3.4.: AdaSub for high-dimensional example: Plot of the evolution of the expected search size along the iterations (t) for different values of q ($K = 100$ fixed).

such that it reflects the subjectively expected size of the best model $S^* = f_C(\mathcal{P})$ according to criterion C. However, a general observation is that the choice of the tuning parameter q seems not to be as crucial as the proper choice of the learning rate K.

3.6. Discussion of computational complexity

In this section we discuss the computational cost of the proposed AdaSub method. Note that it is intrinsically difficult to derive general computational complexity results for AdaSub due to the stochastic nature of the algorithm: First, all the sampled subsets $V^{(t)}$ in AdaSub depend on the previously sampled subsets $V^{(i)}$, $i = 1, \ldots, t-1$, and second, the number of iterations T necessary for the "convergence" of the algorithms is random and unknown in general.

For a rough estimate of the computational complexity assume that it takes at most L

operations to compute (an approximation to) $C(V)$ for a subset $V \in \mathcal{M}$. Furthermore, assume that we have $|V^{(t)}| \leq U_C$ for all iterations t of AdaSub. Then the computational complexity of AdaSub with $T \in \mathbb{N}$ iterations when using full enumeration for the sub-problems is roughly bounded by $T \times 2^{U_C} \times L$. On the other hand, the computational complexity of solving the original high-dimensional problem directly by full enumeration is of order $2^p \times L$. Now, if the sizes of sampled sub-problems are relatively small (i.e. $U_C \ll p$ and thus $2^{U_C} \ll 2^p$) and the number of iterations T is not too large (in particular $T \lll 2^p$), then AdaSub yields a significant reduction of the computational complexity in comparison to a full enumeration approach. In particular, if the number of iterations T as well as the number of operations L needed to compute $C(V)$ both scale at most polynomially with p and if the upper bound U_C on the sizes of the sampled models is fixed, then the computational complexity of AdaSub is at most polynomial in p, in contrast to the exponential complexity of a full enumeration approach for the original problem.

While AdaSub is more efficient than a full model enumeration for a given ℓ_0-type selection criterion, ℓ_1-type regularization methods have, due to their convexity, usually a significant smaller computational complexity than AdaSub. For example in the setting of normal linear models with p possible explanatory variables and sample size n, the LARS algorithm for computing (exactly) the whole regularization path of the Lasso has computational complexity $\mathcal{O}(np\min(n,p))$ (Efron et al., 2004)), and coordinate descent algorithms are usually even faster for computing approximate (near-optimal) Lasso solutions (see e.g. Bühlmann and van de Geer, 2011, p. 36-38). Despite the computational advantages of ℓ_1-type methods like the Lasso, various simulation studies in Section 5.1 confirm that the models selected by AdaSub show crucial statistical advantages with respect to the variable selection performance in comparison to the models selected by convex optimization methods. Furthermore, the practical computational time for AdaSub with a decent convergence behaviour is usually not prohibitively large. We are therefore convinced, that the extra computational time spent for AdaSub in comparison to faster competitors can pay off in many practical situations.

Nevertheless, the original AdaSub method might be computationally expensive in certain situations, since the evaluation of the criterion $C(V)$ for single models $V \in \mathcal{M}$ can be quite costly depending on the employed model family. Note that for normal linear models the MLEs of the regression coefficients are given in closed form (see Remark 2.3) and thus

the computation of criterion values like the EBIC for single models is usually fast. In fact, it is easy to see that computing an ℓ_0-type selection criterion $C(V)$ for some $V \in \mathcal{M}$ with $|V| < n - 2$ has complexity $\mathcal{O}(n \times |V|^2)$ for normal linear models of sample size n. Furthermore, very efficient branch-and-bound algorithm (see e.g. Narendra and Fukunaga, 1977 and Lumley and Miller, 2009) can be employed in the setting of normal linear models (compare Section 2.3.1). On the other hand, for GLMs like logistic or Poisson regression models the MLEs of the regression coefficients are generally not given in closed form and therefore each evaluation of $C(V)$ for some subset $V \in \mathcal{M}$ requires the solution of a separate (continuous) optimization problem using numerical methods. This issue is an important motivation for certain modifications of the original AdaSub method which solve the sampled sub-problems by greedy stepwise methods instead of solving them exactly. Details of such variants of AdaSub and their respective properties are presented in Chapter 6.

3.7. Comparison to existing methodology

In this section we give a brief overview of different variable selection methods that are related to AdaSub. In particular, we want to emphasize the novelty of the proposed AdaSub method.

The idea of considering different, randomly chosen subspaces of all available explanatory variables has already appeared in certain parts of the literature: Ho (1998) propose the Random Subspace Method for classification, where multiple regression trees are constructed using randomly sampled subspaces of all explanatory variables; the final classifier is based on a certain form of averaging the decisions from the single trees. The Random Subspace Method has been extended by Lai et al. (2006) to the variable selection context, where certain variable selection methods are subsequently applied to random subspaces and a final ranking of the explanatory variables is obtained by averaging their "scores" (measuring the importance of the respective variables) obtained in the different subspaces. A similar idea is used in Beinrucker et al. (2016), where the original Stability Selection procedure (see Algorithm 2.3) is extended to iteratively apply a baseline method like the Lasso on random submatrices $\boldsymbol{X}_{I,J} \in \mathbb{R}^{|I| \times |J|}$, with $I \subseteq \{1, \ldots, n\}$ and $J \subseteq \{1, \ldots, p\}$, of the original design matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ (i.e. "sub-sampling" both in the sample space and in the feature space).

A Tournament Screening method is proposed by Chen and Chen (2009) where in the first stage all explanatory variables are randomly divided into groups of equal size and then

a regularization method like the Lasso is applied on each group separately. The selected variables from the different groups are pooled together and in the next stage of the algorithm they are again randomly divided into new groups. This procedure is repeated until the dimensionality of the remaining feature space is sufficiently small. Then the EBIC is used for the selection of the final model, again considering only submodels which are obtained by a regularization method like the Lasso.

Wang et al. (2016) propose DECOrrelated feature space partitioning which is based on a single split of the high-dimensional feature space into disjoint subspaces. After a preconditioning step which aims at decorrelating the variables across the partitioned subspaces, a regularization method like the Lasso is applied on the single subspaces for estimation and variable selection. Then it is possible to carry out a succeeding "refinement" step on the aggregated set of selected variables by using Ridge Regression.

Similar ideas have also been considered in the Bayesian variable selection framework: Song and Liang (2015a) propose a two-stage Split-and-Merge (SAM) approach where in the first stage of the algorithm all explanatory variables are divided into disjoint groups and then a fully Bayesian variable selection procedure is applied to each of the different groups. In the second stage the variables selected from each of the groups are aggregated and then a fully Bayesian variable selection procedure is again applied on the aggregated set of variables. A drawback of SAM as well as DECO might be that there is only a single split of the data, so that it might happen that variables in different groups are not selected even though they would have been selected if they were together in one group.

A distinguishing feature of the proposed AdaSub method in comparison to all the methods described above is that — by using a certain form of adaptive stochastic learning — it makes explicit and effective use of the information learned from the subspaces already considered. In particular, the sizes of the sampled subspaces in AdaSub are not fixed in advance but are automatically adapted during the algorithm. Furthermore, AdaSub does not rely on relaxations (e.g. the Lasso with an ℓ_1-penalty) of the original ℓ_0-type problem.

A further relevant method is the PC-simple algorithm for variable selection proposed by Bühlmann et al. (2010), which we want to describe in some more detail since its underlying concept of partial faithfulness is related to the theoretical properties of AdaSub (see Chapter 4 and in particular Definition 4.6). Let $\hat{\rho}(Y, X_j \mid X_S)$ denote the estimated partial correlation

(based on the observed sample) between the response Y and variable X_j given the set of variables $X_S := \{X_k;\ k \in S\}$ for some subset $S \subseteq \mathcal{P}$; in particular $\hat{\rho}(Y, X_j \,|\, X_\emptyset) \equiv \hat{\rho}(Y, X_j)$ denotes the sample correlation between Y and X_j. The PC-simple algorithm builds a "decreasing" sequence of nested subsets

$$\hat{S}^{(1)} \supseteq \hat{S}^{(2)} \supseteq \ldots \supseteq \hat{S}^{(m)}, \tag{3.19}$$

where

$$\hat{S}^{(1)} = \{j \in \mathcal{P};\ \hat{\rho}(Y, X_j) \neq 0\} \tag{3.20}$$

and, for $k = 2, \ldots, m$,

$$\hat{S}^{(k)} = \left\{j \in \mathcal{P};\ \hat{\rho}(Y, X_j \,|\, X_V) \neq 0 \text{ for all } V \subseteq \hat{S}^{(k-1)} \setminus \{j\} \text{ with } |V| = k-1\right\}. \tag{3.21}$$

In equations (3.20) and (3.21), by slight abuse of notation, $\hat{\rho}(Y, X_j \,|\, X_V) \neq 0$ means that a two-sided test based on Fisher's Z-transform rejects the null-hypothesis that the true underlying partial correlation $\rho(Y, X_j \,|\, X_V)$ is equal to zero (see Bühlmann et al., 2010 for details). The number of steps m of the PC-simple algorithm is determined by $m = \min\{k;\ |\hat{S}^{(k)}| \leq k\}$ and then $\hat{S}^{(m)}$ is returned as the final model selected by the algorithm. The PC-simple algorithm is based on the assumption of partial faithfulness which implies that a truly "relevant" variable (i.e. $\rho(Y, X_j \,|\, X_{\mathcal{P}\setminus\{j\}}) \neq 0$) is also "relevant" when conditioning on any subset $V \subseteq \mathcal{P} \setminus \{j\}$ (i.e. $\rho(Y, X_j \,|\, X_V) \neq 0$). For a detailed discussion of partial faithfulness and its connection to AdaSub we refer to Chapter 4.

Another related approach is the variable selection procedure based on Tilting proposed by Cho and Fryzlewicz (2012). It builds an "increasing" sequence of nested subsets

$$\emptyset = \hat{S}^{(0)} \subset \hat{S}^{(1)} \subset \hat{S}^{(2)} \subset \ldots \subset \hat{S}^{(m)}, \tag{3.22}$$

with $|\hat{S}^{(i)}| = i$, for $i = 0, \ldots, m$, by gradually adding explanatory variables based on "tilted" correlations. On an empirical basis, the number of steps m of the Tilting algorithm can be set to $m = \lfloor \frac{n}{2} \rfloor$, where n is the sample size (see Cho and Fryzlewicz, 2012, Section 3.1 for a detailed discussion). The final model $\hat{S} = \arg\min_{\hat{S}^{(i)}} \mathrm{EBIC}_\gamma(\hat{S}^{(i)})$ is selected according to EBIC_γ for some $\gamma \in [0, 1]$, when applied to the generated sequence (3.22) of nested models. The main difference of Tilting in comparison to the PC-simple algorithm is that the "conditioning sets" for the "tilted" correlations are somehow adaptively chosen for each

explanatory variable. Cho and Fryzlewicz (2012) demonstrate that both proposed versions (TCS1 and TCS2) of Tilting often outperform the PC-simple algorithm.

Even though the PC-simple algorithm as well as the Tilting approach are similar to AdaSub in the sense that they also consider the contribution of the explanatory variables to the response conditionally on different subspaces, we want to stress that these approaches are fundamentally different in comparison to AdaSub, since they only consider (deterministic) sequences of nested models (see equations (3.19) and (3.22)), while AdaSub is a stochastic algorithm which randomly samples more and more "interesting" subspaces. Therefore, the PC-simple algorithm and Tilting are not as adaptive as AdaSub in the choice of the "conditioning sets" for the partial correlations. In particular, in the PC-simple algorithm variables that are not included in some $\hat{S}^{(i)}$ cannot be be included in any $\hat{S}^{(k)}$ for $k > i$, while in Tilting variables that are included in some $\hat{S}^{(i)}$ will also be included in any $\hat{S}^{(k)}$ for $k > i$. These properties of the PC-simple algorithm and Tilting are similar to the properties of Backward and Forward Stepwise Selection, respectively (compare Section 2.3.2).

A distinguishing feature of AdaSub is that it aims at identifying the best model according to a given criterion (including ℓ_0-type selection criteria like the EBIC) by exactly solving low-dimensional sub-problems of the original (generally NP-hard) high-dimensional problem. In the subsequent Chapter 4 we will show that, under certain conditions, AdaSub is guaranteed to converge against the best model according to the criterion used (see Theorem 4.8). Furthermore, we will argue that in a situation in which AdaSub does not converge against the best model according to the employed criterion, this fact actually provides further information that this particular model is not "stable" in a certain sense and that a thresholded model from AdaSub might be preferred (see Theorem 4.12 and related discussions). To the best of our knowledge, we are not aware of similar convergence results for other stochastic algorithms in the variable selection context.

4. Theoretical results for AdaSub

In this chapter we want to investigate the limiting properties of AdaSub (Algorithm 3.1 of Section 3.2) by analysing the evolution of the selection probabilities $r_j^{(t)}$, given in equation (3.3), along the iterations $t \in \mathbb{N}$. For this purpose, we introduce the following shorthand notation.

Notation 4.1. The selection probabilities in AdaSub after iteration $t \in \mathbb{N}$ are given by

$$r_j^{(t)} = \frac{q + K \sum_{i=1}^{t} W_j^{(i)}}{p + K \sum_{i=1}^{t} Z_j^{(i)}}\,, \tag{4.1}$$

where $Z_j^{(i)} = \mathbb{1}_{V^{(i)}}(j)$ and $W_j^{(i)} = \mathbb{1}_{S^{(i)}}(j)$ are indicator variables for the inclusion of variable X_j in the sampled subspace $V^{(i)} \subseteq \mathcal{P}$ and in the "best" submodel $S^{(i)} = f_C(V^{(i)}) \in \mathcal{M}$, respectively, for $j \in \mathcal{P}$, $i \in \mathbb{N}$.

Note that in this chapter we will assume throughout that the original AdaSub method (Algorithm 3.1) is used. Furthermore, for ease of exposition we make the assumption given in equation (3.2) of Notation 3.1, that $C(S) \neq C(S')$ for all $S, S' \in \mathcal{M}$ with $S \neq S'$. In particular, this assumption implies that the C-optimal model S^* is unique. A brief discussion of the situation in which this assumption does not hold is provided in Remark 4.4 at the end of Section 4.1.

The aim of AdaSub is to identify the best model S^* according to the employed selection criterion C. Thus, we give the following definition for the correct convergence of AdaSub.

Definition 4.2. The AdaSub algorithm is said to converge correctly if and only if for all $j \in \mathcal{P}$ we have

$$r_j^{(t)} \overset{\text{a.s.}}{\to} \begin{cases} 1 & \text{, if } j \in S^*, \\ 0 & \text{, if } j \notin S^*, \end{cases} \qquad \text{for } t \to \infty\,, \tag{4.2}$$

where $S^* = f_C(\mathcal{P}) = \arg\max_{S \in \mathcal{M}} C(S)$ denotes the C-optimal model, i.e. the best model according to the used selection criterion C.

The definition of correct convergence can intuitively be understood via the (naive) Bayesian interpretation of AdaSub in Section 3.4. According to this interpretation, one might guess that for each variable X_j the corresponding expectation $r_j^{(t)}$ of the "posterior inclusion probability" π_j converges against its "true" value as we gather more and more (pseudo) data $\mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(t)}$ (as $t \to \infty$). However, the correct convergence of AdaSub cannot be expected in general, since the datasets $\mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(t)}$ are not independent and the Bayesian motivation of AdaSub does not directly correspond to a formal full Bayesian analysis. It is an important target of this chapter to give a sufficient condition under which the correct convergence of AdaSub can be guaranteed.

In Section 4.1 we present theoretical results concerning the limiting behaviour of AdaSub. In particular, we show that, under the so-called ordered importance property (OIP), AdaSub converges correctly in the sense of Definition 4.2 (see Theorem 4.8). Furthermore, if AdaSub does not converge correctly, we propose to select a thresholded model obtained by AdaSub which can provide a more "stable" set of variables with less false positive selections than the C-optimal model itself (see Theorem 4.12 and related discussions). In Section 4.2 we illustrate the convergence properties of AdaSub by a simulated data example. Finally, in Section 4.3 we discuss the variable selection consistency of AdaSub, i.e. we address the question under which conditions it can be guaranteed that (asymptotically, as the sample size n tends to infinity and under appropriate conditions on the number of variables p) AdaSub converges correctly against the true underlying model $S_0 = \{j \in \mathcal{P};\ \beta_{0,j} \neq 0\}$.

Note that parts of the material presented in Sections 4.1 and 4.2 have been submitted for publication in Staerk et al. (2018).

4.1. Limiting properties of AdaSub

In order to describe the information available after iteration t of the AdaSub algorithm, we define a filtration $(\mathcal{F}^{(t)})_{t \in \mathbb{N}_0}$ on the underlying probability space Ω of the process.

Notation 4.3. Let $\mathcal{F}^{(0)} := \{\emptyset, \Omega\}$ and for $t \in \mathbb{N}$ let

$$\mathcal{F}^{(t)} := \sigma(W_1^{(1)}, Z_1^{(1)}, W_2^{(1)}, Z_2^{(1)}, \ldots, W_p^{(1)}, Z_p^{(1)}, \ldots, W_p^{(t)}, Z_p^{(t)}) \tag{4.3}$$

be the σ-algebra generated by $W_1^{(1)}, \ldots, Z_p^{(t)}$. Then by the construction of AdaSub, for

$t \in \mathbb{N}_0$ and $j \in \mathcal{P}$, the sampling probability of variable X_j in iteration $t+1$ (conditional on the information from the first t iterations) is given by

$$r_j^{(t)} = P(Z_j^{(t+1)} = 1 \mid \mathcal{F}^{(t)}) = 1 - P(Z_j^{(t+1)} = 0 \mid \mathcal{F}^{(t)}). \tag{4.4}$$

In addition, for $t \in \mathbb{N}_0$ and $j \in \mathcal{P}$, the conditional probability that variable X_j is selected to be in the "best" submodel $S^{(t+1)} = f_C(V^{(t+1)})$ provided that it is included in $V^{(t+1)}$ is given by

$$p_j^{(t+1)} := P(W_j^{(t+1)} = 1 \mid Z_j^{(t+1)} = 1, \mathcal{F}^{(t)}) = 1 - P(W_j^{(t+1)} = 0 \mid Z_j^{(t+1)} = 1, \mathcal{F}^{(t)}), \tag{4.5}$$

where, throughout the rest of this thesis, for events $A, B \in \mathcal{F}^{(t+1)}$ the conditional probabilities under $\mathcal{F}^{(t)}$ are defined by

$$P(A \,|\, \mathcal{F}^{(t)}) := E[\mathbb{1}_A \,|\, \mathcal{F}^{(t)}] \tag{4.6}$$

and

$$P(A \,|\, B, \mathcal{F}^{(t)}) := \frac{E[\mathbb{1}_{A \cap B} \,|\, \mathcal{F}^{(t)}]}{E[\mathbb{1}_B \,|\, \mathcal{F}^{(t)}]} \tag{4.7}$$

almost surely on $\{E[\mathbb{1}_B \,|\, \mathcal{F}^{(t)}] > 0\}$, while we set $P(A \,|\, B, \mathcal{F}^{(t)}) = 0$ almost surely on $\{E[\mathbb{1}_B \,|\, \mathcal{F}^{(t)}] = 0\}$.

In order to analyse the limiting behaviour of AdaSub, we will make repeated use of the following generalization of Borel-Cantelli's lemma and the strong law of large numbers, which is due to Dubins and Freedman (1965).

Theorem 4.1 (Dubins and Freedman, 1965)**.** *Let $(\mathcal{F}_n)_{n \in \mathbb{N}_0}$ be a filtration and let $A_n \in \mathcal{F}_n$ be events for $n \in \mathbb{N}$. For $i \in \mathbb{N}$ we define random variables $q_i := P(A_i \,|\, \mathcal{F}_{i-1})$. Then:*

(a) On $\{\sum_{i=1}^{\infty} q_i < \infty\}$ we almost surely have $\sum_{i=1}^{\infty} \mathbb{1}_{A_i} < \infty$.

(b) On $\{\sum_{i=1}^{\infty} q_i = \infty\}$ we have

$$\frac{\sum_{i=1}^{n} \mathbb{1}_{A_i}}{\sum_{i=1}^{n} q_i} \xrightarrow{a.s.} 1, \ n \to \infty\,.$$

A first simple but important observation is that, with probability 1, each variable X_j with $j \in \mathcal{P}$ is considered infinitely many times in the model search of AdaSub (provided that the number of iterations T in AdaSub tends to infinity).

Lemma 4.2. *Let $j \in \mathcal{P}$. Then we have*

$$P\left(\sum_{t=1}^{\infty} \mathbb{1}_{V^{(t)}}(j) = \infty\right) = P\left(\sum_{t=1}^{\infty} Z_j^{(t)} = \infty\right) = 1\,. \tag{4.8}$$

Proof. We apply Theorem 4.1: Let $(\mathcal{F}^{(t)})_{t\in\mathbb{N}_0}$ be the filtration given in Notation 4.3. Fix $j \in \mathcal{P}$ and for $t \in \mathbb{N}$ let $A_j^{(t)} := \{Z_j^{(t)} = 1\} \in \mathcal{F}^{(t)}$. For $t \in \mathbb{N}$ we have

$$q_j^{(t)} := P(A_j^{(t)} \mid \mathcal{F}^{(t-1)}) = r_j^{(t-1)} = \frac{q + K\sum_{i=1}^{t-1} \mathbb{1}_{S^{(i)}}(j)}{p + K\sum_{i=1}^{t-1} \mathbb{1}_{V^{(i)}}(j)} \geq \frac{q}{p + K(t-1)}$$

and therefore $\sum_{i=1}^{\infty} q_j^{(i)} \overset{\text{a.s.}}{=} \infty$. So by Theorem 4.1 we conclude that

$$\frac{\sum_{i=1}^{t} \mathbb{1}_{A_j^{(i)}}}{\sum_{i=1}^{t} q_j^{(i)}} \xrightarrow{\text{a.s.}} 1,\ t \to \infty\,.$$

Since $\sum_{i=1}^{\infty} q_j^{(i)} \overset{\text{a.s.}}{=} \infty$, we also have

$$\sum_{i=1}^{\infty} \mathbb{1}_{A_j^{(i)}} = \sum_{i=1}^{\infty} Z_j^{(i)} \overset{\text{a.s.}}{=} \infty\,.$$

□

It is illuminating to take a closer look at the conditional expectation $E\left[r_j^{(t)} \mid \mathcal{F}^{(t-1)}\right]$ of the selection probability of variable X_j in iteration $t+1$ given the information after $t-1$ iterations. It can be shown that one obtains the following compact expression in terms of the probability $p_j^{(t)}$ (compare Notation 4.3).

Lemma 4.3. *For $j \in \mathcal{P}$ and $t \in \mathbb{N}$ we have*

$$E\left[r_j^{(t)} \,\middle|\, \mathcal{F}^{(t-1)}\right] \overset{a.s.}{=} r_j^{(t-1)}\left(1 + C_j^{(t)}\left(p_j^{(t)} - r_j^{(t-1)}\right)\right), \tag{4.9}$$

where

$$C_j^{(t)} = \frac{K}{p + K\sum_{i=1}^{t-1} Z_j^{(i)} + K} \in (0,1)\,. \tag{4.10}$$

Therefore: $E\left[r_j^{(t)} \,\middle|\, \mathcal{F}^{(t-1)}\right] \overset{a.s.}{\geq} r_j^{(t-1)}$ iff $p_j^{(t)} \overset{a.s.}{\geq} r_j^{(t-1)}$.

Proof. Let $j \in \mathcal{P}$ and $t \in \mathbb{N}$. Then it holds almost surely:

$$\begin{aligned}
&E\left[r_j^{(t)} \,\middle|\, \mathcal{F}^{(t-1)}\right] \\
&= r_j^{(t-1)} P\left(Z_j^{(t)} = 0 \,\middle|\, \mathcal{F}^{(t-1)}\right) + \frac{q + K\sum_{i=1}^{t-1} W_j^{(i)}}{p + K\sum_{i=1}^{t-1} Z_j^{(i)} + K} \times P\left(Z_j^{(t)} = 1, W_j^{(t)} = 0 \,\middle|\, \mathcal{F}^{(t-1)}\right) \\
&\quad + \frac{q + K\sum_{i=1}^{t-1} W_j^{(i)} + K}{p + K\sum_{i=1}^{t-1} Z_j^{(i)} + K} \times P\left(Z_j^{(t)} = 1, W_j^{(t)} = 1 \,\middle|\, \mathcal{F}^{(t-1)}\right)
\end{aligned}$$

$$
\begin{aligned}
&= r_j^{(t-1)}\left(1-r_j^{(t-1)}\right) + r_j^{(t-1)}\left(1-C_j^{(t)}\right) P\left(W_j^{(t)}=0 \,\middle|\, Z_j^{(t)}=1, \mathcal{F}^{(t-1)}\right) P\left(Z_j^{(t)}=1 \,\middle|\, \mathcal{F}^{(t-1)}\right) \\
&\quad + \left(r_j^{(t-1)}\left(1-C_j^{(t)}\right) + C_j^{(t)}\right) P\left(W_j^{(t)}=1 \,\middle|\, Z_j^{(t)}=1, \mathcal{F}^{(t-1)}\right) P\left(Z_j^{(t)}=1 \,\middle|\, \mathcal{F}^{(t-1)}\right) \\
&= r_j^{(t-1)}\left(1-r_j^{(t-1)}\right) + r_j^{(t-1)}\left(1-C_j^{(t)}\right)\left(1-p_j^{(t)}\right) r_j^{(t-1)} \\
&\quad + \left(r_j^{(t-1)}\left(1-C_j^{(t)}\right) + C_j^{(t)}\right) p_j^{(t)} r_j^{(t-1)} \\
&= r_j^{(t-1)}\left(1 + C_j^{(t)}\left(p_j^{(t)} - r_j^{(t-1)}\right)\right)
\end{aligned}
$$

□

The result of Lemma 4.3 indicates that the selection probability $r_j^{(t)}$ of a variable X_j being considered in the model search in iteration $t+1$ automatically adjusts to the conditional probability $p_j^{(t)}$ that variable X_j is selected to be in the best submodel given that X_j is considered in iteration t. Actually, we shall prove that the convergence of $p_j^{(t)}$ as $t \to \infty$ determines the convergence of $r_j^{(t)}$ (see Theorem 4.5). Before this, we first show a weaker result by making use of the expression (4.9) for the conditional expectation $E\left[r_j^{(t)} \,|\, \mathcal{F}^{(t-1)}\right]$ from Lemma 4.3, which provides some further insights and might be interesting in itself.

Lemma 4.4. *Let $j \in \mathcal{P}$. If $p_j^{(t)} \overset{t\to\infty}{\to} p_j^*$ (a.s.) for some random variable p_j^* with values in $[0,1]$, then, with probability one, p_j^* is an accumulation (AC) point of $\left(r_j^{(t)}\right)_{t\in\mathbb{N}_0}$.*

Proof. Let

$$
A := \left\{\omega \in \Omega;\ p_j^*(\omega) \text{ is not an AC point of } \left(r_j^{(t)}(\omega)\right)_{t\in\mathbb{N}_0}\right\}
$$

and suppose that $P(A) > 0$.

For $\epsilon > 0$ and $T_0 \in \mathbb{N}$, we define

$$
\begin{aligned}
A_{T_0,\epsilon} &:= \{\omega \in \Omega;\ \forall t \geq T_0 : |p_j^*(\omega) - r_j^{(t)}(\omega)| > \epsilon\} \\
&= \underbrace{\{\omega \in \Omega;\ \forall t \geq T_0 : p_j^*(\omega) - r_j^{(t)}(\omega) > \epsilon\}}_{=:A^{>}_{T_0,\epsilon}} \cup \underbrace{\{\omega \in \Omega;\ \forall t \geq T_0 : r_j^{(t)}(\omega) - p_j^*(\omega) > \epsilon\}}_{=:A^{<}_{T_0,\epsilon}}.
\end{aligned}
$$

Then we have

$$
A = \bigcup_{n\in\mathbb{N}} \bigcup_{T_0\in\mathbb{N}} A_{T_0,\frac{1}{n}} = \bigcup_{n\in\mathbb{N}} \bigcup_{T_0\in\mathbb{N}} \left(A^{>}_{T_0,\frac{1}{n}} \cup A^{<}_{T_0,\frac{1}{n}}\right). \tag{4.11}
$$

Since $P(A) > 0$ and since the union in (4.11) is countable, there exist $t_0 \in \mathbb{N}, n_0 \in \mathbb{N}$ such that $P(A_{t_0,\epsilon_0}) > 0$, where $\epsilon_0 := \frac{1}{n_0} > 0$. Therefore we have $P(A^{>}_{t_0,\epsilon_0}) > 0$ or $P(A^{<}_{t_0,\epsilon_0}) > 0$. For simplicity, we suppose that $P(A^{>}_{t_0,\epsilon_0}) > 0$ (the case $P(A^{<}_{t_0,\epsilon_0}) > 0$ can be treated analogously).

Let $B := \{\omega \in \Omega;\ p_j^{(t)}(\omega) \overset{t\to\infty}{\to} p_j^*(\omega)\}$. Then by assumption we have $P(B) = 1$.

For $\epsilon > 0$ and $T_0 \in \mathbb{N}$ we define

$$B_{T_0,\epsilon} := \{\omega \in \Omega;\ \forall t \geq T_0 : |p_j^*(\omega) - p_j^{(t+1)}(\omega)| \leq \epsilon\}.$$

Then $B = \bigcap_{n\in\mathbb{N}} \bigcup_{T_0\in\mathbb{N}} B_{T_0,\frac{1}{n}}$, so that $B \subseteq \bigcup_{T_0\in\mathbb{N}} B_{T_0,\frac{\epsilon_0}{2}} =: \widetilde{B}$. Since $P(B) = 1$, we also have $P(\widetilde{B}) = 1$, so that

$$0 < P(A^{>}_{t_0,\epsilon_0}) = P(\widetilde{B} \cap A^{>}_{t_0,\epsilon_0}) = P\left(\bigcup_{T_0\in\mathbb{N}} \left(B_{T_0,\frac{\epsilon_0}{2}} \cap A^{>}_{t_0,\epsilon_0}\right)\right).$$

Therefore there exists $t_1 \in \mathbb{N}$ such that $P\left(B_{t_1,\frac{\epsilon_0}{2}} \cap A^{>}_{t_0,\epsilon_0}\right) > 0$.

With $t_2 := \max\{t_0, t_1\}$ we have

$$\begin{aligned} B_{t_1,\frac{\epsilon_0}{2}} \cap A^{>}_{t_0,\epsilon_0} &\subseteq \left\{\omega \in \Omega;\ \forall t \geq t_2 : p_j^*(\omega) - r_j^{(t)}(\omega) > \epsilon_0 \text{ and } |p_j^*(\omega) - p_j^{(t+1)}(\omega)| \leq \frac{\epsilon_0}{2}\right\} \\ &\subseteq \left\{\omega \in \Omega;\ \forall t \geq t_2 : p_j^{(t+1)}(\omega) - r_j^{(t)}(\omega) > \frac{\epsilon_0}{2}\right\} =: D\,. \end{aligned}$$

Here, we used the fact that $p_j^*(\omega) - r_j^{(t)}(\omega) > \epsilon_0$ and $|p_j^*(\omega) - p_j^{(t+1)}(\omega)| \leq \frac{\epsilon_0}{2}$ implies

$$p_j^{(t+1)}(\omega) - r_j^{(t)}(\omega) = \underbrace{p_j^{(t+1)}(\omega) - p_j^*(\omega)}_{\geq -\epsilon_0/2} + \underbrace{p_j^*(\omega) - r_j^{(t)}(\omega)}_{>\epsilon_0} > \epsilon_0 - \frac{\epsilon_0}{2} = \frac{\epsilon_0}{2}\,.$$

Since $P(B_{t_1,\frac{\epsilon_0}{2}} \cap A^{>}_{t_0,\epsilon_0}) > 0$, we also have $P(D) > 0$. Now for $t \geq t_2$ it holds almost surely on D:

$$\begin{aligned} E\left[r_j^{(t+2)} \,\middle|\, \mathcal{F}^{(t)}\right] &= E\left[E\left[r_j^{(t+2)} \,\middle|\, \mathcal{F}^{(t+1)}\right] \,\middle|\, \mathcal{F}^{(t)}\right] \\ &\overset{(4.9)}{=} E\left[r_j^{(t+1)}\left(1 + C_j^{(t+2)}\left(p_j^{(t+2)} - r_j^{(t+1)}\right)\right) \,\middle|\, \mathcal{F}^{(t)}\right] \\ &\geq \left(1 + \frac{K}{p + K(t+2)}\frac{\epsilon_0}{2}\right) E\left[r_j^{(t+1)} \,\middle|\, \mathcal{F}^{(t)}\right] \\ &\overset{(4.9)}{\geq} \left(1 + \frac{K}{p + K(t+2)}\frac{\epsilon_0}{2}\right)\left(1 + \frac{K}{p + K(t+1)}\frac{\epsilon_0}{2}\right) r_j^{(t)} \\ &\geq r_j^{(t)}\left(1 + \frac{\epsilon_0}{2}\left(\frac{K}{p + K(t+1)} + \frac{K}{p + K(t+2)}\right)\right). \end{aligned}$$

By induction on $k \in \mathbb{N}$ we similarly obtain for $t \geq t_2$ almost surely on D:

$$\begin{aligned} E\left[r_j^{(t+k)} \,\middle|\, \mathcal{F}^{(t)}\right] &\geq r_j^{(t)}\left(1 + K\frac{\epsilon_0}{2}\sum_{l=1}^{k}\frac{1}{p + K(t+l)}\right) \\ &\geq \frac{q}{p + Kt}\left(1 + K\frac{\epsilon_0}{2}\sum_{l=1}^{k}\frac{1}{p + K(t+l)}\right) \overset{k\to\infty}{\to} \infty\,. \end{aligned}$$

So there exists $k_0 \in \mathbb{N}$ such that $E\left[r_j^{(t_2+k_0)} \,\middle|\, \mathcal{F}^{(t_2)}\right] > 1$ almost surely on D. Now we have $r_j^{(t_2+k_0)} \in (0,1)$ by definition and with $P(D) > 0$ we obtain the contradiction. Therefore we must have $P(A) = 0$, which proves the claim of the lemma. □

Theorem 4.5. *For each $j \in \mathcal{P}$ we have: If $p_j^{(t)} \overset{a.s.}{\to} p_j^*$ as $t \to \infty$ for some random variable p_j^*, then $r_j^{(t)} \overset{a.s.}{\to} p_j^*$ as $t \to \infty$.*

Proof. Fix $j \in \mathcal{P}$ and suppose that $p_j^{(t)} \overset{\text{a.s.}}{\to} p_j^*$ as $t \to \infty$. We apply Theorem 4.1 again, but using a different filtration $(\mathcal{G}^{(t)})_{t\in\mathbb{N}_0}$, where

$$\mathcal{G}^{(0)} = \sigma\left(\left\{Z_j^{(1)};\ j \in \mathcal{P}\right\}\right),$$

and

$$\mathcal{G}^{(t)} = \sigma\left(\left\{Z_j^{(i)};\ j \in \mathcal{P}, i = 1,\dots,t+1\right\} \cup \left\{W_j^{(i)};\ j \in \mathcal{P}, i = 1,\dots,t\right\}\right),\ t \in \mathbb{N}.$$

Further let $A_j^{(t)} := \{W_j^{(t)} = 1\} \in \mathcal{G}^{(t)}$, for $t \in \mathbb{N}$, with

$$\begin{aligned} q_j^{(t)} :=& P\left(A_j^{(t)} \mid \mathcal{G}^{(t-1)}\right) = P\left(W_j^{(t)} = 1 \mid \mathcal{G}^{(t-1)}\right) \\ =& \underbrace{P\left(W_j^{(t)} = 1 \mid Z_j^{(t)} = 1, \mathcal{G}^{(t-1)}\right)}_{=p_j^{(t)}} Z_j^{(t)} + \underbrace{P\left(W_j^{(t)} = 1 \mid Z_j^{(t)} = 0, \mathcal{G}^{(t-1)}\right)}_{=0}\left(1 - Z_j^{(t)}\right) \\ =& p_j^{(t)} Z_j^{(t)}. \end{aligned}$$

Define

$$\Omega' := \left\{\omega \in \Omega;\ \sum_{i=1}^{\infty} Z_j^{(i)}(\omega) = \infty\right\}.$$

By Lemma 4.2 we have $P(\Omega') = 1$. Let

$$\Omega_1 := \{\omega \in \Omega';\ p_j^{(t)}(\omega) \to p_j^*(\omega), t \to \infty \text{ with } p_j^*(\omega) \in (0,1]\}$$

and

$$\Omega_2 := \{\omega \in \Omega';\ p_j^{(t)}(\omega) \to p_j^*(\omega), t \to \infty \text{ with } p_j^*(\omega) = 0\}.$$

Then on Ω_1 we have

$$\sum_{i=1}^{\infty} q_j^{(i)} = \sum_{i=1}^{\infty} p_j^{(i)} Z_j^{(i)} \overset{\text{(a1)}}{=} \sum_{i=1}^{\infty} p_j^{(l_i^\omega)} = \infty,$$

where equality in (a1) holds since for each $\omega \in \Omega_1$ there exists an increasing sequence $(l_i^\omega)_{i\in\mathbb{N}}$ with $l_i^\omega \in \mathbb{N}$ and $Z_j^{(l_i^\omega)}(\omega) = 1$ for all $i \in \mathbb{N}$. Furthermore, we have used the fact that on Ω_1

we have $p_j^{(l_i^\omega)} \to p_j^* > 0$ as $i \to \infty$, and therefore $\sum_{i=1}^\infty p_j^{(l_i^\omega)} = \infty$. So on Ω_1 we have for t large enough (to avoid division by 0)

$$\lim_{t\to\infty} \frac{\sum_{i=1}^t q_j^{(i)}}{\sum_{i=1}^t Z_j^{(i)}} = \lim_{t\to\infty} \frac{\sum_{i=1}^t p_j^{(i)} Z_j^{(i)}}{\sum_{i=1}^t Z_j^{(i)}} = \lim_{t\to\infty} \frac{\sum_{i=1}^t p_j^{(l_i^\omega)}}{t} = p_j^*\,,$$

which holds for those increasing sequences $(l_i^\omega)_{i\in\mathbb{N}}$ that additionally fulfil $Z_j^{(i)}(\omega) = 0$ for all $i \notin \{l_k^{(\omega)};\ k \in \mathbb{N}\}$. Here we applied Cauchy's limit theorem using the fact that $p_j^{(l_i^\omega)} \to p_j^*$ as $i \to \infty$. Combining this result with Theorem 4.1 it follows that on Ω_1 we have (for t large enough)

$$\frac{\sum_{i=1}^t W_j^{(i)}}{\sum_{i=1}^t Z_j^{(i)}} = \underbrace{\frac{\sum_{i=1}^t W_j^{(i)}}{\sum_{i=1}^t q_j^{(i)}}}_{\xrightarrow{\text{a.s.}} 1} \underbrace{\frac{\sum_{i=1}^t q_j^{(i)}}{\sum_{i=1}^t Z_j^{(i)}}}_{\xrightarrow{\text{a.s.}} p_j^*} \xrightarrow{\text{a.s.}} p_j^*, \quad t \to \infty\,.$$

Now on $\Omega_2 \cap \left\{\sum_{i=1}^\infty q_j^{(i)} = \infty\right\}$ we can use the same argument as above and obtain

$$\frac{\sum_{i=1}^t W_j^{(i)}}{\sum_{i=1}^t Z_j^{(i)}} \xrightarrow{\text{a.s.}} p_j^*, \quad t \to \infty\,.$$

On $\Omega_2 \cap \left\{\sum_{i=1}^\infty q_j^{(i)} < \infty\right\}$ we almost surely have $\sum_{i=1}^\infty W_j^{(i)} < \infty$ by Theorem 4.1, but since $\sum_{i=1}^t Z_j^{(i)} \xrightarrow{\text{a.s.}} \infty$ it also follows that

$$\frac{\sum_{i=1}^t W_j^{(i)}}{\sum_{i=1}^t Z_j^{(i)}} \xrightarrow{\text{a.s.}} 0 = p_j^*, \quad t \to \infty\,.$$

Noting that $P(\Omega_1 \cup \Omega_2) = 1$ by assumption and combining the arguments on Ω_1 and Ω_2, we conclude that on Ω we have

$$r_j^{(t)} = \frac{q + K\sum_{i=1}^t W_j^{(i)}}{p + K\sum_{i=1}^t Z_j^{(i)}} = \frac{\frac{q}{K\sum_{i=1}^t Z_j^{(i)}} + \frac{\sum_{i=1}^t W_j^{(i)}}{\sum_{i=1}^t Z_j^{(i)}}}{\frac{p}{K\sum_{i=1}^t Z_j^{(i)}} + 1} \xrightarrow{\text{a.s.}} p_j^*, \quad t \to \infty\,.$$

□

Up to this point, we have only considered the evolution of the individual selection probabilities $r_j^{(t)}$ under the assumption that the limiting behaviour of the conditional probabilities $p_j^{(t)}$ is known. In order to investigate the behaviour of $p_j^{(t)}$, we first summarize some simple properties of the operator

$$f_C : \mathfrak{P}(\{1,\ldots,p\}) \to \mathcal{M},\ f_C(V) := \operatorname*{arg\,max}_{S\subseteq V,\, S\in\mathcal{M}} C(S)\,,$$

which projects a set $V \subseteq \mathcal{P}$ to the best subset $f_C(V) \in \mathcal{M}$ with respect to some selection criterion $C : \mathcal{M} \to \mathbb{R}$.

Lemma 4.6. *The map f_C enjoys the following properties:*

(a) $f_C(V) \subseteq V$ for all $V \subseteq \mathcal{P}$.

(b) $f_C(\mathcal{P}) = S^*$.

(c) $f_C(V) = S^*$ if and only if $V \supseteq S^*$.

(d) If $f_C(V) \subseteq V' \subseteq V$ with $V, V' \subseteq \mathcal{P}$, then we have $f_C(V') = f_C(V)$.

Proof. The properties immediately follow from the definitions of the map f_C and the C-optimal model S^*. □

These simple properties of f_C will be used throughout the rest of this thesis, often without explicit mention. Note that in general $V' \subseteq V$ does not imply $f_C(V') \subseteq f_C(V)$, which obviously complicates the analysis. However, property (c) of Lemma 4.6 gives already a hint on how to find a sufficient condition for the correct convergence of AdaSub: In the limit, we just need to sample subsets $V^{(t)} \subseteq \mathcal{P}$ with $V^{(t)} \supseteq S^*$ (i.e. $V^{(t)}$ should include all the "relevant" variables in S^*), so that we have $f_C(V^{(t)}) = S^*$. This desirable behaviour of the algorithm can be achieved by imposing the so called ordered importance property (OIP), which we introduce next.

Definition 4.4. Given that data $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$ is observed, let $C_{\mathcal{D}} : \mathcal{M} \to \mathbb{R}$ be a selection criterion with C-optimal model $S^* = f_C(\mathcal{P}) = \{j_1, \ldots, j_{s^*}\}$ of size $s^* = |S^*|$. Then the selection criterion C is said to fulfil the *ordered importance property (OIP)* for the sample $\mathcal{D}$, if there exists a permutation $(k_1, \ldots, k_{s^*})$ of $(j_1, \ldots, j_{s^*})$ such that for each $i = 1, \ldots, s^* - 1$ it holds

$$k_i \in f_C(V) \quad \text{for all } V \subseteq \mathcal{P} \setminus N_{i-1} \text{ with } \{k_1, \ldots, k_i\} \subseteq V\,, \tag{4.12}$$

where

$$N_0 := \{j \in \mathcal{P};\ j \notin f_C(V) \text{ for all } V \subseteq \mathcal{P}\} \tag{4.13}$$

and

$$N_i := \{j \in \mathcal{P};\ j \notin f_C(V) \text{ for all } V \subseteq \mathcal{P} \setminus N_{i-1} \text{ with } \{k_1, \ldots, k_i\} \subseteq V\}\,. \tag{4.14}$$

Remark 4.1. Note that $S^* = f_C(V)$ for all $V \subseteq \mathcal{P}$ with $S^* \subseteq V$ (see Lemma 4.6 (c)). Therefore, (4.12) always holds for $i = s^*$ since $k_{s^*} \in S^*$. Furthermore, we have

$$N_{s^*} = \{j \in \mathcal{P};\ j \notin f_C(V) \text{ for all } V \subseteq \mathcal{P} \setminus N_{s^*-1} \text{ with } S^* \subseteq V\} = \mathcal{P} \setminus S^* \,. \tag{4.15}$$

Note that OIP requires only the existence of such a permutation of the variables with indices in S^* and not its identification or uniqueness. So in order to guarantee that OIP holds, we do not need to know any concrete permutation, but only that such a permutation exists. On the other hand, this condition cannot be easily checked, since we do not know the set S^*, which AdaSub actually tries to identify.

The idea behind the ordered importance property (OIP) is connected to the concept of partial faithfulness (PF) underlying the PC-simple algorithm of Bühlmann et al. (2010) for variable selection in linear regression models (compare Section 3.7). In order to describe the concept of partial faithfulness, we introduce some additional notation for the setting of a random design.

Notation 4.5. In a random design setting for the linear model, let $\boldsymbol{X} = (X_1, \ldots, X_p)^T$ denote the vector of random explanatory variables with finite expectation $E(\boldsymbol{X}) = \boldsymbol{\mu_X} \in \mathbb{R}^p$ and finite covariance matrix $\mathrm{Cov}(\boldsymbol{X}) = \boldsymbol{\Sigma_X} \in \mathbb{R}^{p \times p}$. Let $(X_1, \ldots, X_p, Y)$ be the joint (random) vector of the explanatory variables $X_1, \ldots, X_p$ and the response variable Y, where the relationship between the response Y and the explanatory variables $X_1, \ldots, X_p$ is given by the linear model

$$Y = \mu + \sum_{j=1}^{p} \beta_{0,j} X_j + \epsilon \,, \tag{4.16}$$

where μ is the intercept and ϵ is the error with $E(\epsilon) = 0$, $\mathrm{Var}(\epsilon) = \sigma^2 > 0$ and ϵ is independent of $\boldsymbol{X}$. For a subset $S \subseteq \mathcal{P}$, let $\rho(Y, X_j \,|\, X_S)$ denote the partial correlation between Y and X_j given the set of variables $X_S := \{X_k;\ k \in S\}$. In particular, $\rho(Y, X_j \,|\, X_\emptyset) \equiv \rho(Y, X_j)$ denotes the marginal correlation between Y and X_j.

Definition 4.6 (Bühlmann et al., 2010)**.** Given the setting of Notation 4.5, the joint distribution of $(X_1, \ldots, X_p, Y)$ is said to be *partial faithful (PF)* if for each $j \in \mathcal{P}$ we have

$$\rho(Y, X_j \,|\, X_S) = 0 \text{ for some } S \subseteq \mathcal{P} \setminus \{j\} \implies \rho(Y, X_j \,|\, X_{\mathcal{P}\setminus\{j\}}) = 0 \,. \tag{4.17}$$

Before we describe the implications of the PF assumption and how it is related to OIP, we cite a result which shows that the PF assumption is rather "weak" in the sense that

distributions which do not satisfy PF have Lebesgue measure zero.

Theorem 4.7 (Bühlmann et al., 2010)**.** *Given the setting of Notation 4.5, assume that the covariance matrix* $Cov(\boldsymbol{X}) = \boldsymbol{\Sigma_X}$ *is strictly positive definite and assume that*

$$\{\beta_{0,j};\ j \in S_0\} \sim f(b)db\,, \tag{4.18}$$

where $S_0 = \{k \in \mathcal{P};\ \beta_{0,k} \neq 0\}$ *is the true active set and* f *denotes a density on a subset of* $\mathbb{R}^{|S_0|}$ *of an absolutely continuous distribution with respect to the Lebesgue measure. Then the distribution of* $(X_1, \ldots, X_p, Y)$ *satisfies PF almost surely with respect to the distribution generating the non-zero regression coefficients.*

Remark 4.2 (Bühlmann et al., 2010)**.** In the setting of Notation 4.5 we have

$$\rho(Y, X_j \,|\, X_{\mathcal{P}\setminus\{j\}}) = 0 \iff \beta_{0,j} = 0\,. \tag{4.19}$$

Furthermore, if the distribution of $(X_1, \ldots, X_p, Y)$ satisfies PF, then for each $j \in \mathcal{P}$ it holds

$$\rho(Y, X_j \,|\, X_S) \neq 0 \text{ for all } S \subseteq \mathcal{P} \setminus \{j\} \iff j \in S_0 = \{k \in \mathcal{P};\ \beta_{0,k} \neq 0\}\,. \tag{4.20}$$

Remark 4.2 implies that, under PF, any truly important variable X_j (i.e. $\beta_{0,j} \neq 0$) remains "important" when conditioning on any subset $S \subseteq \mathcal{P} \setminus \{j\}$ (i.e. the corresponding partial correlation is non-zero). Therefore, if PF holds, one might expect that the criterion C, which aims at identifying S_0, does also satisfy the following analogous property (for each $j \in \mathcal{P}$):

$$j \in f_C(V) \text{ for all } V \subseteq \mathcal{P} \text{ with } j \in V \iff j \in S^* = f_C(\mathcal{P})\,. \tag{4.21}$$

Note that OIP is significantly weaker than the assumption given in (4.21) in the sense that in order to have $j = k_i \in S^*$ (with corresponding permutation $(k_1, \ldots, k_{s^*})$), we do not need to have $j \in f_C(V)$ for **all** $V \subseteq \mathcal{P}$ with $j \in V$, but only for each $V \subseteq \mathcal{P} \setminus N_{i-1}$ with $k_1, \ldots, k_i \in V$. One cannot generally expect that the PF property (4.20) on the population level implies the analogous property (4.21) or the weaker OIP in the given finite sample situation. But if OIP does not hold, then this indicates that the best model S^* according to the criterion C is not "stable" in the sense of (4.21) and that there does not even exist a "learning" path $(k_1, \ldots, k_{s^*})$, such that variable X_{k_i} is selected to be important in each "relevant experiment" in which $X_{k_1}, \ldots, X_{k_i}$ are considered (where variables with indices in N_{i-1} are not considered).

The next theorem shows that OIP is really a sufficient condition for the correct convergence of AdaSub against S^*.

Theorem 4.8. *Suppose that the ordered importance property (OIP) is satisfied. Then AdaSub converges correctly in the sense of Definition 4.2; that is for $j \in \mathcal{P}$ we have $r_j^{(t)} \to r_j^*$ (a.s.) as $t \to \infty$, where*

$$r_j^* \overset{a.s.}{=} \begin{cases} 1 & , \text{ if } j \in S^* \,, \\ 0 & , \text{ if } j \notin S^* \,. \end{cases}$$

Before we present the formal proof of Theorem 4.8, we briefly describe the main idea behind OIP and the corresponding proof of the correct convergence of AdaSub: First, variables with indices in N_0 are never selected to be in a best subset $f_C(V)$ for any set $V \subseteq \mathcal{P}$, so by Theorem 4.5 we have $r_j^{(t)} \to 0$ (a.s.) for $j \in N_0$. This means that in the following we can focus on subsets $V \subseteq \mathcal{P} \setminus N_0$. Now OIP assumes that there exists an $k_1 \in S^*$ (the "most important" variable X_{k_1}) such that it is always selected to be in the best subset $f_C(V)$ for all sets $V \subseteq \mathcal{P} \setminus N_0$ with $k_1 \in V$. By Theorem 4.5 we conclude that $r_{k_1}^{(t)} \to 1$ (a.s.). Variables with indices in N_1 are "unimportant variables" which are never selected to be in a best subset $f_C(V)$ for any set $V \subseteq \mathcal{P} \setminus N_0$ with $k_1 \in V$, so in the following we focus on subsets $V \subseteq \mathcal{P} \setminus N_1$. Now OIP assumes that there exists an $k_2 \in S^*$ (the "second most important" variable X_{k_2}) such that it is always selected to be in the best subset $f_C(V)$ for all sets $V \subseteq \mathcal{P} \setminus N_1$ with $k_1, k_2 \in V$. In other words, variable X_{k_2} is always selected to be in the best subset as long as variable X_{k_1} is also considered (while disregarding variables with indices in N_1). By Theorem 4.5 we similarly conclude that $r_{k_2}^{(t)} \to 1$ (a.s.). We continue in that way and obtain the following inclusions

$$N_0 \subseteq N_1 \subseteq \ldots \subseteq N_{s^*} = \mathcal{P} \setminus S^* \,, \tag{4.22}$$

for $N_0, N_1, \ldots, N_{s^*}$ as defined in Definition 4.4.

Proof of Theorem 4.8. Let $S^* = f_C(\mathcal{P}) = \{j_1, \ldots, j_{s^*}\}$ be the C-optimal model of size $s^* = |S^*|$. Since OIP is satisfied there exists a permutation $(k_1, \ldots, k_{s^*})$ of $(j_1, \ldots, j_{s^*})$ such that equation (4.12) holds for each $i = 1, \ldots, s^* - 1$ (with corresponding sets $N_0 \subseteq N_1 \subseteq \ldots \subseteq N_{s^*}$). Let $j \in N_0$. Then by definition we have $j \notin f_C(V)$ for all $V \subseteq \mathcal{P}$, so that

$$p_j^{(t+1)} = P(j \in f_C(V^{(t+1)}) \mid j \in V^{(t+1)}, \mathcal{F}^{(t)}) = 0$$

for all $t \in \mathbb{N}_0$. With Theorem 4.5 we conclude that $r_j^{(t)} \overset{\text{a.s.}}{\to} 0$ as $t \to \infty$ for $j \in N_0$.

Now by OIP we have $k_1 \in f_C(V)$ for all $V \subseteq \mathcal{P} \setminus N_0$ with $\{k_1\} \subseteq V$, so that for all $t \in \mathbb{N}_0$ we have

$$P(k_1 \in f_C(V^{(t+1)}) \mid k_1 \in V^{(t+1)}, N_0 \cap V^{(t+1)} = \emptyset, \mathcal{F}^{(t)}) = 1\,.$$

Note that by the independence of the Bernoulli trials in AdaSub we have

$$P(N_0 \cap V^{(t+1)} = \emptyset \mid k_1 \in V^{(t+1)}, \mathcal{F}^{(t)}) = P(N_0 \cap V^{(t+1)} = \emptyset \mid \mathcal{F}^{(t)}) = \prod_{l \in N_0} \left(1 - r_l^{(t)}\right) \overset{\text{a.s.}}{\to} 1$$

and therefore

$$P(N_0 \cap V^{(t+1)} \neq \emptyset \mid k_1 \in V^{(t+1)}, \mathcal{F}^{(t)}) = 1 - \prod_{l \in N_0} \left(1 - r_l^{(t)}\right) \overset{\text{a.s.}}{\to} 0\,.$$

Thus we conclude with the law of total probability that

$$\begin{aligned}
p_{k_1}^{(t+1)} &= P(k_1 \in f_C(V^{(t+1)}) \mid k_1 \in V^{(t+1)}, \mathcal{F}^{(t)}) \\
&= P(k_1 \in f_C(V^{(t+1)}) \mid k_1 \in V^{(t+1)}, N_0 \cap V^{(t+1)} = \emptyset, \mathcal{F}^{(t)}) \times \prod_{l \in N_0} \left(1 - r_l^{(t)}\right) \\
&\quad + P(k_1 \in f_C(V^{(t+1)}) \mid k_1 \in V^{(t+1)}, N_0 \cap V^{(t+1)} \neq \emptyset, \mathcal{F}^{(t)}) \times \left(1 - \prod_{l \in N_0} \left(1 - r_l^{(t)}\right)\right) \\
&\overset{\text{a.s.}}{\to} 1 \times 1 + 0 = 1, \quad t \to \infty\,.
\end{aligned}$$

By Theorem 4.5 we also obtain $r_{k_1}^{(t)} \overset{\text{a.s.}}{\to} 1$ as $t \to \infty$.

Now let $j \in N_1 \setminus N_0$. Then by definition we have $j \notin f_C(V)$ for all $V \subseteq \mathcal{P} \setminus N_0$ with $\{k_1\} \subseteq V$, so that

$$P(j \in f_C(V^{(t+1)}) \mid j \in V^{(t+1)}, N_0 \cap V^{(t+1)} = \emptyset, k_1 \in V^{(t+1)}, \mathcal{F}^{(t)}) = 0$$

for all $t \in \mathbb{N}_0$. Note that again by the independence of the Bernoulli trials in AdaSub we have

$$P(N_0 \cap V^{(t+1)} = \emptyset, k_1 \in V^{(t+1)} \mid j \in V^{(t+1)}, \mathcal{F}^{(t)}) = \prod_{l \in N_0} \left(1 - r_l^{(t)}\right) \times r_{k_1}^{(t)} \overset{\text{a.s.}}{\to} 1\,.$$

Thus we similarly conclude with the law of total probability that

$$\begin{aligned}
p_j^{(t+1)} &= P(j \in f_C(V^{(t+1)}) \mid j \in V^{(t+1)}, \mathcal{F}^{(t)}) \\
&= P(j \in f_C(V^{(t+1)}) \mid k_1, j \in V^{(t+1)}, N_0 \cap V^{(t+1)} = \emptyset, \mathcal{F}^{(t)}) \times \prod_{l \in N_0} \left(1 - r_l^{(t)}\right) \times r_{k_1}^{(t)} \\
&\quad + \dots \\
&\overset{\text{a.s.}}{\to} 0 \times 1 + 0 = 0, \quad t \to \infty\,.
\end{aligned}$$

By Theorem 4.5 we also obtain $r_j^{(t)} \overset{\text{a.s.}}{\to} 0$ as $t \to \infty$ for $j \in N_1 \setminus N_0$.
Now by OIP we have $k_2 \in f_C(V)$ for all $V \subseteq \mathcal{P} \setminus N_1$ with $\{k_1, k_2\} \subseteq V$, so that for all $t \in \mathbb{N}_0$ we have

$$P(k_2 \in f_C(V^{(t+1)}) \mid k_2 \in V^{(t+1)}, N_1 \cap V^{(t+1)} = \emptyset, k_1 \in V^{(t+1)}, \mathcal{F}^{(t)}) = 1\,.$$

Note that again by the independence of the Bernoulli trials in AdaSub we have

$$P(N_1 \cap V^{(t+1)} = \emptyset, k_1 \in V^{(t+1)} \mid \mathcal{F}^{(t)}) = \prod_{l \in N_1} \left(1 - r_l^{(t)}\right) \times r_{k_1}^{(t)} \overset{\text{a.s.}}{\to} 1\,.$$

Thus we similarly conclude with the law of total probability that

$$\begin{aligned} p_{k_2}^{(t+1)} &= P(k_2 \in f_C(V^{(t+1)}) \mid k_2 \in V^{(t+1)}, \mathcal{F}^{(t)}) \\ &= P(k_2 \in f_C(V^{(t+1)}) \mid k_1, k_2 \in V^{(t+1)}, N_1 \cap V^{(t+1)} = \emptyset,\, \mathcal{F}^{(t)}) \times \prod_{l \in N_1} \left(1 - r_l^{(t)}\right) \times r_{k_1}^{(t)} \\ &\quad + \dots \\ &\overset{\text{a.s.}}{\to} 1 \times 1 + 0 = 1, \quad t \to \infty\,. \end{aligned}$$

By Theorem 4.5 we also obtain $r_{k_2}^{(t)} \overset{\text{a.s.}}{\to} 1$ as $t \to \infty$.
Proceeding by induction we similarly conclude that for each $i = 2, \dots, s^* - 1$ we have $r_j^{(t)} \overset{\text{a.s.}}{\to} 0$ as $t \to \infty$ for all $j \in N_i \setminus N_{i-1}$; and for each $i = 3, \dots, s^* - 1$ we have $r_{k_i}^{(t)} \overset{\text{a.s.}}{\to} 1$ as $t \to \infty$.
Note that $k_{s^*} \in S^* = f_C(V)$ for all $V \subseteq \mathcal{P}$ with $\{k_1, \dots, k_{s^*}\} \subseteq V$ and that $N_{s^*} = \mathcal{P} \setminus S^*$ (see Remark 4.1). Therefore, by using the same arguments, we also obtain $r_{k_{s^*}}^{(t)} \overset{\text{a.s.}}{\to} 1$ as $t \to \infty$ and $r_j^{(t)} \overset{\text{a.s.}}{\to} 0$ as $t \to \infty$ for all $j \in N_{s^*} = \mathcal{P} \setminus S^*$. This completes the proof. □

Corollary 4.9. *If $|S^*| \leq 1$, then OIP is satisfied and therefore AdaSub converges correctly.*

We now return to the general situation without assuming that OIP holds.

Lemma 4.10. *For $j \in \mathcal{P}$ and $t \in \mathbb{N}_0$ we have (almost surely):*

$$p_j^{(t+1)} = \sum_{V \subseteq \mathcal{P}:\ j \in f_C(V)} \left(\prod_{k \in V \setminus \{j\}} r_k^{(t)} \right) \left(\prod_{k \in \mathcal{P} \setminus V} (1 - r_k^{(t)}) \right)\,. \tag{4.23}$$

Proof. Let $j \in \mathcal{P}$ and $t \in \mathbb{N}_0$. Then we have (almost surely)

$$p_j^{(t+1)} = P(W_j^{(t+1)} = 1 \mid Z_j^{(t+1)} = 1, \mathcal{F}^{(t)}) = P(j \in f_C(V^{(t+1)}) \mid j \in V^{(t+1)}, \mathcal{F}^{(t)})\,.$$

Since $j \in f_C(V^{(t+1)})$ implies $j \in V^{(t+1)}$, we (almost surely) conclude that

$$\begin{aligned} p_j^{(t+1)} &= \frac{P(j \in f_C(V^{(t+1)}) \mid \mathcal{F}^{(t)})}{P(j \in V^{(t+1)} \mid \mathcal{F}^{(t)})} \\ &= \frac{1}{r_j^{(t)}} \sum_{V \subseteq \mathcal{P}:\ j \in f_C(V)} P(V^{(t+1)} = V \mid \mathcal{F}^{(t)}) \\ &= \frac{1}{r_j^{(t)}} \sum_{V \subseteq \mathcal{P}:\ j \in f_C(V)} \left(\prod_{k \in V} r_k^{(t)} \right) \left(\prod_{k \in \mathcal{P} \setminus V} (1 - r_k^{(t)}) \right) \\ &= \sum_{V \subseteq \mathcal{P}:\ j \in f_C(V)} \left(\prod_{k \in V \setminus \{j\}} r_k^{(t)} \right) \left(\prod_{k \in \mathcal{P} \setminus V} (1 - r_k^{(t)}) \right) . \end{aligned}$$

□

Corollary 4.11. *If for all $j \in \mathcal{P}$ we have that $\left(p_j^{(t)}\right)_{t \in \mathbb{N}_0}$ converges almost surely against some random variable r_j^* with values in $[0, 1]$, then it holds*

$$r_j^* \overset{a.s.}{=} \sum_{V \subseteq \mathcal{P}:\ j \in f_C(V)} \left(\prod_{k \in V \setminus \{j\}} r_k^* \right) \left(\prod_{k \in \mathcal{P} \setminus V} (1 - r_k^*) \right) , \quad j \in \mathcal{P} . \tag{4.24}$$

Note that

$$r_j^* \overset{a.s.}{=} \begin{cases} 1 & , \text{ if } j \in S^*, \\ 0 & , \text{ if } j \notin S^*, \end{cases}$$

is always a solution to the given system of equations. If OIP is satisfied, then this solution is unique (almost surely).

Proof. Suppose that for all $j \in \mathcal{P}$ we have $p_j^{(t)} \overset{\text{a.s.}}{\to} r_j^*$, as $t \to \infty$. Then by Theorem 4.5 it holds that $r_j^{(t)} \overset{\text{a.s.}}{\to} r_j^*$, as $t \to \infty$. Thus, the system of equations (4.24) immediately follows from Lemma 4.10. Furthermore, it is easy to see that $r_j^* = \mathbb{1}_{S^*}(j)$, $j \in \mathcal{P}$, is always a solution to (4.24).

If OIP is satisfied with corresponding permutation $(k_1, \ldots, k_{s^*})$ of $(j_1, \ldots, j_{s^*})$ for the C-optimal model $S^* = \{j_1, \ldots, j_{s^*}\}$ with $|S^*| = s^*$, then we can show that the solution to the system of equations (4.24) is unique by using an analogous proof technique as for Theorem 4.8: For $j \in N_0$ we have $j \notin f_C(V)$ for all $V \subseteq \mathcal{P}$, which implies that the sum on the right hand side of (4.24) is empty; thus we have $r_j^* \overset{\text{a.s.}}{=} 0$. Now by OIP it holds that $k_1 \in f_C(V)$ for all $V \subseteq \mathcal{P} \setminus N_0$ with $k_1 \in V$; therefore, we have

$$r_{k_1}^* \overset{\text{a.s.}}{=} \sum_{V \subseteq \mathcal{P}:\, k_1 \in f_C(V)} \left(\prod_{l \in V \setminus \{k_1\}} r_l^* \right) \left(\prod_{l \in \mathcal{P} \setminus V} (1 - r_l^*) \right)$$

$$\overset{\text{a.s.}}{\geq} \sum_{V\subseteq\mathcal{P}\setminus N_0:\, k_1\in V} \left(\prod_{l\in V\setminus\{k_1\}} r_l^*\right)\left(\prod_{l\in(\mathcal{P}\setminus N_0)\setminus V}(1-r_l^*)\right)$$
$$\overset{\text{a.s.}}{=} \sum_{\tilde{V}\subseteq\tilde{\mathcal{P}}_1} \left(\prod_{l\in\tilde{V}} r_l^*\right)\left(\prod_{l\in\tilde{\mathcal{P}}_1\setminus\tilde{V}}(1-r_l^*)\right) \overset{\text{a.s.}}{=} 1\,,$$

with $\tilde{V} := V\setminus\{k_1\}$ and $\tilde{\mathcal{P}}_1 := \mathcal{P}\setminus(\{k_1\}\cup N_0)$. Thus, we conclude that $r_{k_1}^* \overset{\text{a.s.}}{=} 1$.

For $j \in N_1$ we have $j \notin f_C(V)$ for all $V \subseteq \mathcal{P}\setminus N_0$ with $k_1 \in V$; thus, for $j \in N_1 \setminus N_0$ it holds

$$r_j^* \overset{\text{a.s.}}{=} \sum_{V\subseteq\mathcal{P}:\, j\in f_C(V)} \left(\prod_{l\in V\setminus\{j\}} r_l^*\right)\left(\prod_{l\in\mathcal{P}\setminus V}(1-r_l^*)\right)$$
$$\overset{\text{a.s.}}{\leq} \sum_{V\subseteq\mathcal{P}:\, k_1\notin V} \left(\prod_{l\in V\setminus\{j\}} r_l^*\right)\underbrace{\left(\prod_{l\in\mathcal{P}\setminus V}(1-r_l^*)\right)}_{\overset{\text{a.s.}}{=}0} + \sum_{V\subseteq\mathcal{P}:\, V\cap N_0\neq\emptyset} \underbrace{\left(\prod_{l\in V\setminus\{j\}} r_l^*\right)}_{\overset{\text{a.s.}}{=}0}\left(\prod_{l\in\mathcal{P}\setminus V}(1-r_l^*)\right) \overset{\text{a.s.}}{=} 0\,,$$

so that $r_j^* \overset{\text{a.s.}}{=} 0$. Continuing in a similar way, for $i = 2,\ldots,s^*$, we obtain $r_j^* \overset{\text{a.s.}}{=} 0$ for all $j \in N_i$ and $r_{k_i}^* \overset{\text{a.s.}}{=} 1$. Thus, the solution to the system of equations (4.24) is unique. □

So in order to find the limit $r^* = (r_1^*,\ldots,r_p^*) \in [0,1]^p$, we need to solve the system of p equations given by (4.24). The following low-dimensional examples illustrate, how the solutions to this system of equations will look like in very simple situations (all statements below should be understood to hold almost surely).

Example 4.1. Let us consider the case $p = 2$, i.e. $\mathcal{P} = \{1,2\}$. We will examine all possible (up to relabelling) functions $f_C : \mathfrak{P}(\{1,2\}) \to \mathfrak{P}(\{1,2\})$ that are induced by a criterion $C : \mathfrak{P}(\{1,2\}) \to \mathbb{R}$.

(a) Let $f_C(V) = \emptyset$ for all $V \subseteq \{1,2\}$. Then for $j = 1,2$ the sum on the right hand side of equation (4.24) is empty, so we have $r_1^* = r_2^* = 0$.

(b) Let $f_C(\{1\}) = \{1\}$, $f_C(\{2\}) = \emptyset$ and $f_C(\{1,2\}) = \{1\}$. Then for $j = 1$, equation (4.24) implies $r_1^* = r_2^* + (1 - r_2^*) = 1$ and, for $j = 2$, it implies $r_2^* = 0$.

(c) Let $f_C(\{1\}) = \{1\}$, $f_C(\{2\}) = \{2\}$ and $f_C(\{1,2\}) = \{1\}$. Then for $j = 1$, equation (4.24) implies $r_1^* = 1$ and, for $j = 2$, it implies $r_2^* = 1 - r_1^* = 0$.

(d) Let $f_C(\{1\}) = \{1\}$, $f_C(\{2\}) = \{2\}$ and $f_C(\{1,2\}) = \{1,2\}$. Then we have $r_1^* = r_2^* = 1$.

(e) Let $f_C(\{1\}) = \{1\}$, $f_C(\{2\}) = \emptyset$ and $f_C(\{1,2\}) = \{1,2\}$. Then $r_1^* = 1$ and $r_2^* = r_1^* = 1$.

(f) Let $f_C(\{1\}) = \emptyset$, $f_C(\{2\}) = \emptyset$ and $f_C(\{1,2\}) = \{1,2\}$. Then the equations only imply $r_1^* = r_2^*$, so $r^* \in \{(t,t);\ t \in [0,1]\}$.

Note that in examples (a) to (e) OIP is satisfied and thus the unique solution to the system of equations is the desired one, i.e. $r_j^* = \mathbb{1}_{S^*}(j)$ for $j = 1,2$. The situation in (f) is, for $p = 2$, the only case, where OIP is not satisfied (since $1 \notin f_C(\{1\})$ and $2 \notin f_C(\{2\})$) and the solution to the equations is not unique. Still, the "correct" solution $r_j^* = \mathbb{1}_{S^*}(j)$, $j = 1,2$, is an element of the set of solutions.

Example 4.2. Let us now consider the case $p = 3$, i.e. $\mathcal{P} = \{1,2,3\}$. For brevity, we do not examine all possible functions f_C, but present four representative examples, for which the C-optimal model is given by the "full" model $S^* = f_C(\{1,2,3\}) = \{1,2,3\}$:

(a) Suppose that $f_C(\{1,2\}) = \{1,2\}$, $f_C(\{2,3\}) = \emptyset$, $f_C(\{1,3\}) = \{1\}$, $f_C(\{1\}) = \{1\}$, $f_C(\{2\}) = f_C(\{3\}) = \emptyset$. Then the system of equations (4.24) is given by

$$r_1^* = r_2^* r_3^* + (1 - r_2^*) r_3^* + r_2^*(1 - r_3^*) + (1 - r_2^*)(1 - r_3^*) = 1\,, \tag{4.25}$$

$$r_2^* = r_1^* r_3^* + r_1^*(1 - r_3^*) = 1 \times r_3^* + 1 \times (1 - r_3^*) = 1\,, \tag{4.26}$$

$$r_3^* = r_1^* r_2^* = 1 \times 1 = 1\,. \tag{4.27}$$

Thus $(r_1^*, r_2^*, r_3^*) = (1,1,1)$ is the unique solution, as desired and expected, since OIP is satisfied for this example with corresponding permutation $(k_1, k_2, k_3) = (1,2,3)$, i.e. we have $1 \in f_C(V)$ for all $V \subseteq \mathcal{P}$ with $\{1\} \subseteq V$ and further we have $2 \in f_C(V)$ for all $V \subseteq \mathcal{P}$ with $\{1,2\} \subseteq V$.

(b) Consider the same setting as in (a), but with $f_V(\{1,2\}) = \{1\}$ (instead of $f_C(\{1,2\}) = \{1,2\}$). Then the system of equations (4.24) is given by

$$r_1^* = r_2^* r_3^* + (1 - r_2^*) r_3^* + r_2^*(1 - r_3^*) + (1 - r_2^*)(1 - r_3^*) = 1\,, \tag{4.28}$$

$$r_2^* = r_1^* r_3^* = r_3^*\,, \tag{4.29}$$

$$r_3^* = r_1^* r_2^* = r_2^*\,. \tag{4.30}$$

Thus the set of solutions to the system of equations is given by

$$\mathcal{L} = \left\{(1, r_2^*, r_3^*) \in [0,1]^3;\ r_2^* = r_3^*\right\}\,. \tag{4.31}$$

Note that, in contrast to example (a), here OIP is not satisfied and the solution to the

system of equations is not unique. However, we have $1 \in f_C(V)$ for all $V \subseteq \mathcal{P}$ with $1 \in V$. Therefore, $r_1^* = 1$ is unique and $\left(r_1^{(t)}\right)$ converges "correctly" against $r_1^* = 1$ (compare Theorem 4.12 below).

(c) Suppose that $f_C(\{1,2\}) = \emptyset$, $f_C(\{2,3\}) = \{2,3\}$, $f_C(\{1,3\}) = \emptyset$, $f_C(\{1\}) = f_C(\{2\}) = f_C(\{3\}) = \emptyset$. Then the system of equations (4.24) is given by

$$r_1^* = r_2^* r_3^*, \quad r_2^* = r_1^* r_3^* + (1 - r_1^*) r_3^* = r_3^*, \quad r_3^* = r_1^* r_2^* + (1 - r_1^*) r_2^* = r_2^* \,. \tag{4.32}$$

It is straightforward to derive the set of solutions to the system of equations, which is given by

$$\mathcal{L} = \left\{ (r_1^*, r_2^*, r_3^*) \in [0,1]^3;\ \sqrt{r_1^*} = r_2^* = r_3^* \right\} . \tag{4.33}$$

In this example, OIP is not satisfied and no variable is "stable". Here, we possible have $r_j^* \in [0,1]$ for each $j \in \mathcal{P}$ (with the restriction that $\sqrt{r_1^*} = r_2^* = r_3^*$). Nevertheless, the "correct" solution $(r_1^*, r_2^*, r_3^*) = (1,1,1)$ is an element of $\mathcal{L}$.

(d) Suppose that $f_C(V) = \emptyset$ for all $V \subset \mathcal{P}$ with $V \neq \{1,2,3\}$. Then the system of equations (4.24) is given by

$$r_1^* = r_2^* r_3^*, \quad r_2^* = r_1^* r_3^*, \quad r_3^* = r_1^* r_2^* \,, \tag{4.34}$$

with corresponding set of solutions

$$\mathcal{L} = \{(0,0,0), (1,1,1)\} \,. \tag{4.35}$$

As in example (c), OIP is not satisfied and no variable is "stable". In contrast to example (c), the set of solutions $\mathcal{L}$ consists of only two elements, of which one is the "correct" solution $(r_1^*, r_2^*, r_3^*) = (1,1,1)$.

We have empirically investigated the limiting behaviour of AdaSub in the considered Examples 4.1 and 4.2. The details of the associated toy simulations are not presented here, but we want to remark that the results confirm the solutions r_j^* to the system of equations (4.24) as possible limits of the selection probabilities $r_j^{(t)}$, for $j \in \mathcal{P}$, as $t \to \infty$.

Now if we observe that the AdaSub algorithm does not converge correctly, i.e. if there exists $j \in \mathcal{P}$ with $r_j^{(t)} \to r_j^*$, $r_j^* \in (0,1)$ with positive probability, then we can conclude that OIP is not satisfied. In that situation we actually might not want to select $S^* = f_C(\mathcal{P})$, since then there is no "stable learning path" in the sense of OIP (as described above). In

particular, not all variables X_j with $j \in S^*$ have the strong property (compare PF), that they are always selected by the criterion C if we restrict ourselves to subsets $V \subseteq \mathcal{P}$ with $j \in V$. In that situation, we propose to consider the thresholded model $\hat{S}_\rho$ for some large threshold value (e.g. $\rho = 0.9$). Example 4.2 (b) gives already an idea of what can be expected in such a situation. This observation is generalized in the following theorem.

Theorem 4.12. *Let $S^* = \{j_1, \ldots, j_{s^*}\}$ and let $D = \{l_1, \ldots, l_d\} \subseteq S^*$ be of maximal cardinality $|D| = d$ such that there exists a permutation $(k_1, \ldots, k_d)$ of $(l_1, \ldots, l_d)$ such that for all $i = 1, \ldots, d$ we have*

$$k_i \in f_C(V) \quad \text{for all } V \subseteq \mathcal{P} \setminus N_{i-1} \text{ with } \{k_1, \ldots, k_i\} \subseteq V\,, \tag{4.36}$$

where the sets $N_0, \ldots, N_d$ are defined as in Definition 4.4. In particular we have

$$N_d = \{j \in \mathcal{P};\ j \notin f_C(V) \text{ for all } V \subseteq \mathcal{P} \setminus N_{d-1} \text{ with } \{k_1, \ldots, k_d\} \subseteq V\}\,. \tag{4.37}$$

Then for all $j \in D$ we have $r_j^{(t)} \overset{a.s.}{\to} 1$, $t \to \infty$ and for all $j \in N_d$ we have $r_j^{(t)} \overset{a.s.}{\to} 0$, $t \to \infty$.

Proof. The proof is along the lines of the proof of Theorem 4.8, using the (partial) permutation $(k_1, \ldots, k_d)$ of variables in $D \subseteq S^*$ instead of the (full) permutation $(k_1, \ldots, k_{s^*})$ of all variables in S^*. □

Theorem 4.12 implies that in a situation where OIP does not hold the thresholded model $\hat{S}_\rho$ will (for fixed $\rho \in (0, 1)$ and T large enough) contain at least those variables in S^* that are included in a maximal "learning path" in the sense of OIP. However, $\hat{S}_\rho$ might also contain additional variables which are possibly not in S^*. Nevertheless, simulation studies (compare Sections 4.2 and 5.1) show that in most of the cases when OIP is not satisfied the thresholded model $\hat{S}_\rho$ provides a sparser and often more "stable" model (with less false positives) than the "best" model $\hat{S}_\mathrm{b}$ found by AdaSub.

Remark 4.3. Note that the results of Theorem 4.8 and Theorem 4.12 indicate that the choice of the threshold $\rho \in (0, 1)$ for the thresholded model $\hat{S}_\rho$ of AdaSub should be relatively large, since the selection probabilities $r_j^{(t)}$ of all variables X_j which are included in a maximal "learning path" in the sense of OIP ($j \in D \subseteq S^*$) will finally converge to one as the number of iterations t goes to infinity. However, in practice the threshold ρ should actually not be chosen too close to one, since otherwise the selection probabilities $r_j^{(T)}$, $j \in D$, may not have exceeded that threshold after a finite number of iterations $T \in \mathbb{N}$. We observe that the choice

$\rho = 0.9$ works empirically well in combination with a sufficiently large number of iterations T (see Section 3.5, Section 4.2 and Chapter 5).

We conclude this section with a short discussion of the general case in which there are possibly different subsets of explanatory variables with the same model selection criterion value (i.e. assumption (3.2) does not hold). It turns out that the original AdaSub method (Algorithm 3.1) can be easily extended and the derived limiting properties of AdaSub do also carry over to this more general situation (with slight modifications).

Remark 4.4. Suppose that the assumption (3.2) is not satisfied. In this general setting, a subset $S \in \mathcal{M}$ is called a C-optimal model (which may not be unique), if and only if it holds $C(S) = \max_{\tilde{S} \in \mathcal{M}} C(\tilde{S})$. Furthermore, in this setting we define the map $f_C : \mathfrak{P}(\{1, \ldots, p\}) \to \mathfrak{P}(\{1, \ldots, p\})$ by

$$f_C(V) := \bigcup_{S \in M_V} S\,, \tag{4.38}$$

where, for $V \subseteq \{1, \ldots, p\} = \mathcal{P}$,

$$M_V := \left\{ S \subseteq V;\, S \in \mathcal{M} \text{ with } C(S) = \max_{\tilde{S} \subseteq V,\, \tilde{S} \in \mathcal{M}} C(\tilde{S}) \right\}. \tag{4.39}$$

Note that if the assumption that $C(S) \neq C(S')$ for all $S \neq S'$ with $S, S' \in \mathcal{M}$ is satisfied, then the C-optimal model is unique and we have $|M_V| = 1$ for all $V \subseteq \mathcal{P}$, so that the definition of the map f_C in equation (4.38) coincides with the original definition of f_C, i.e. $f_C(V) = \arg\max_{S \subseteq V,\, S \in \mathcal{M}} C(S)$. In addition, note that in this general setting $f_C(V) \in \mathcal{M}$ for $V \subseteq \mathcal{P}$ cannot be guaranteed, so that the whole power set $\mathfrak{P}(\{1, \ldots, p\})$ is chosen as the codomain of the map f_C.

By making use of the generalized definition of the map f_C, we can apply the AdaSub method (Algorithm 3.1) without any modifications also in the situation where different subsets have the same criterion value. Furthermore, if we let $S^* = f_C(\mathcal{P})$ via the definition in (4.38) (i.e. S^* is either the unique C-optimal model or the union of all C-optimal models), then the OIP assumption can be defined in the same way as in Definition 4.4 and the limiting properties of AdaSub given in Theorem 4.8 and Theorem 4.12 hold without any changes. In particular, this implies that, under OIP, AdaSub "converges" against the smallest set S^* which includes all variables that are inside a C-optimal model. In practice one may choose any of the C-optimal models in S^* for further inference; alternatively, one may use a different criterion C'

(typically inducing more sparsity) for which the C'-optimal model becomes unique.

We illustrate the general situation described in Remark 4.4 with the following simple example.

Example 4.3. Suppose that $p = 2$, i.e. $\mathcal{P} = \{1, 2\}$. Furthermore, suppose that

$$C(\emptyset) < C(\{1\}) = C(\{2\}) > C(\{1, 2\})\,, \tag{4.40}$$

i.e. the C-optimal models are given by $\{1\}$ and $\{2\}$. This implies that, for the map f_C as defined in equation (4.38), we have

$$f_C(\{1\}) = \{1\}, \quad f_C(\{2\}) = \{2\}, \quad f_C(\{1, 2\}) = \{1, 2\}\,. \tag{4.41}$$

Note that the set $\{1, 2\}$ in equation (4.40) stands for the model containing both variables X_1 and X_2, while in equation (4.41) it represents the union of the C-optimal models $\{1\}$ and $\{2\}$, as described in Remark 4.4. Here, we are in the same situation as in Example 4.1 (d) in which OIP is satisfied and AdaSub converges against the union of the C-optimal models $S^* = \{1\} \cup \{2\} = \{1, 2\}$.

4.2. Illustrative example of limiting properties

In order to illustrate the limiting properties of AdaSub in a high-dimensional setup, we consider a simulated data example with $p = 1000$ possible explanatory variables and sample size $n = 60$.

We generate one particular dataset $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{Y})$ by simulating $\boldsymbol{X} = (X_{i,j}) \in \mathbb{R}^{n \times p}$ with independent rows $\boldsymbol{X}_{i,*} \sim \mathcal{N}_p(\boldsymbol{0}, \boldsymbol{I}_p)$, where the covariance matrix is the identity matrix $\boldsymbol{I}_p \in \mathbb{R}^{p \times p}$ (i.e. "independent explanatory variables"). Furthermore, let

$$\boldsymbol{\beta}_0 = (0.4, 0.8, 1.2, 1.6, 2.0, 0, \ldots, 0)^T \in \mathbb{R}^p$$

be the true vector of regression coefficients with active set $S_0 = \{1, \ldots, 5\}$. The response $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^T$ is simulated via $Y_i \overset{\text{ind.}}{\sim} N(\boldsymbol{X}_{i,*}\boldsymbol{\beta}_0, 1)$, $i = 1, \ldots, n$.

We want to emphasize that even though the explanatory variables are generated independently in the considered example, this does not imply that all empirical correlations between the variables are close to zero. In fact, Figure 4.1 shows that the pairwise empir-

ical correlations $\text{cor}(\boldsymbol{X}_j, \boldsymbol{X}_k)$ between the observed explanatory variables $\boldsymbol{X}_j$ and $\boldsymbol{X}_k$, for $1 \leq j < k \leq p$, are approximately between -0.45 and 0.45, which is due to the small size size of $n = 60$ in comparison to the large number of variables $p = 1000$. This relatively simple example already indicates that in high-dimensional settings with small sample sizes one should typically expect a relatively diverse range of (empirical) interdependencies between the variables (which aggravates the implicit "multiple testing issue").

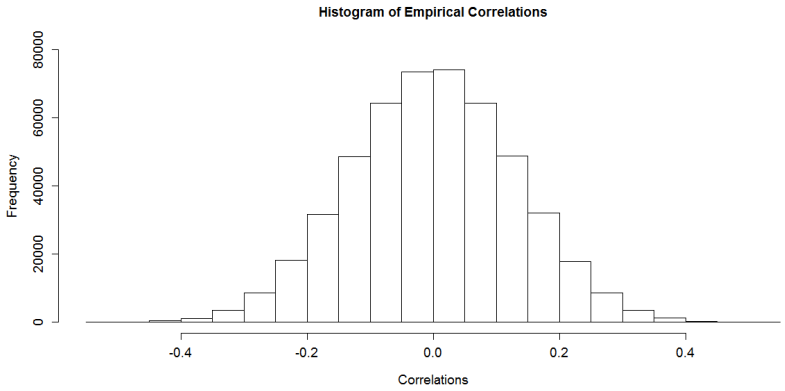
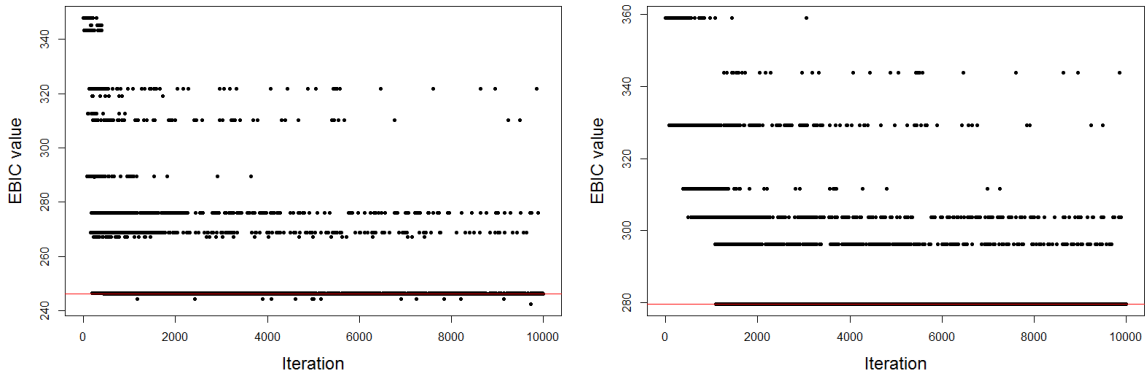


Figure 4.1.: High-dimensional simulated example: Histogram of (pairwise) empirical correlations $\text{cor}(\boldsymbol{X}_j, \boldsymbol{X}_k)$ between observed explanatory variables $\boldsymbol{X}_j$ and $\boldsymbol{X}_k$, for $1 \leq j < k \leq p$.

Here, we adopt the (negative) extended BIC (EBIC_γ, as defined in equation (2.28) of Section 2.2.2) as the criterion C and consider the tuning parameter choices $\gamma = 0.6$ and $\gamma = 1$ in EBIC_γ. For both cases, we apply AdaSub with $T = 10,000$ iterations on the same dataset simulated as above and choose $q = 10$ and $K = n$ as the tuning parameters of AdaSub (compare Section 3.5). As usual, we make use of the R-package `leaps` (Lumley and Miller, 2009) for the computation of $S^{(t)} = f_C(V^{(t)})$ for $V^{(t)} \subseteq \mathcal{P}$.

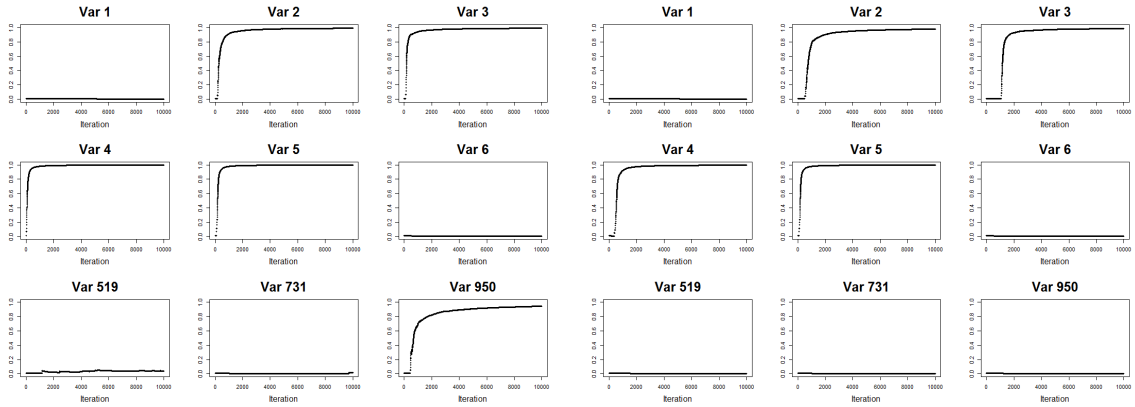
Figure 4.2 shows the evolution of the $\text{EBIC}_\gamma(S^{(t)})$-values along the iterations t for $\gamma = 0.6$ and $\gamma = 1$, while the red lines indicate the values of EBIC_γ for the thresholded model $\hat{S}_{0.9}$ with threshold $\rho = 0.9$. For $\gamma = 0.6$ it is obvious that the algorithm does not converge correctly (in the sense of Definition 4.2) against the "best" sampled model $\hat{S}_\text{b} = \arg\min\{\text{EBIC}_{0.6}(S^{(1)}), \ldots, \text{EBIC}_{0.6}(S^{(T)})\}$, indicating that OIP does not hold in this situation. The "best" model identified by AdaSub is given by $\hat{S}_\text{b} = \{2, 3, 4, 5, 519, 731, 950\}$, while the thresholded model $\hat{S}_{0.9} = \{2, 3, 4, 5, 950\}$ with threshold $\rho = 0.9$ does not include the "noise" variables X_{519} and X_{731} and is therefore closer to the true underlying model. This is an example, where the thresholded model from AdaSub reduces the number of false positives



(a) $\gamma = 0.6$ (b) $\gamma = 1$

Figure 4.2.: AdaSub for the high-dimensional simulated example. Plots of the evolution of $\text{EBIC}_\gamma(S^{(t)})$ along the iterations t for (a) $\gamma = 0.6$ and (b) $\gamma = 1$. The red lines indicate the EBIC_γ-values of the thresholded model $\hat{S}_{0.9}$.

in a situation where the criterion used is too liberal (compare Theorem 4.12). On the other hand, for $\gamma = 1$, the algorithm appears to have converged correctly; the "best" sampled model $\hat{S}_\text{b}$ and the thresholded model $\hat{S}_{0.9}$ agree: $\hat{S}_\text{b} = \hat{S}_{0.9} = \{2, 3, 4, 5\}$. This indicates, that the found model is "stable" in the sense of OIP.

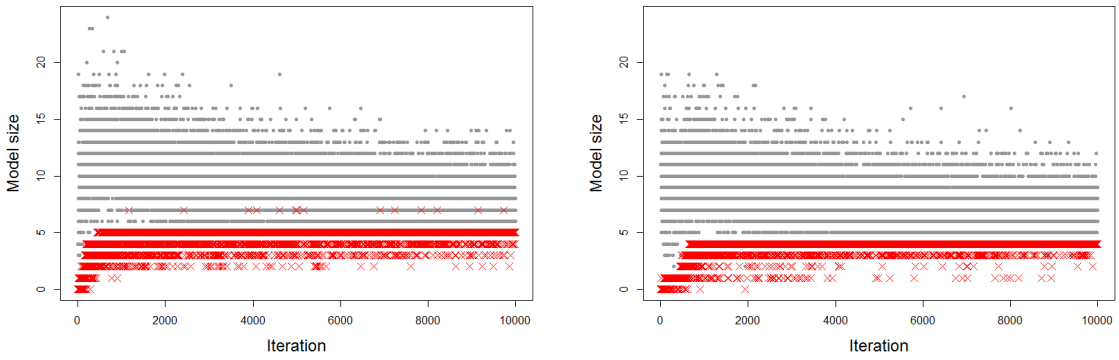


(a) $\gamma = 0.6$ (b) $\gamma = 1$

Figure 4.3.: AdaSub for the high-dimensional simulated example. Plots of the evolution of $r_j^{(t)}$ (with $j \in \{1, \ldots, 6, 519, 731, 950\}$) along the iterations t for (a) $\gamma = 0.6$ and (b) $\gamma = 1$.

Figure 4.3 shows the evolution of some of the selection probabilities $r_j^{(t)}$ along the iterations t for $\gamma = 0.6$ and $\gamma = 1$. In both cases, the selection probabilities $r_j^{(t)}$ for $j \in \{2, 3, 4, 5\}$ quickly approach the value of one, while $r_6^{(t)}$ tends to zero on the other hand. The "signal" variable X_1 is not selected in both cases (note that $\beta_1 = 0.4$ is quite small), since $r_1^{(t)}$ tends to

zero. Additionally, the evolution of the selection probabilities $r_j^{(t)}$ for $j \in \{519, 731, 950\}$ are shown. While for $\gamma = 1$ these selection probabilities all tend to 0 as desired, the behaviour is different for $\gamma = 0.6$: $r_{950}^{(t)}$ tends to one; $r_{519}^{(t)}$ and $r_{731}^{(t)}$ seem to converge to values close to, but not exactly zero. This reflects a situation, where OIP does not hold and variables X_{519} and X_{731} are not "stable" in the sense of OIP.



(a) $\gamma = 0.6$ (b) $\gamma = 1$

Figure 4.4.: AdaSub for the high-dimensional simulated example. Plots of the evolution the sizes of the sampled sets $V^{(t)}$ (grey dots) and the sizes of the selected subsets $f_C(V^{(t)}) = S^{(t)}$ (red crosses) along the iterations t for (a) $\gamma = 0.6$ and (b) $\gamma = 1$.

Figure 4.4 shows the evolution of the sizes of the sampled sets $V^{(t)}$ and the sizes of the selected subsets $S^{(t)}$ along the iterations t. Note that for $\gamma = 0.6$, AdaSub keeps on selecting subsets $S^{(t)}$ with seven variables occasionally (namely $\hat{S}_{\text{b}} = \{2, 3, 4, 5, 519, 731, 950\}$). On the other hand, for $\gamma = 1$, AdaSub apparently converges against a model with four variables (namely $\hat{S}_{0.9} = \hat{S}_{\text{b}} = \{2, 3, 4, 5\}$).

4.3. Variable selection consistency of AdaSub

In this section we investigate the variable selection consistency properties of AdaSub in a given GLM setting. In particular, we address the question under which conditions the variable selection consistency of AdaSub can be guaranteed, provided that the considered selection criterion C is (quasi-)consistent. We want to note that the results presented in this section should be considered as a starting point for the further development of consistency results for specific choices of the selection criterion C.

We first introduce some notation that deals with the challenging situation of a misspecified

model $S \subseteq \mathcal{P}$ with $S_0 \not\subseteq S$, i.e. a model S which does not include all of the truly important variables in $S_0 = \{j \in \mathcal{P};\, \beta_{0,j} \neq 0\}$.

Notation 4.7. For some observed data $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$, consider a particular GLM with likelihood $f(\boldsymbol{y} \,|\, \boldsymbol{X}_S, \mu_S, \boldsymbol{\beta}_S, \psi_S)$, induced by a subset $S \subseteq \mathcal{P}$ with $S_0 \not\subseteq S$. In this situation, the vector of (quasi-)MLEs (see e.g. White, 1982) is defined by

$$\left(\hat{\mu}_S, \hat{\boldsymbol{\beta}}_S, \hat{\psi}_S\right) \in \underset{(\mu_S, \boldsymbol{\beta}_S, \psi_S)}{\arg\max}\, f(\boldsymbol{y} \,|\, \boldsymbol{X}_S, \mu_S, \boldsymbol{\beta}_S, \psi_S) \tag{4.42}$$

and estimates its "true" population version given by

$$(\mu_S^*, \boldsymbol{\beta}_S^*, \psi_S^*) = \underset{(\mu_S, \boldsymbol{\beta}_S, \psi_S)}{\arg\max}\, E\left[f(Y \,|\, \boldsymbol{X}_S, \mu_S, \boldsymbol{\beta}_S, \psi_S)\right], \tag{4.43}$$

where the expectation is taken with respect to the (joint) distribution of the response Y and, possibly, the vector of covariates $\boldsymbol{X}_S = (X_j)_{j \in S}$, as well (in a random design setting). In the following, it is assumed that $(\mu_S^*, \boldsymbol{\beta}_S^*, \psi_S^*)$ defined by equation (4.43) is always unique, which can be guaranteed under weak regularity conditions (see e.g. Lv and Liu, 2014, Section 4). Furthermore, for $j \in S$, let $(\boldsymbol{\beta}_S^*)_j$ denote the "true" coefficient corresponding to variable X_j in the GLM induced by the subset S. Using this notation, for $V \subseteq \mathcal{P}$, let

$$S_{0,V} := \left\{j \in V;\;\; (\boldsymbol{\beta}_V^*)_j \neq 0\right\} \tag{4.44}$$

be the "true" active set when restricting to the subspace V.

Remark 4.5. Note that in general $S_{0,V} \subseteq S_0$ does not hold for $V \subseteq \mathcal{P}$ with $S_0 \not\subseteq V$. On the other hand, for all $V \subseteq \mathcal{P}$ with $S_0 \subseteq V$ we have $S_{0,V} = S_0$. Furthermore, it can be shown that, under some mild regularity conditions, the vector of (quasi-)MLEs $(\hat{\mu}_S, \hat{\boldsymbol{\beta}}_S, \hat{\psi}_S)$ is a consistent estimator of $(\mu_S^*, \boldsymbol{\beta}_S^*, \psi_S^*)$, even for misspecified GLMs induced by $S \subseteq \mathcal{P}$ with $S_0 \not\subseteq S$ (see e.g. White, 1982 and Lv and Liu, 2014 for details).

We next define the analogue to the ordered importance property (OIP) on the "population level", which will turn out to be very useful for obtaining variable selection consistency results about AdaSub.

Definition 4.8. Let $S_0 = \{j \in \mathcal{P};\, \beta_{0,j} \neq 0\} = \{j_1, \ldots, j_{s_0}\}$ be the true active set of size $|S_0| = s_0$. Then the *population ordered importance property (POIP)* is satisfied if there exists

a permutation $(k_1, \ldots, k_{s_0})$ of $(j_1, \ldots, j_{s_0})$ such that for each $i = 1, \ldots, s_0 - 1$ it holds

$$(\boldsymbol{\beta}^*_V)_{k_i} \neq 0 \text{ for all } V \subseteq \mathcal{P} \setminus N_{0,i-1} \text{ with } \{k_1, \ldots, k_i\} \subseteq V\,, \tag{4.45}$$

where

$$N_{0,0} := \{j \in \mathcal{P};\ (\boldsymbol{\beta}^*_V)_j = 0 \text{ for all } V \subseteq \mathcal{P} \text{ with } j \in V\} \tag{4.46}$$

and

$$N_{0,i} := \{j \in \mathcal{P};\ (\boldsymbol{\beta}^*_V)_j = 0 \text{ for all } V \subseteq \mathcal{P} \setminus N_{0,i-1} \text{ with } \{k_1, \ldots, k_i, j\} \subseteq V\}\,. \tag{4.47}$$

Remark 4.6. Although it is obviously not possible to verify the POIP condition in practice (since for example S_0 and $\boldsymbol{\beta}^*_V$ for $V \subseteq \mathcal{P}$ are unknown), POIP is arguably a very weak condition. In particular, note that, for the case of normal linear models with random designs, POIP is weaker than the partial faithfulness condition proposed by Bühlmann et al. (2010) (see Definition 4.6 of Section 4.1), which basically assumes that $(\boldsymbol{\beta}^*_V)_j \neq 0$ for **all** $j \in S_0$ and **all** subspaces $V \subseteq \mathcal{P}$ with $j \in V$. Very recently, Li and Liu (2017) make use of a so-called *stepwise detectable condition (SDC)* in order to show the consistency of their proposed forward-backward method "SODA" for variable and interaction selection in logistic regression models. Since SDC is of a similar nature as POIP, a thorough investigation of the relation between SDC and POIP will be an interesting topic for further research.

In order to show the variable selection consistency of AdaSub, we need to assume that the employed selection criterion C is variable selection consistent and, furthermore, that the criterion is also "quasi-consistent" in the case of misspecified models, which is formalized in the following definition.

Definition 4.9. Consider an asymptotic setting with certain conditions on the growth rates of the number of variables p_n and the number of relevant variables $|S_0^{(n)}| = s_0^{(n)}$ with respect to the sample size n. For $n \in \mathbb{N}$ and observed data $\mathcal{D}_n$, let $C_{\mathcal{D}_n} \equiv C_n : \mathcal{M}^{(n)} \to \mathbb{R}$ be a selection criterion, where $\mathcal{P}^{(n)} = \{1, \ldots, p_n\}$ denotes the index set of explanatory variables and $\mathcal{M}^{(n)}$ the corresponding model space.
Then the criterion C_n is called *quasi-variable selection consistent* (compare e.g. Nishii, 1988) for the given asymptotic setting if for each sequence $(V^{(n)})_{n\in\mathbb{N}}$ with $V^{(n)} \subseteq \mathcal{P}^{(n)}$ we have

$$P\left(f_{C_n}\left(V^{(n)}\right) = S^{(n)}_{0,V^{(n)}}\right) \to 1\,, \quad n \to \infty\,, \tag{4.48}$$

where $S^{(n)}_{0,V^{(n)}}$ denotes the "true" active set when restricting to the subspace $V^{(n)}$ (see Notation 4.7) and f_{C_n} denotes the usual projection operator for criterion C_n as defined in Notation 3.1 of Section 3.2.

Theorem 4.13. *Suppose that C_n is a quasi-consistent variable selection criterion in the sense of Definition 4.9, under some conditions $\mathfrak{C}$, for a given asymptotic setting with $p_n = \mathcal{O}(h(n))$ for some function h. Assume that the true active set*

$$S_0 = S_0^{(n)} = \{j \in \mathcal{P}^{(n)};\ \beta^{(n)}_{0,j} \neq 0\} = \{j_1, \ldots, j_{s_0}\} \tag{4.49}$$

is fixed with size $|S_0| = s_0$. Furthermore, for $n \in \mathbb{N}$ define

$$M^{(n)} := \left\{ j \in \mathcal{P}^{(n)} \setminus S_0;\ \left(\boldsymbol{\beta}^{*(n)}_V\right)_j \neq 0 \text{ for some } V \subseteq \mathcal{P}^{(n)} \text{ with } j \in V \right\} \tag{4.50}$$

and assume that $|M^{(n)}| \leq \Gamma_1$ for some finite constant $\Gamma_1 \in \mathbb{N}$ independent of n.
If POIP is satisfied for almost all $n \in \mathbb{N}$ and the conditions $\mathfrak{C}$ are also satisfied, then, under the same asymptotic setting with $p_n = \mathcal{O}(h(n))$, it holds

$$\lim_{n\to\infty} P\Big(\text{The criterion } C_n \text{ fulfils OIP} \Big) = 1 \tag{4.51}$$

and

$$\lim_{n\to\infty} P\Big(\text{AdaSub with criterion } C_n \text{ converges correctly against } S_0 \Big) = 1\,. \tag{4.52}$$

Proof. For ease of presentation and without loss of generality, suppose that POIP is satisfied for all $n \in \mathbb{N}$. Then, by the definition of POIP, for all $n \in \mathbb{N}$ there exists a permutation $(k_1, \ldots, k_{s_0})$ of $(j_1, \ldots, j_{s_0})$ such that for each $i = 1, \ldots, s_0 - 1$ it holds

$$(\boldsymbol{\beta}^{*(n)}_V)_{k_i} \neq 0 \text{ for all } V \subseteq \mathcal{P}^{(n)} \setminus N^{(n)}_{0,i-1} \text{ with } \{k_1, \ldots, k_i\} \subseteq V\,,$$

where $N^{(n)}_{0,i}$, $i = 0, \ldots, s_0 - 2$, are defined as in Definition 4.8. Note that

$$\mathcal{P}^{(n)} = S_0 \cup M^{(n)} \cup N^{(n)}_{0,0}\,.$$

For $n \in \mathbb{N}$ and each subset $W \subseteq S_0 \cup M^{(n)}$ we define events

$$E^{(n)}_W : \quad f_{C_n}\left(W \cup N^{(n)}_{0,0}\right) = S^{(n)}_{0,W}\,,$$

where $S^{(n)}_{0,W}$ is defined as in Notation 4.7 and f_{C_n} denotes the usual projection map as defined

in Notation 3.1 of Section 3.2. Since C_n is a quasi-consistent criterion for the considered asymptotic setting with $p_n = \mathcal{O}(h(n))$ under the assumed conditions $\mathfrak{C}$, for each sequence $(W^{(n)})_{n\in\mathbb{N}}$ with $W^{(n)} \subseteq S_0 \cup M^{(n)}$ we have $P\left(E^{(n)}_{W^{(n)}}\right) \overset{n\to\infty}{\to} 1$ (note that $|W^{(n)}| \leq |S_0 \cup M^{(n)}| \leq p_n$).

Define

$$E^{(n)} = \bigcap_{W\subseteq S_0\cup M^{(n)}} E^{(n)}_W \,.$$

Then it follows that

$$P\left(E^{(n)}\right) = 1 - P\left(\bigcup_{W\subseteq S_0\cup M^{(n)}} \left(E^{(n)}_W\right)^C\right) \geq 1 - \sum_{W\subseteq S_0\cup M^{(n)}} \underbrace{\left(1 - P\left(E^{(n)}_W\right)\right)}_{\to 0} \overset{n\to\infty}{\to} 1\,, \quad (4.53)$$

since we have $|M^{(n)}| \leq \Gamma_1$ by assumption and thus the sum in (4.53) over $W \subseteq S_0 \cup M^{(n)}$ consists of at most $2^{s_0+\Gamma_1}$ summands (independent of $n \in \mathbb{N}$).

Hence we conclude that $P\left(E^{(n)}\right) \overset{n\to\infty}{\to} 1$.

For $n \in \mathbb{N}$ we define the event

$$A^{(n)}_0 : \quad \text{For all } j \in N^{(n)}_{0,0} \text{ and } V \subseteq \mathcal{P}^{(n)} \text{ with } j \in V \text{ it holds } j \notin f_{C_n}(V)\,.$$

We show that $E^{(n)} \subseteq A^{(n)}_0$: Let $j \in N^{(n)}_{0,0}$ and $V \subseteq \mathcal{P}^{(n)}$ with $j \in V$. Then we have

$$V = \underbrace{\left(V \cap \left(S_0 \cup M^{(n)}\right)\right)}_{=:W\subseteq S_0\cup M^{(n)}} \cup \underbrace{\left(V \cap N^{(n)}_{0,0}\right)}_{=:W'\subseteq N^{(n)}_{0,0}}\,.$$

Now under the event $E^{(n)}$ it holds $f_{C_n}\left(W \cup N^{(n)}_{0,0}\right) = S^{(n)}_{0,W}$ and by a property of the map f_{C_n} (see Lemma 4.6 (d) of Section 4.1) we also have $f_{C_n}(V) = f_{C_n}(W \cup W') = S^{(n)}_{0,W}$ for $W' \subseteq N^{(n)}_{0,0}$. With the definition of the set $N^{(n)}_{0,0}$ we conclude that under $E^{(n)}$ it holds $j \notin S^{(n)}_{0,W} = f_{C_n}(V)$. Thus, we have shown that $E^{(n)} \subseteq A^{(n)}_0$.

Similarly, for $n \in \mathbb{N}$ and $i = 1, \ldots, s_0 - 1$ we define the events

$$A^{(n)}_i : \text{ For all } j \in N^{(n)}_{0,i} \text{ and } V \subseteq \mathcal{P}^{(n)} \setminus N^{(n)}_{0,i-1} \text{ with } \{k_1, \ldots, k_i, j\} \subseteq V \text{ it holds } j \notin f_{C_n}(V)$$

and

$$B^{(n)}_i : \; k_i \in f_{C_n}(V) \text{ for all } V \subseteq \mathcal{P}^{(n)} \setminus N^{(n)}_{0,i-1} \text{ with } \{k_1, \ldots, k_i\} \subseteq V\,.$$

Using similar arguments as above, under the event $E^{(n)}$ it holds $f_{C_n}(V) = S^{(n)}_{0,V}$ for all $V \subseteq S_0 \cup M^{(n)}$.

Thus, by the definition of $S_{0,V}^{(n)}$ and POIP we have

$$E^{(n)} \subseteq \bigcap_{i=0}^{s_0-2} A_i^{(n)} \cap \bigcap_{i=1}^{s_0-1} B_i^{(n)} .$$

Since $P\left(E^{(n)}\right) \overset{n\to\infty}{\to} 1$, we conclude that

$$P\Big(\text{ The criterion } C_n \text{ fulfils OIP }\Big) = P\left(\bigcap_{i=0}^{s_0-2} A_i^{(n)} \cap \bigcap_{i=1}^{s_0-1} B_i^{(n)}\right) \overset{n\to\infty}{\to} 1 .$$

Note that under the event $E^{(n)} \subseteq E^{(n)}_{S_0 \cup M^{(n)}}$ we have

$$f_{C_n}\left(\mathcal{P}^{(n)}\right) = f_{C_n}\left(S_0 \cup M^{(n)} \cup N_{0,0}^{(n)}\right) = S_{0,S_0\cup M^{(n)}} = S_0 .$$

Thus, by Theorem 4.8 of Section 4.1 we conclude that

$$P\Big(\text{ AdaSub with criterion } C_n \text{ converges correctly against } f_{C_n}\left(\mathcal{P}^{(n)}\right) = S_0\Big) \overset{n\to\infty}{\to} 1 .$$

□

Corollary 4.14. *Suppose that C_n is a quasi-consistent variable selection criterion, under some conditions $\mathfrak{C}$, for the classical asymptotic setting where $n \to \infty$ but the number of explanatory variables p and the true active set S_0 are fixed. If POIP is satisfied for almost all $n \in \mathbb{N}$ and the conditions $\mathfrak{C}$ are also satisfied, then, under the classical asymptotic setting, it holds*

$$\lim_{n\to\infty} P\Big(\textit{ AdaSub with criterion } C_n \textit{ converges correctly against } S_0\Big) = 1 . \tag{4.54}$$

Proof. The result follows immediately from Theorem 4.13 by noting that $M^{(n)} \subseteq \mathcal{P}^{(n)} \setminus S_0$ and thus

$$|M^{(n)}| \leq |\mathcal{P}^{(n)} \setminus S_0| = p - s_0 =: \Gamma_1 ,$$

where $\Gamma_1 \in \mathbb{N}$ is independent of n. □

Remark 4.7. For the classical asymptotic setting where the number of explanatory variables p is fixed, it has been shown by Nishii (1988) that the BIC is a quasi-consistent variable selection criterion under mild regularity conditions which, for each GLM induced by $S \in \mathcal{M}$, ensure identifiability and strong consistency of the vector of quasi-MLEs $\hat{\boldsymbol{\theta}}_S = (\hat{\mu}_S, \hat{\boldsymbol{\beta}}_S, \hat{\psi}_S)$ for estimating $\boldsymbol{\theta}_S^* = (\mu_S^*, \boldsymbol{\beta}_S^*, \psi_S^*)$, i.e. $\hat{\boldsymbol{\theta}}_S \overset{\text{a.s.}}{\to} \boldsymbol{\theta}_S^*$ for $n \to \infty$ (compare also White, 1982

and Chen et al., 1999). Thus, Corollary 4.14 implies that, for the classical asymptotic setting, AdaSub in combination with BIC as the employed selection criterion yields a variable selection consistent procedure provided that only the POIP condition and the regularity conditions of Nishii (1988) are satisfied.

Remark 4.8. In order to apply Theorem 4.13 in a high-dimensional asymptotic setting where the number of explanatory variables p_n goes to infinity with the sample size n, we need to assume that the true active set $S_0^{(n)} = S_0$ is fixed (or, at least, that the size of the true active set is bounded by some constant $\Gamma_0 \in \mathbb{N}$, i.e. $|S_0^{(n)}| \leq \Gamma_0$) and, furthermore, that the size of the set

$$M^{(n)} = \left\{ j \in \mathcal{P}^{(n)} \setminus S_0;\, \left(\boldsymbol{\beta}_V^{*(n)}\right)_j \neq 0 \text{ for some } V \subseteq \mathcal{P}^{(n)} \text{ with } j \in V \right\}$$

is bounded by some constant $\Gamma_1 \in \mathbb{N}$. In other words, there should only be a bounded number of "noise" variables (i.e. $j \in \mathcal{P}^{(n)} \setminus S_0$) , which become "relevant" when certain "signal" variables are not considered to be in the model. This is arguably a strong condition in a high-dimensional setting, where usually many "noise" variables are correlated with "signal" variables. However, this condition can probably be relaxed when one makes use of specific consistency rates for certain variable selection criteria in the proof of Theorem 4.13. As we have already remarked at the beginning of this section, Theorem 4.13 should only be considered as a starting point for further consistency results including settings where possibly $|M^{(n)}| \to \infty$ as well as $|S_0^{(n)}| \to \infty$ with certain rates depending on n and p_n. In addition, we want to note that — although we do expect similar results to hold as presented in Nishii (1988) for the BIC — we are not aware of detailed expositions of "quasi-consistency" results for selection criteria like the EBIC in high-dimensional misspecified asymptotic settings.

Remark 4.9. We want to emphasize that the obtained consistency result for AdaSub in Theorem 4.13 is of the form

$$\lim_{n\to\infty} P\Big(\text{AdaSub with criterion } C_n \text{ converges correctly against } S_0 \Big) = 1\,,$$

which implies that

$$\lim_{n\to\infty, T\to\infty} P\Big(\hat{S}_\rho^{(n,T)} = S_0 \Big) = 1\,,$$

where $\hat{S}_\rho^{(n,T)}$ denotes the thresholded model of AdaSub for some threshold $\rho \in (0,1)$ after iteration $T \in \mathbb{N}$ when using criterion C_n corresponding to data $\mathcal{D}_n$. So it is apparent that the

consistency of AdaSub is with respect to the divergence of both the sample size n (of the data) and the number of iterations T (of the algorithm). However, in practice we usually observe data with only finite sample sizes n and run the AdaSub algorithm for only a finite number of iterations T. It would be particularly desirable to obtain theoretical results about the "speed of convergence" of AdaSub and further practical tools for "checking the convergence" of AdaSub. We think that these are important and challenging problems for future research.

Remark 4.10. On a final note, we want to make clear that in this section we have considered the (asymptotic) identification of the "true" active set $S_0 = \{j \in \mathcal{P};\ \beta_j \neq 0\}$ in a GLM (assuming its existence), i.e. we have addressed the question whether there is a nonzero association between the response variable Y and an explanatory variable X_j, when controlling for the other explanatory variables X_k, $k \in \mathcal{P} \setminus \{j\}$ (meaning that the other variables are kept **fixed**). Recall the general wisdom that "association does not imply causation". In fact, we have not aimed at addressing the fundamentally more challenging question whether there is a causal relationship between the respective explanatory variables and the response. Note that if we would for example make an intervention modifying one particular explanatory variable X_j, in many applications some of the other explanatory variables are likely to change as well, so that the regression coefficient β_j in the corresponding GLM is not necessarily the "right" quantity for measuring the causal effect of the intervention on the response. We refer to Pearl (2009) and Bühlmann (2013a) for nice reviews of casual inference, addressing even high-dimensional settings. Furthermore, we want to stress that identifying the "true" underlying model and finding an "optimal" model with respect to prediction are also fundamentally different targets (see the discussion of the dichotomy concerning the AIC and the BIC in Section 2.2.1, as well as the results from simulation studies in Sections 5.1 and 6.3).

5. Performance of AdaSub on simulated and real data examples

In this chapter we investigate the performance of the proposed AdaSub method (Algorithm 3.1) when applied on various simulated and real data examples. Here, we focus on linear regression models, so that the application of the original AdaSub algorithm is computationally fast (compare Section 3.6). We refer to Chapter 6 for a discussion of different modifications of AdaSub and their application to different GLMs like logistic regression models.

In Section 5.1 we present results from simulation studies in low- and high-dimensional settings and compare AdaSub with other prominent variable selection methods that have been discussed in Chapter 2, including Best Subset Selection, Forward Stepwise Selection, the Lasso, the Adaptive Lasso, the SCAD and Stability Selection. In Section 5.2 we illustrate the efficiency of AdaSub via two high-dimensional real data examples from the field of genomics, where the number of possible explanatory variables is in the order of ten thousands.

Note that this chapter is an extended version of the material submitted for publication in Staerk et al. (2018).

5.1. Simulation study

We have investigated the performance of AdaSub in extensive simulation studies and in this section we want to present some representative results. The discussion is divided into two parts: In the first part (Section 5.1.1) we examine relatively low-dimensional simulated examples where it is feasible to compute the best model according to an ℓ_0-type selection criterion C, so that the C-optimal model can be compared to the output of AdaSub. In the second part (Section 5.1.2) we apply AdaSub on high-dimensional simulated examples and compare its performance with different well-known methods.

We will make use of the following simulation setup.

Simulation Setup 5.1. For a given sample size $n \in \mathbb{N}$ and a number of explanatory variables $p \in \mathbb{N}$ we simulate the design matrix $\boldsymbol{X} = (X_{i,j}) \in \mathbb{R}^{n \times p}$ with i-th row $\boldsymbol{X}_{i,*} \sim \mathcal{N}_p(\boldsymbol{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} \in \mathbb{R}^{p \times p}$ is a positive definite correlation matrix with entries $\Sigma_{k,k} = 1$ for $k = 1, \ldots, p$. We consider different correlation structures between the explanatory variables induced by the matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{p \times p}$:

(a) **Equal-Correlation Structure:** For some fixed $c \in [0, 1)$ let $\Sigma_{k,l} = c$ for all $k \neq l$.

(b) **Toeplitz-Correlation Structure:** For some fixed $c \in (-1, 1)$ let $\Sigma_{k,l} = c^{|k-l|}$ for all $k \neq l$.

(c) **Block-Correlation Structure:** For some fixed $c \in (0, 1)$ and a fixed number of blocks $b \in \mathbb{N}$ let $\Sigma_{k,l} = c$ for all $k \neq l$ with $(k - l) \bmod b = 0$, and let $\Sigma_{k,l} = 0$ otherwise.

For each dataset, we select $s_0 \in \{0, \ldots, 10\}$ and $S_0 \subset \mathcal{P} = \{1, \ldots, p\}$ of size $|S_0| = s_0$ randomly; then for $j \in S_0$ we independently simulate $\beta_{0,j} \sim \mathcal{U}(-2, 2)$ from the uniform distribution on $[-2, 2]$, while we set $\beta_{0,j} = 0$ for $j \notin S_0$. The response vector $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^T$ is simulated according to the normal linear model via $Y_i \overset{\text{ind.}}{\sim} N(\boldsymbol{X}_{i,*}\boldsymbol{\beta}_0, \sigma^2)$, $i = 1, \ldots, n$ with error variance $\sigma^2 = 1$, where $\boldsymbol{\beta}_0 = (\beta_{0,1}, \ldots, \beta_{0,p})^T \in \mathbb{R}^p$ denotes the true underlying vector of regression coefficients.

In the following, we apply AdaSub in combination with the (negative) EBIC_γ (see equation (2.28) of Section 2.2.2) as the selection criterion C for different regularization constants $\gamma \in [0, 1]$ (recall that $\gamma = 0$ corresponds to the usual BIC). In AdaSub we always use the "leaps and bounds" algorithm implemented in the R-package `leaps` (Lumley and Miller, 2009) to compute at iteration t the best model $S^{(t)} = f_C(V^{(t)})$ according to the criterion C contained in the sampled subspace $V^{(t)} \subseteq \mathcal{P}$.

In order to compare the performance of AdaSub with other variable selection methods we consider the following measures.

Notation 5.1. For a given observed dataset $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$ (referred to as the training set) with $\boldsymbol{X} \in \mathbb{R}^{n \times p}$, let $\hat{S}(\mathcal{D}) \equiv \hat{S} \subseteq \mathcal{P} = \{1, \ldots, p\}$ denote the final model selected by some variable selection method and let $S_0 = \{j \in \mathcal{P};\, \beta_{0,j} \neq 0\} \subseteq \mathcal{P}$ denote the true active set (corresponding to the true underlying model). In this situation, the number of *false positives*

is given by $|\hat{S} \cap (\mathcal{P} \setminus S_0)|$ and the number of *false negatives* is given by $|(\mathcal{P} \setminus \hat{S}) \cap S_0|$. Furthermore, if $\hat{\boldsymbol{\beta}} \equiv \hat{\boldsymbol{\beta}}(\hat{S}) = (\hat{\boldsymbol{\beta}}_1(\hat{S}), \ldots, \hat{\boldsymbol{\beta}}_p(\hat{S}))^T \in \mathbb{R}^p$ denotes the (full) estimated vector of regression coefficients under the estimated model $\hat{S}$ (with $\hat{\boldsymbol{\beta}}_j(\hat{S}) = 0$ for $j \in \mathcal{P} \setminus \hat{S}$), while $\boldsymbol{\beta}_0 \in \mathbb{R}^p$ denotes the true vector of coefficients, then the *Squared Error (SqE)* for estimating $\boldsymbol{\beta}_0$ is given by

$$\mathrm{SqE}(\hat{\boldsymbol{\beta}}) = \left\| \hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0 \right\|_2^2 . \tag{5.1}$$

Let $\mathcal{D}' = (\boldsymbol{X}', \boldsymbol{y}')$ denote a given test set of sample size n' (assumed to be independently generated from the same underlying distribution as the training set $\mathcal{D}$). Furthermore, let $\hat{\boldsymbol{y}}' = \boldsymbol{X}'\hat{\boldsymbol{\beta}} \in \mathbb{R}^{n'}$ denote the estimated response (using the previously estimated $\hat{\boldsymbol{\beta}}$ based on the training set $\mathcal{D}$). Then the (out-of-sample) *Mean Squared Prediction Error (MSPE)* for the test set $\mathcal{D}'$ is given by

$$\mathrm{MSPE}(\mathcal{D}') = \frac{1}{n'} \left\| \hat{\boldsymbol{y}}' - \boldsymbol{y}' \right\|_2^2 . \tag{5.2}$$

In the following simulation studies, we will often evaluate the performance of a variable selection procedure many times on different simulated data examples from similar underlying distributions. In order to obtain measures of the average performance for such situations, we compute the mean of the number of false negatives (*mean false negatives*) and the mean of the number of false positives (*mean false positives*), as well as the mean of the Squared Errors (abbreviated by *MSE*) and the mean of the Mean Squared Prediction Errors (*averaged MSPE*, or for brevity simply *MSPE*).

5.1.1. Low-dimensional setting

It is illuminating to analyse the performance of AdaSub in a relatively low-dimensional situation where we actually can compute the best model according to the criterion used. We are thus able to answer the question whether AdaSub really recovers the C-optimal model. In order to compute the C-optimal model in reasonable computational time using the "leaps and bounds" algorithm, we set the number of explanatory variables to be $p = 30$. For a given correlation structure, the sample size n is increased from 40 to 200 in steps of size 20 and for each value of n we simulate 100 different datasets according to the Simulation Setup 5.1 described above. We consider the (negative) BIC (i.e. EBIC_γ with $\gamma = 0$) as the selection criterion C in AdaSub. This is a sensible choice for the given situation, since it is well-known that the BIC is variable selection consistent in the low-dimensional asymptotic

setting in which the sample size n tends to infinity while the number of explanatory variables p is fixed (compare Section 2.2.1). In AdaSub we set $q = 5$ as the initial expected search size and $K = n$ as the learning rate (compare the discussion in Sections 3.4 and 3.5). For each simulated example we run AdaSub for $T = 2000$ iterations (which empirically yields a sufficient convergence behaviour of AdaSub in the considered set-up).

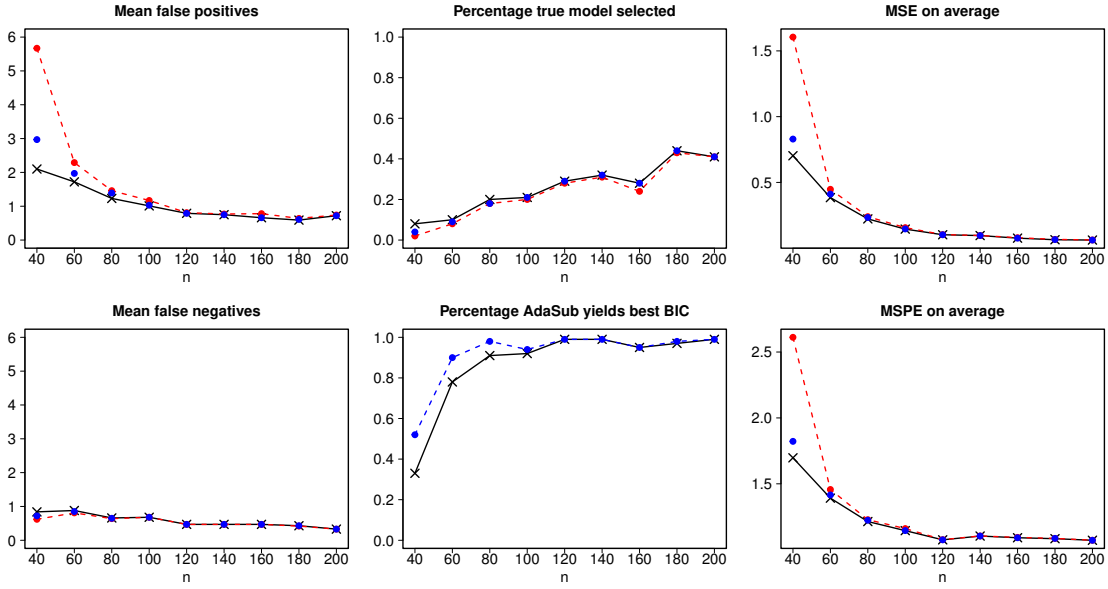


Figure 5.1.: Low-dimensional example ($p = 30$) with equal-correlation structure ($c = 0$): Comparison of $\hat{S}_{0.9}$ (black) and $\hat{S}_{\mathrm{b}}$ (blue) from AdaSub with BIC-optimal model S^* (red) in terms of mean false positives/ false negatives, percentage of selecting the true model S_0, percentage of agreement between AdaSub models and S^*, MSE and (averaged) MSPE on independent test set with sample size 100.

Figure 5.1 summarizes the results of the low-dimensional simulation study in the case of an equal-correlation structure with $c = 0$ (i.e. independent explanatory variables). For moderately large sample sizes ($n \geq 100$), the models selected by AdaSub agree with the BIC-optimal models in most of the cases and the relative frequency of agreement seems to "converge" to one as the sample size increases. This empirical observation confirms the theoretical consistency results of AdaSub when using the BIC as a selection criterion in the classical asymptotic setting (see Theorem 4.13 and Remark 4.7 of Section 4.3).

However, for small sample sizes ($n \leq 60$), the BIC-optimal model S^* tends to include many false positives and "overfits" the data in many cases. On the other hand, $\hat{S}_{0.9}$ and $\hat{S}_{\mathrm{b}}$ from AdaSub yield sparser models and often reduce the number of falsely selected variables in a situation where the BIC is too liberal. This comes at the price of a slightly increased

number of false negatives (for small n), but the overall effect of selecting a sparser model with AdaSub is beneficial yielding higher relative frequencies of selecting the true model S_0, smaller Mean Squared Errors (MSE) and smaller Mean Squared Prediction Errors (MSPE). Although the "best" model $\hat{S}_{\mathrm{b}}$ from AdaSub identifies the BIC-optimal model more often than the thresholded model $\hat{S}_{0.9}$ from AdaSub, the choice of $\hat{S}_{0.9}$ is beneficial for the given situation.

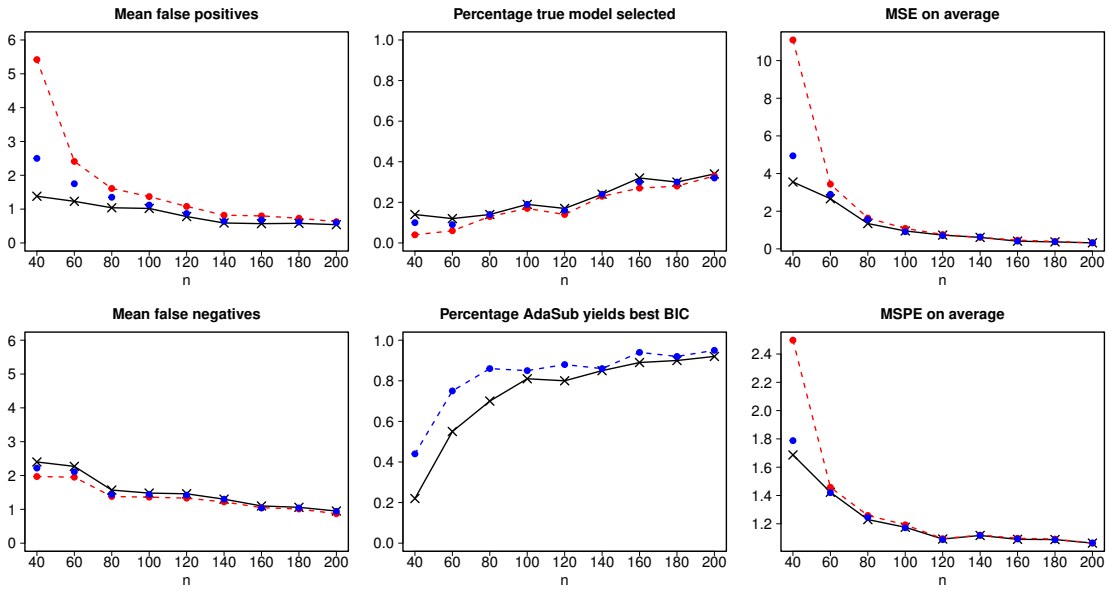


Figure 5.2.: Low-dimensional example ($p = 30$) with Toeplitz-correlation structure ($c = 0.9$): Comparison of $\hat{S}_{0.9}$ (black) and $\hat{S}_{\mathrm{b}}$ (blue) from AdaSub with BIC-optimal model S^* (red) in terms of mean false positives/ false negatives, percentage of selecting the true model S_0, percentage of agreement between AdaSub models and S^*, MSE and (averaged) MSPE on independent test set with sample size 100.

Figure 5.2 depicts the results in a situation with large correlations between the explanatory variables (Toeplitz-correlation structure with $c = 0.9$). The observations are similar to the case of underlying independence above (see Figure 5.1). The relative frequency of agreement between the models selected by AdaSub and the BIC-optimal models increases towards one when the sample size n increases (again, confirming the theoretical consistency results derived in Section 4.3), but the "convergence" is slower than in the independent case. This shows that the models from AdaSub may have different (and in the given setting preferable) statistical properties in comparison to the BIC-optimal model even if the sample size is moderately large.

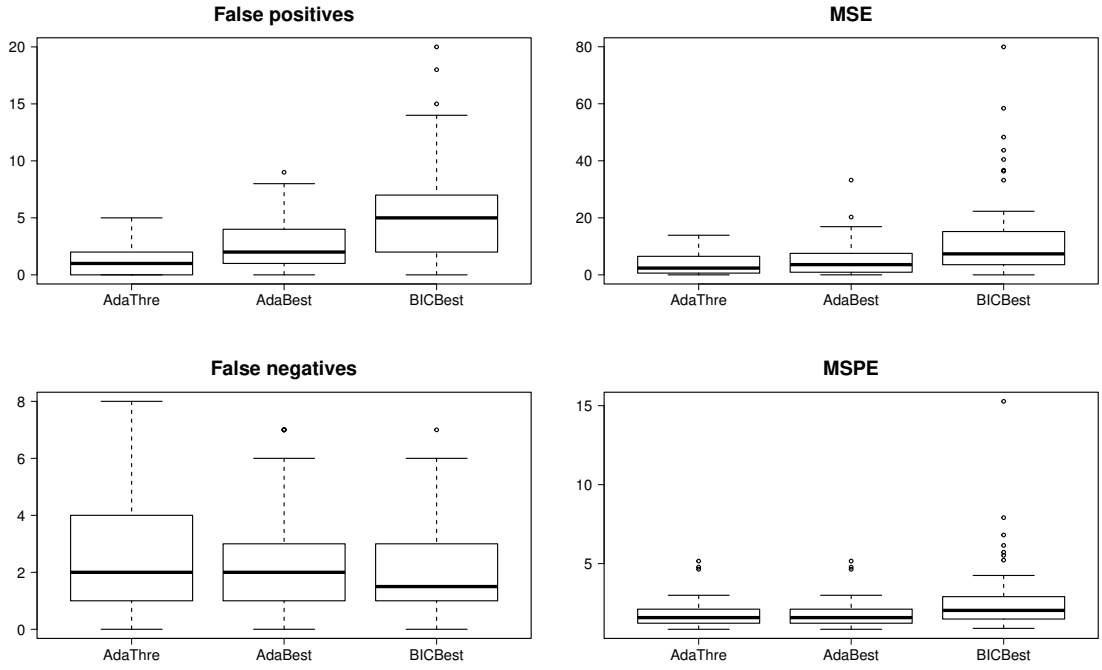


Figure 5.3.: Low-dimensional example ($p = 30$) with Toeplitz-correlation structure ($c = 0.9$): Boxplots for the case $n = 40$, for $\hat{S}_{0.9}$ (AdaThre) and $\hat{S}_{\mathrm{b}}$ (AdaBest) from AdaSub, as well as for the BIC-optimal model S^* (BICBest).

In order to assess the variability of the results, Figure 5.3 depicts boxplots of the performance measures in the specific case of $n = 40$ (for the Toeplitz-correlation structure with $c = 0.9$ and 100 replicates). It is apparent that the BIC-optimal model often "overfits" the data of small sample size, yielding very large numbers of false positives and inaccurate predictions. On the other hand, even though the BIC is employed as the selection criterion in AdaSub, the thresholded model from AdaSub yields relatively small numbers of false positives as well as reasonable predictions in most cases.

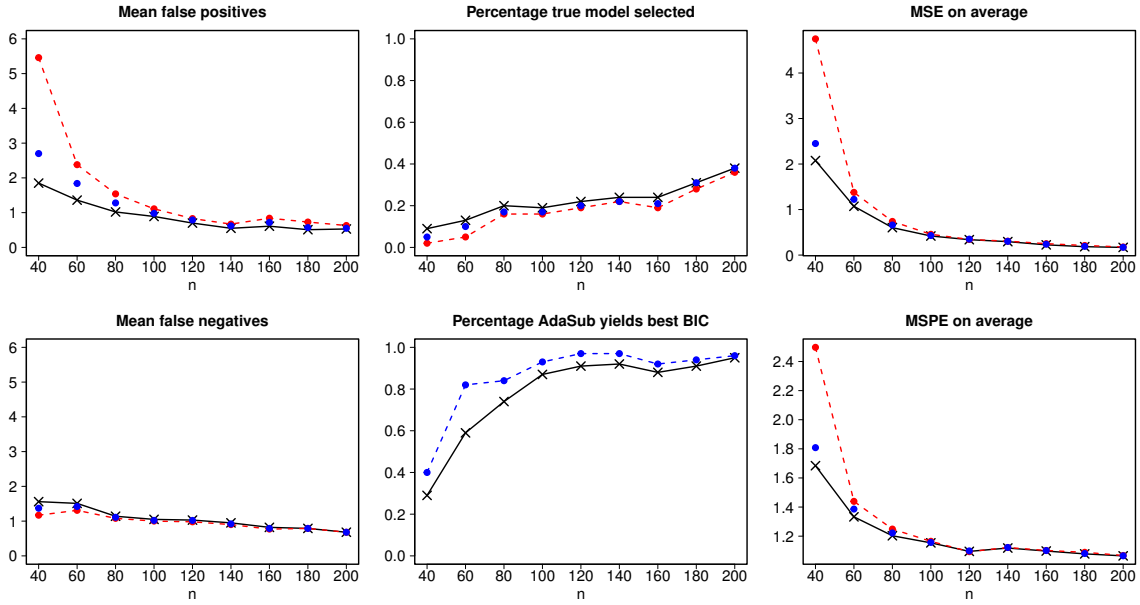
A reason for the undesirable behaviour of the best BIC model is that the discrete nature of the ℓ_0-penalty can lead to "overfitting" of the criterion, since the optimization is carried out among all possible $2^{30} \approx 10^9$ models. This general problem underlying Best Subset Selection has been addressed both by the statistics community (see e.g. Breiman, 1996) and the machine learning community (see e.g. Loughrey and Cunningham, 2005). In particular, Tibshirani (2015) analyses the effective degrees of freedom of Best Subset Selection and demonstrates that it has generally much larger degrees of freedom than the expected number of parameters in the finally selected model by Best Subset Selection (due to additional *search degrees of freedom*).

Remark 5.1. The simulation results presented above show that AdaSub can mitigate the overfitting problem of Best Subset Selection with BIC in the given situation of a sparse underlying true model. This empirical observation is in accordance with the theoretical result in Theorem 4.12 of Section 4.1, namely that the thresholded model $\hat{S}_{0.9}$ from AdaSub is only guaranteed to contain all the variables which are "stable" in the sense of OIP (see Definition 4.4). Intuitively, the OIP condition can also be interpreted as a kind of discrete "smoothness" condition for the criterion $C : \mathcal{M} \to \mathbb{R}$ (to be maximized): AdaSub will generally only converge against the C-optimal model $S^* = \{j_1, \ldots, j_{s^*}\}$ with $|S^*| = s^*$, if there exists a "learning path" $(k_1, \ldots, k_{s^*})$ — a permutation of $(j_1, \ldots, j_{s^*})$ — in the sense of OIP, which particularly satisfies

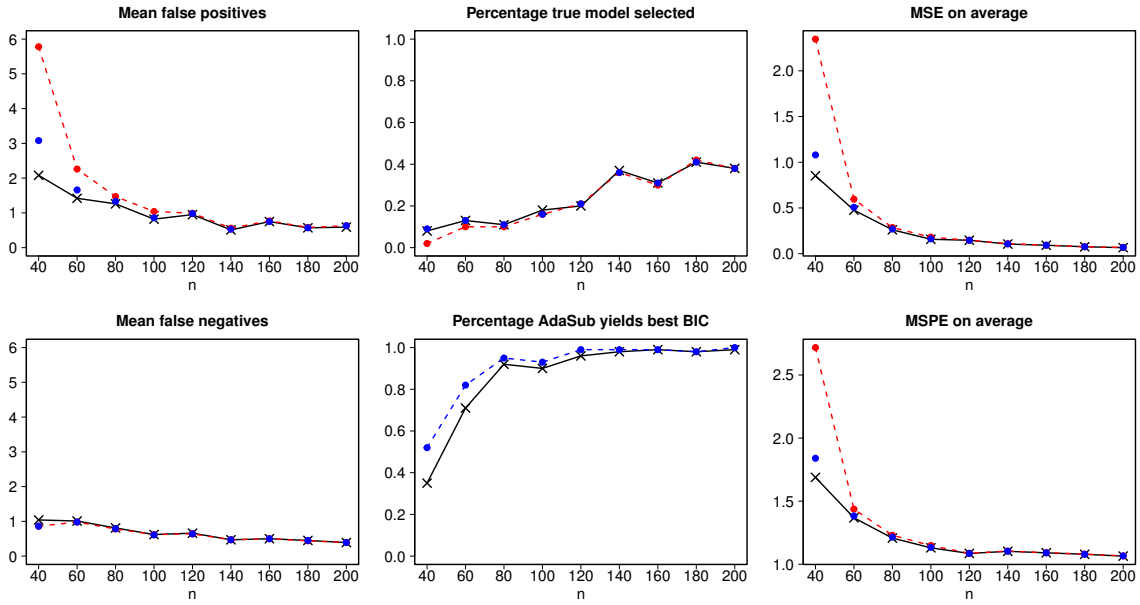
$$C(\emptyset) < C(\{k_1\}) < C(\{k_1, k_2\}) < \cdots < C(\{k_1, \ldots, k_{s^*}\}) = C(S^*)\,. \tag{5.3}$$

Thus, on a rough level, OIP requires that the maximum of C in S^* is not too "sharp", so that overfitting might be avoided. This reasoning is very vaguely related to the Bayesian framework where maxima of the posterior with lots of probability mass in their neighbourhood are to be preferred, as well as related to the framework of neural nets where "flat" minima of the loss function are thought to be associated with better generalization (see e.g. Buntine and Weigend, 1991 and Hochreiter and Schmidhuber, 1997). In conclusion, AdaSub is a stochastic algorithm which does not search as "aggressively" as Best Subset Selection in the generally large space of possible models and thus apparently leads to smaller search degrees of freedom in practice. It would be an interesting, though challenging topic for future research to formally analyse the search degrees of freedom of AdaSub within the theoretical framework developed in Tibshirani (2015).

The tendency that AdaSub selects sparser models in "unstable" situations is also observed in further simulations with different correlation structures of the design matrix $\boldsymbol{X}$. The results of additional simulations using an equal-correlation structure with $c = 0.7$ and a block-correlation structure with $b = 10$ blocks and $c = 0.5$ can be found in Figure 5.4 below.



(a) Equal-correlation structure ($c = 0.7$)



(b) Block-correlation structure ($b = 10$ blocks and $c = 0.5$)

Figure 5.4.: Results of low-dimensional examples ($p = 30$) with (a) equal-correlation structure and (b) block-correlation structure: Comparison of $\hat{S}_{0.9}$ (black) and $\hat{S}_{\mathrm{b}}$ (blue) from AdaSub with BIC-optimal model S^* (red) in terms of mean false positives/ false negatives, percentage of selecting the true model S_0, percentage of agreement between AdaSub models and S^*, MSE and (averaged) MSPE on independent test set with sample size 100.

5.1.2. High-dimensional setting

We now turn to a high-dimensional scenario, in which both the sample size n and the number of explanatory variables p tend to infinity with a certain rate. In particular, we consider the setting $p = 10 \times n$ where n is increased from 40 to 140 in steps of size 20 (and thus p is increased from 400 to 1400). For each pair (n, p) we simulate 100 datasets according to the Simulation Setup 5.1 described above.

We compare the thresholded model $\hat{S}_\rho$ with $\rho = 0.9$ from AdaSub with different well-known methods for high-dimensional variable selection: We consider the Lasso, Forward Stepwise Selection, the SCAD, the Adaptive Lasso and Stability Selection with Lasso. For the computation of the Lasso and the Adaptive Lasso we use the R-package `glmnet`, for Stability Selection the R-package `c060` and for the SCAD the R-package `ncvreg`. In AdaSub we choose the (negative) EBIC_γ with parameter $\gamma = 0.6$ as the criterion C; additionally we set $q = 5$, $K = n$ and $T = 5000$. Note that $p = \mathcal{O}(n^k)$ with $k = 1$, so that we have $\gamma > 1 - \frac{1}{2k}$ and thus $\text{EBIC}_{0.6}$ is a variable selection consistent criterion for the given asymptotic setting (see Theorem 2.2 in Chapter 2).

For comparison reasons we also choose the regularization parameter of the Lasso, the SCAD and Forward Stepwise Selection according to $\text{EBIC}_{0.6}$. Instead of the usual Lasso and SCAD estimators we use versions of the Lasso-OLS-hybrid (see the discussion after Definition 2.25 of Section 2.4.3; see also Efron et al., 2004 and Belloni and Chernozhukov, 2013), where we compute the $\text{EBIC}_{0.6}$-values of all models along the Lasso-path (and the SCAD-path, respectively) using the ordinary least-squared (OLS) estimators and finally select the model (with corresponding OLS estimator) yielding the lowest $\text{EBIC}_{0.6}$-value. The additional tuning parameter of the SCAD penalty is set to the default value of 3.7 (see Fan and Li, 2001). For the Adaptive Lasso we derive the initial estimator with the usual Lasso where the regularization parameter is chosen using 10-fold cross-validation and compute in the second step an additional Lasso path where the regularization parameter is chosen according to $\text{EBIC}_{0.6}$. The parameters for Stability Selection (with 100 iterations, i.e. 100 different subsamples) are chosen such that the expected number of false positives is bounded by 1 (using the per-family error rate bound, compare Theorem 2.6 in Chapter 2). The final estimator for Stability Selection is the OLS estimator for the model identified by Stability Selection.

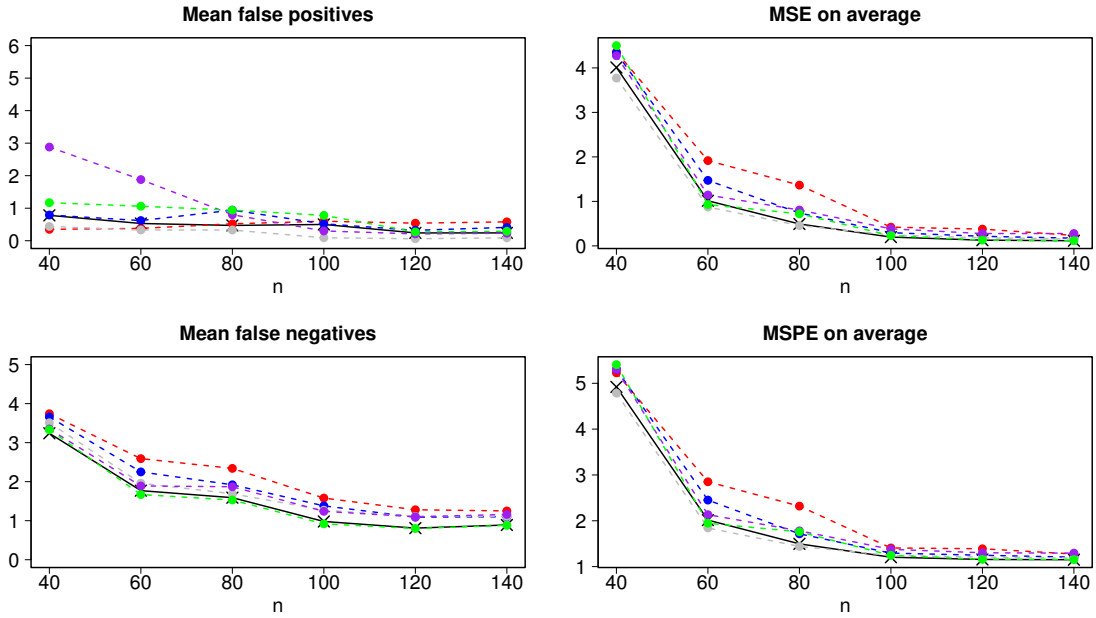


Figure 5.5.: High-dimensional example ($p = 10n$) with equal-correlation structure ($c = 0$): Comparison of AdaSub (black), Lasso (blue), Forward Stepwise (green), SCAD (purple), Adaptive Lasso (gray) and Stability Selection (red) in terms of mean false positives/ false negatives, MSE and (averaged) MSPE on independent test set with sample size 100.

Figure 5.5 summarizes the results of the high-dimensional simulation study in the case of an equal-correlation structure with $c = 0$ (i.e. independent explanatory variables). It is apparent that the Lasso and even more Stability Selection with Lasso yield models which miss more important variables than the other methods (i.e. larger numbers of mean false negatives). SCAD selects too many false positives if the sample size is small. For this correlation structure, the Adaptive Lasso selects on average less false positives than AdaSub, but at the prize of an increased number of mean false negatives. Forward Stepwise Selection and AdaSub show similar performances in the given situation of independence, though Forward Stepwise selects a larger number of false positives for small sample sizes. The Mean Squared Errors (MSE) and the Mean Squared Prediction Errors (MSPE) of AdaSub and the Adaptive Lasso are similar and the lowest among the methods compared yielding the best estimative and predictive performances in the given situation.

Figure 5.6 summarizes the results of the high-dimensional simulation study for a Toeplitz-correlation structure with large correlation $c = 0.9$. Here, AdaSub clearly yields the best variable selection performance on average, while the other methods select more false positives in general without a reduction in the mean of false negatives. Note that even though we

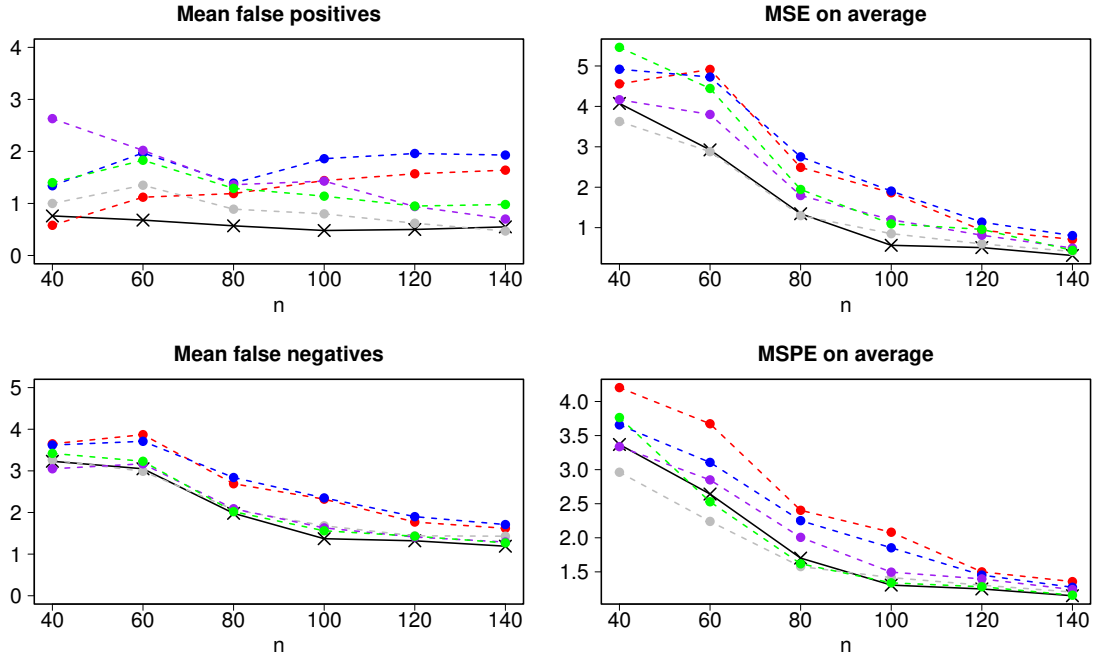
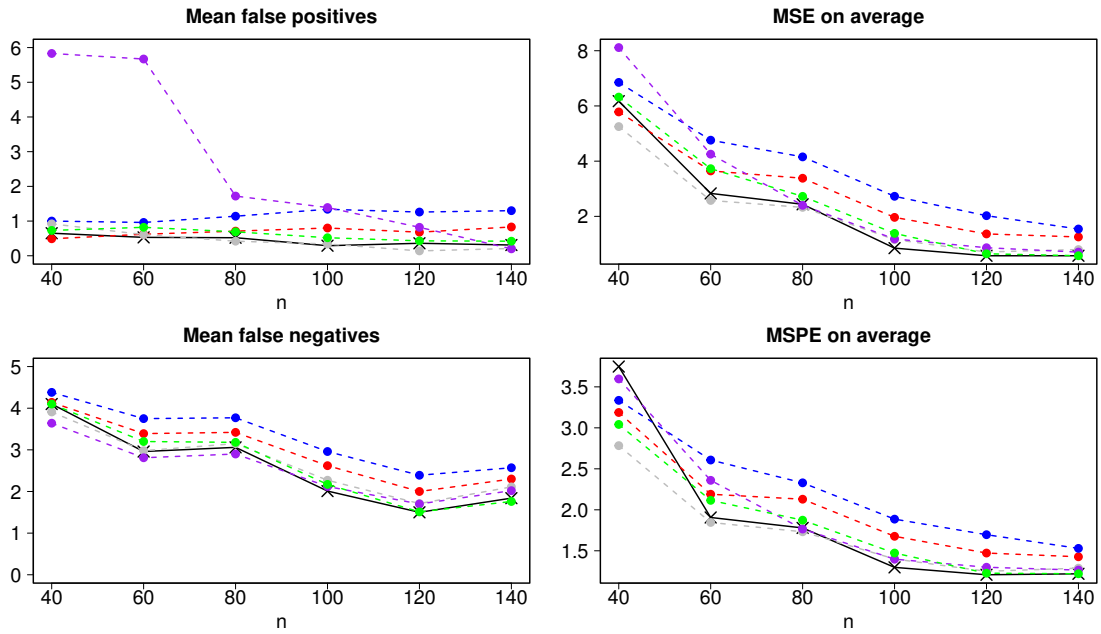


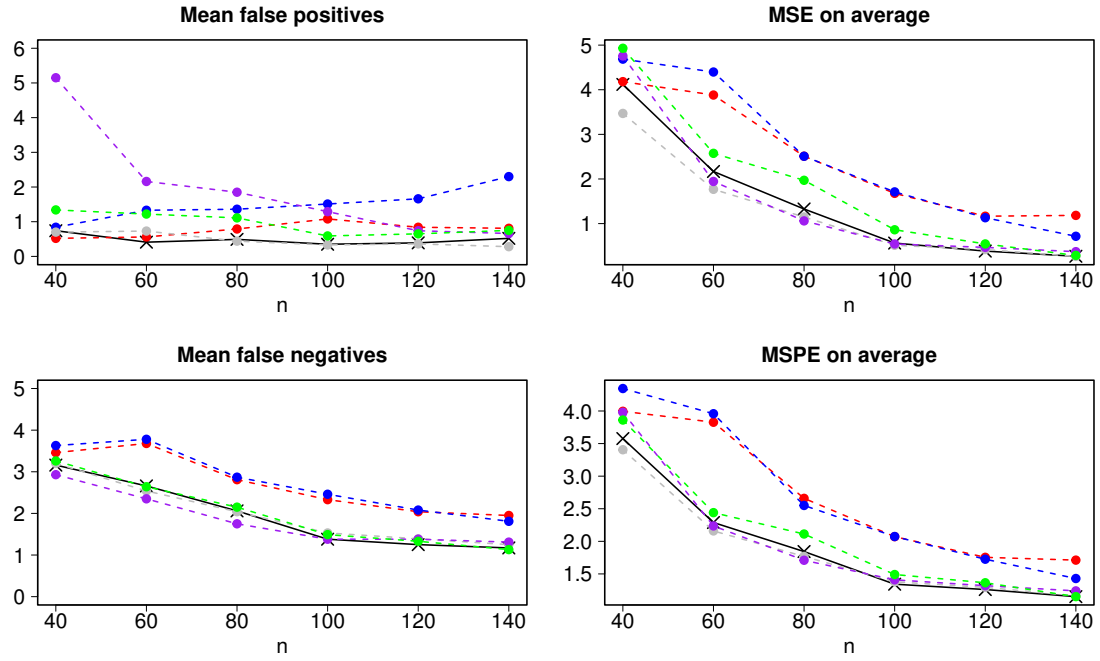
Figure 5.6.: High-dimensional example ($p = 10n$) with Toeplitz-correlation structure ($c = 0.9$): Comparison of AdaSub (black), Lasso (blue), Forward Stepwise (green), SCAD (purple), Adaptive Lasso (gray) and Stability Selection (red) in terms of mean false positives/ false negatives, MSE and (averaged) MSPE on independent test set with sample size 100.

choose the threshold in Stability Selection such that the expected number of false positives should be bounded by 1, this aim is not achieved for larger values of n. AdaSub and the Adaptive Lasso perform similar and best when the aim is estimation of the regression coefficients. AdaSub and Forward Stepwise Selection yield similar predictive performance, while the Adaptive Lasso yields smaller MSPE than AdaSub for small values of n. This is due to the fact that the Adaptive Lasso selects larger models (with more false positives) than AdaSub, which turns out to be beneficial for prediction in the given situation with large positive correlations between the explanatory variables.

The summary of the results of additional simulations using an equal-correlation structure with $c = 0.7$ and a block-correlation structure with $b = 10$ blocks and $c = 0.5$ can be found in Figure 5.7 below. All in all, the performance of AdaSub is very competitive to state-of-the-art methods like the Adaptive Lasso or SCAD and AdaSub can produce better results than these methods in certain situations. Additionally, AdaSub outperforms Stability Selection in all of the situations considered.



(a) Equal-correlation structure ($c = 0.7$)



(b) Block-correlation structure ($b = 10$ blocks and $c = 0.5$)

Figure 5.7.: Results of high-dimensional examples ($p = 10n$) with (a) equal-correlation structure and (b) block-correlation structure: Comparison of AdaSub (black), Lasso (blue), Forward Stepwise (green), SCAD (purple), Adaptive Lasso (gray) and Stability Selection (red) in terms of mean false positives/ false negatives, MSE and (averaged) MSPE on independent test set with sample size 100.

Finally, we briefly comment on the performance of the PC-simple algorithm (Bühlmann et al., 2010) and Tilting (Cho and Fryzlewicz, 2012). As already mentioned in Section 3.7, Cho and Fryzlewicz (2012) demonstrate that both proposed versions (TCS1 and TCS2) of Tilting outperform the PC-simple algorithm in many situations. Our simulations confirm this finding when using the EBIC_γ with $\gamma = 1$ for final model selection. However, we observed that TCS2 often yields a large number of false positives when using EBIC_γ with $\gamma = 0.6$. In addition it seems that, in the scenarios considered, Tilting is not as competitive as AdaSub when the explanatory variables are highly correlated. Since the computational time for Tilting can be quite long in comparison to AdaSub for datasets with p moderately large (e.g., for $p = 1400$ and $n = 140$, Tilting takes more than seven minutes per dataset on a 3.2-GHz processor, while AdaSub with $T = 5000$ iterations takes around 30 seconds), for practical reasons we do not report the detailed performance measures of Tilting for the employed simulation study. A comparison of the algorithms with respect to these issues and their efficiency under different settings is worth for further investigation.

5.2. Real data examples

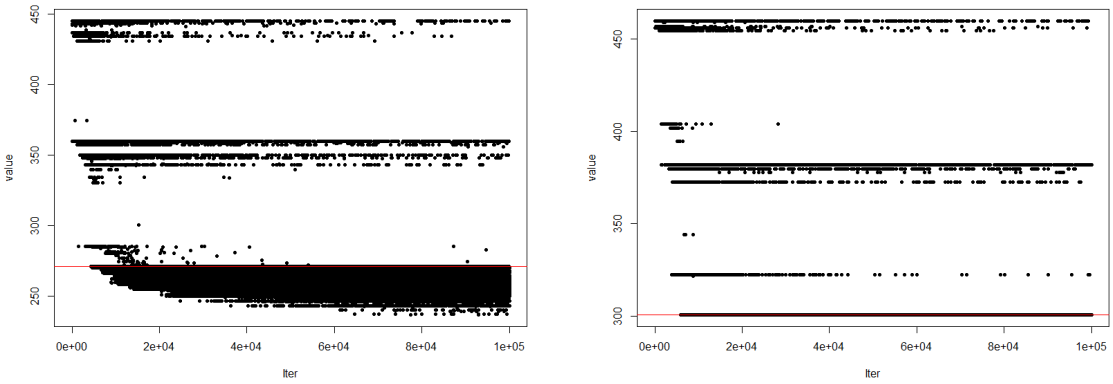
In this section we discuss the application of AdaSub on (ultra)-high-dimensional real data examples. For comparison reasons we consider the two examples analysed in Song and Liang (2015a), where their proposed split-and-merge approach (SAM) is compared to other state-of-the-art methods (see also Section 3.7). The two datasets are publicly available in JRSS(B) Datasets, Vol. 77:5 (2015). Song and Liang (2015a) show that in both examples SAM performs favourably in comparison to hybrid methods like (I)SIS-Lasso and (I)SIS-SCAD, so we do not include the results of these methods here. (I)SIS-Lasso and (I)SIS-SCAD are acronyms for the combination of a screening step with (Iterated) Sure Independence Screening (Fan and Lv 2008) and then a selection step of the final model with Lasso and SCAD, respectively (see Section 2.6). A special intention of this section is to show that it is computationally feasible to apply the AdaSub method even in the situation of ultra-high-dimensional data with ten thousands of explanatory variables and that an additional screening step is not necessarily needed.

5.2.1. Metabolic quantitative trait loci dataset

As the first example we consider a metabolic quantitative trait loci experiment (Dumas et al., 2007). The preprocessed data from Song and Liang (2015a) is used, which consists of $n = 50$ subjects with $p = 9,988$ explanatory variables (Single Nucleotide Polymorphisms, SNPs) that arise from a genome-wide analysis of genes for alanine amino transferase enzyme elevation in liver. A specific metabolite bin that discriminates between the disease status of a subject is chosen as the response variable. For details concerning this example we refer to Dumas et al. (2007), Bottolo and Richardson (2010) and Song and Liang (2015a).

In the described dataset all explanatory variables are categorical with three possible levels: "homozygous common" (coded as 0), "heterozygous" (coded as 1) and "homozygous rare" (coded as 2). Recall that AdaSub has principally no problems in handling categorical variables with more than two levels and can be applied without any methodological changes (just treat the categorical variables as "factors" in the fitting process of individual GLMs included in the sampled subspaces). However, the efficient "leaps and bounds" algorithm implemented in the R-package `leaps` (Lumley and Miller, 2009) does not allow to enforce that either all dummy variables corresponding to one categorical predictor should be included or that all of them should not be included in a model. For practical and comparison reasons (Song and Liang, 2015a), we assume that the explanatory variables are treated as ordinal variables using the equidistant scores $(0, 1, 2)$, as described. In fact, additional experiments (not presented here) show that the variable selection results are not sensitive to different scores assigned to the factor levels (non-equidistant scores included).

We first apply the AdaSub algorithm with $q = 5$, $K = n$ and $T = 100,000$ and choose the $\text{EBIC}_{0.6}$ as the selection criterion (yielding computational time of approx. 34 minutes). The evolution of the values $\text{EBIC}_{0.6}(S^{(t)})$ along the iterations (t) is given in Figure 5.8 (a). Here, AdaSub does not converge to a unique model. In particular, the thresholded model $\hat{S}_{0.9}$ consisting of two variables (SNPs rs17041311 and rs17392161) and the "best" model $\hat{S}_{\text{b}}$ consisting of eleven variables do not coincide. This indicates that OIP is not satisfied in the given situation and the model $\hat{S}_{\text{b}}$ with eleven variables is not "stable". However, the thresholded model $\hat{S}_{0.9}$ consists of the two important SNPs which are also selected by the SAM approach.



(a) $\gamma = 0.6$ (b) $\gamma = 1$

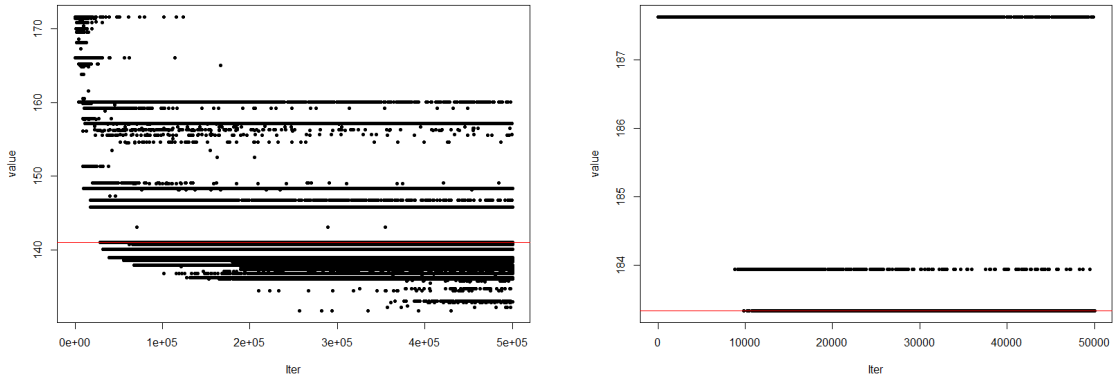
Figure 5.8.: AdaSub for SNP-data. Plot of the evolution of $\text{EBIC}_\gamma(S^{(t)})$ along iterations (t). The red line indicates the EBIC_γ-value of the thresholded model $\hat{S}_{0.9}$.

We now apply AdaSub with $q = 5$, $K = n$ and $T = 100,000$ and choose the EBIC_1 as a selection criterion that enforces more sparsity (computational time approx. 9 minutes). The evolution of the values $\text{EBIC}_1(S^{(t)})$ along the iterations (t) is given in Figure 5.8 (b). Now $\hat{S}_{0.9}$ and $\hat{S}_{\text{b}}$ coincide, which both consist of the two important SNPs (rs17041311 and rs17392161). The agreement between $\hat{S}_{0.9}$ and $\hat{S}_{\text{b}}$ indicates that OIP may hold and that the algorithm has converged correctly. This result emphasizes the stability of the found model with two SNPs.

For this example, the output of Stability Selection with error control of one expected false positive is not "stable" in itself, meaning that the selected model differs if Stability Selection is repeatedly applied on the dataset, identifying two, one or none covariates as "important" over different runs of the algorithm. In any case, Stability Selection does not select the important SNP rs17041311.

5.2.2. Polymerase chain reaction dataset

Our second example considers a polymerase chain reaction (PCR) dataset (Lan et al., 2006). The preprocessed data from Song and Liang (2015a) is used, which consists of $n = 60$ samples (mice) with $p = 22,575$ explanatory variables (expression levels of genes). Phosphoenolpyruvat-carboxykinase (physiological phenotype) is chosen as the response variable. For details concerning this example we refer to Lan et al. (2006) and Song and Liang (2015a).



(a) $\gamma = 0.6$ (b) $\gamma = 1$

Figure 5.9.: AdaSub for PCR-data. Plot of the evolution of $\text{EBIC}_\gamma(S^{(t)})$ along iterations (t). The red line indicates the EBIC_γ-value of the thresholded model $\hat{S}_{0.9}$.

We first apply the AdaSub algorithm with $q = 5$, $K = n$ and $T = 500,000$ and choose $\text{EBIC}_{0.6}$ as the selection criterion (computational time approximately 2 hours and 9 minutes). The evolution of the values $\text{EBIC}_{0.6}(S^{(t)})$ along the iterations (t) is given in Figure 5.9 (a). Again, as in the first example, the criterion $\text{EBIC}_{0.6}$ seems to be too liberal for the given situation resulting in high uncertainty concerning the $\text{EBIC}_{0.6}$-optimal model and (possible) failure of the OIP condition. The thresholded model $\hat{S}_{0.9}$ selected by AdaSub consists of five variables (genes): 1437871_at, 1438937_x_at, 1442771_at, 1446035_at, 1455361_at. The "best" model $\hat{S}_\text{b}$ includes the five genes in the thresholded model $\hat{S}_{0.9}$ as well as five further genes: 1428239_at, 1433056_at, 1440505_at, 1444471_at, 1445645_at.

In order to compare the predictive performances of the selected models we compute the mean and median leave-one-out-cross-validation squared error (CV-error) for each fixed model as described in Song and Liang (2015a). The mean CV-error of the thresholded model $\hat{S}_{0.9}$ with five genes is 0.116; the mean CV-error of the "best" model $\hat{S}_\text{b}$ with ten genes is 0.030. These errors are of the same order or even lower than the errors of the best SAM model with five and ten explanatory variables, respectively (compare Figure 5 in Song and Liang, 2015a).

The final model selected by SAM consists of six genes: 1429089_s_at, 1430779_at, 1432745_at, 1437871_at, 1440699_at, 1459563_x_at. Note that the model selected by SAM has only one gene (1437871_at) in common with the models $\hat{S}_{0.9}$ and $\hat{S}_\text{b}$ selected by AdaSub. In order to compare this model from SAM to a model with six genes selected

by AdaSub we proceed in the following way: Let $g : \mathcal{P} \to \mathcal{P}$ be a permutation such that $r^{(T)}_{g(1)} \geq r^{(T)}_{g(2)} \geq \ldots \geq r^{(T)}_{g(p)}$. Assuming no "ties", for $k \in \mathcal{P}$ we define

$$\hat{S}_k := \{j \in \mathcal{P};\ g^{-1}(j) \geq k\} \tag{5.4}$$

to be the thresholded model from AdaSub with exactly $|\hat{S}_k| = k$ variables. Then we find that in the given example the model $\hat{S}_6$ consists of six genes: 1428239_at, 1437871_at, 1438937_x_at, 1442771_at, 1446035_at, 1455361_at. Even though this model is very different from the model selected by SAM (with only the single common gene 1437871_at), it has similar predictive performance: The mean and median CV-errors of $\hat{S}_6$ are 0.090 and 0.041, respectively; the mean and median CV-errors of the model with six genes selected by SAM are 0.084 and 0.044.

We now apply AdaSub with $q = 5$, $K = n$ and $T = 50,000$ and choose EBIC_1 as a selection criterion that enforces more sparsity (computational time approximately 5 minutes). The evolution of the values $\text{EBIC}_1(S^{(t)})$ along the iterations (t) is given in Figure 5.9 (b). Now $\hat{S}_{0.9}$ and $\hat{S}_{\text{b}}$ coincide, which both consist of only one gene (1438937_x_at). In this case, the algorithm seems to have converged correctly. Note that for the criterion $\text{EBIC}_{0.6}$ the gene 1438937_x_at is also included in the thresholded model $\hat{S}_{0.9}$, in the "best" model $\hat{S}_{\text{b}}$ and in the thresholded model $\hat{S}_1$ with exactly one gene selected by AdaSub, whereas it is not included in the final model selected by SAM.

Stability Selection produces a model with only one variable (gene 1437871_at) even for large error bounds (e.g. 50 expected false positives). This model has mean and median CV-errors of 0.442 and 0.179, respectively. The model from AdaSub with one variable (gene 1438937_x_at) yields lower mean and median CV-errors of 0.403 and 0.158, respectively.

6. Modifications of AdaSub

In the preceding chapter we have investigated the performance of AdaSub for variable selection in high-dimensional normal linear models. Note that the proposed AdaSub method is very general and can for example be applied to any variable selection problem in the framework of generalized linear models (GLMs). However, the practical problem is that for GLMs other than the normal linear model — to the best of our knowledge — there is no efficient algorithm like "leaps-and-bounds" (see e.g. Lumley and Miller, 2009) which could be used for solving the sampled variable selection sub-problems in AdaSub within reasonable computational time. In particular, a full enumeration is costly since the ML-estimators for the single models are not given in closed form, in general. Therefore, the identification of the optimal model according to the used criterion quickly becomes computationally intractable even for relatively low-dimensional problems.

In this chapter we propose simple modifications of the original AdaSub algorithm where more efficient greedy methods are used in place of a full model enumeration in order to derive approximate solutions for the sub-problems. Natural candidates for such greedy methods are Forward and Backward Stepwise Selection, leading to variants of AdaSub which we call FoAdaSub and BackAdaSub, respectively. These methods are introduced in Section 6.1, along with a short discussion of their computational complexity.

In Section 6.2 we analyse the limiting behaviour of the proposed variants of AdaSub. It turns out that we do not gain a lot by considering Forward Stepwise Selection for the solution of the sub-problems. In fact, in Section 6.2.1 we show that FoAdaSub converges with the number of iterations against the model selected by usual Forward Stepwise Selection (see Theorem 6.1). On the other hand, we demonstrate that BackAdaSub can serve as a good surrogate algorithm for the original AdaSub method, in the sense that the selected models by BackAdaSub often agree with the models selected by AdaSub. In particular, we provide conditions under which BackAdaSub converges against the best model according to

the employed criterion and we show that these conditions imply the correct convergence of the original AdaSub algorithm as well (see Theorem 6.2). In Section 6.2.2 we confirm the theoretical results concerning the limiting behaviour of FoAdaSub and BackAdaSub via a simulation study in the simpler setting of normal linear models, in which it is feasible to compare them with the original AdaSub method.

In Section 6.3 we investigate the performance of the proposed methods on simulated data examples for high-dimensional logistic regression models. We demonstrate that BackAdaSub is very competitive for high-dimensional variable selection in comparison to other well-known methods like usual Forward Stepwise Selection, the Lasso, SCAD, the Adaptive Lasso and Stability Selection. In Section 6.4 we illustrate the effectiveness of BackAdaSub when applied on high-dimensional real data examples from the field of genomics with binary response data. In Section 6.5 we summarize the main results of this chapter.

6.1. FoAdaSub and BackAdaSub

In this section we introduce variants of the original AdaSub algorithm which solve the low-dimensional sub-problems by using heuristic instead of exact optimization methods. As before, the aim is to identify the C-optimal model $S^* = f_C(\mathcal{P}) \in \mathcal{M}$, which maximizes a given selection criterion C. For simplicity, we again impose assumption (3.2), so that $C(S) \neq C(S')$ for all $S, S' \in \mathcal{M}$ with $S \neq S'$.

We focus on two well-known stepwise methods, called Forward Stepwise Selection (FS) and Backward Stepwise Selection (BS). We refer to Section 2.3.2 for a detailed discussion of FS (Algorithm 2.1) and BS (Algorithm 2.2), but we briefly recall the main ideas here: In FS one begins with the "null" model including only an intercept (i.e. $S = \emptyset$) and then successively adds one explanatory variable at each time (namely that variable which most improves the criterion C); in BS one starts with the "full" model including all available explanatory variables (i.e. $S = \{1, \ldots, p\}$) and then successively removes that explanatory variable leading to the largest value of the criterion C, until the "null" model is reached. The sequence of subsets $S_F^{(0)} \subset S_F^{(1)} \subset \ldots \subset S_F^{(m)}$ in FS is constructed in such a way that for $i = 1, \ldots, m$ we have

$$C\left(S_F^{(i)}\right) > C\left(S_F^{(i-1)} \cup \{l\}\right) \quad \text{for all } l \in \mathcal{P} \setminus S_F^{(i)}, \tag{6.1}$$

while for the sequence of subsets $S_B^{(0)} \subset S_B^{(1)} \subset \ldots \subset S_B^{(m)}$ in BS, for $i = 1, \ldots, m$, we have

$$C\left(S_B^{(i-1)}\right) > C\left(S_B^{(i)} \setminus \{l\}\right) \quad \text{for all } l \in S_B^{(i-1)}. \tag{6.2}$$

Under the made assumption that $C(S) \neq C(S')$ for all $S, S' \in \mathcal{M}$ with $S \neq S'$, the strict inequalities in (6.1) and (6.2) hold and the finally selected subsets $\hat{S}_F = \arg\max_i C(S_F^{(i)})$ and $\hat{S}_B = \arg\max_i C(S_B^{(i)})$ by FS and BS are unique, respectively. Furthermore, recall that while FS can always be applied even for high-dimensional data, BS is only feasible if the number of variables p is at most $n - p_0$, where n is the sample size and p_0 denotes the number of parameters of the "null" model. However, note that it is indeed possible to apply BS when starting from lower-dimensional submodels $S \in \mathcal{M}$ with $|S| \leq n - p_0$, which allows us to define the variant of AdaSub based on BS below (see Algorithm 6.2). In simulation studies we observe that the use of Forward Stepwise Selection with "early stopping" (referred to as FS2, compare Section 2.3.2) is often beneficial in high-dimensional situations where p is larger than n. Therefore, in the following we focus on the variant FS2 (which is also the version commonly used in practice).

We introduce some further notation in analogy to the definition of the map f_C in Notation 3.1 of Section 3.2.

Notation 6.1. Let $h_C(V) \in \mathcal{M}$ denote the final subset selected by FS2 if only variables in $V \subseteq \mathcal{P}$ are considered and let $g_C(V) \in \mathcal{M}$ denote the final subset selected by BS if one starts BS with the subset $V \in \mathcal{M}$. Note that $g_C(V)$ is only defined for $V \in \mathcal{M}$, i.e. for subsets $V \subseteq \mathcal{P}$ with $|V| \leq n - p_0$, while $h_C(V)$ is defined for all subsets $V \subseteq \mathcal{P}$.

With this notation it is straightforward to define variants of AdaSub, where the solution of the low-dimensional sub-problems is based on Forward Selection (FS2) or Backward Selection (BS). We call these variants FoAdaSub (given as Algorithm 6.1) and BackAdaSub (given as Algorithm 6.2), respectively.

Algorithm 6.1 Forward Adaptive Subspace (FoAdaSub) method

Input, algorithm and output is the same as for AdaSub (see Algorithm 3.1) except that step (c) of the AdaSub algorithm is replaced by the following step:

(c) Compute $S^{(t)} = h_C(V^{(t)})$, i.e. compute the subset selected by FS2 if only variables in $V^{(t)}$ are considered.

Algorithm 6.2 Backward Adaptive Subspace (BackAdaSub) method

Input, algorithm and output is the same as for AdaSub (see Algorithm 3.1) except that step (c) of the AdaSub algorithm is replaced by the following two steps:

(c1) If $|V^{(t)}| > n - p_0$, set $V^{(t)} \leftarrow \text{sample}(V^{(t)}, \text{size} = n - p_0)$ (without replacement).

(c2) Compute $S^{(t)} = g_C(V^{(t)})$, i.e. compute the subset selected by BS when starting with the subset $V^{(t)}$.

Notation 6.2. In the following, for brevity, we use the same notation for AdaSub, FoAdaSub and BackAdaSub, e.g. $r_j^{(t)}$ denotes the selection probability of X_j in iteration $t+1$ as well as $\hat{S}_\text{b}$ and $\hat{S}_\rho$ for $\rho \in (0, 1)$ denote the selected models by the different algorithms (the implied meaning should be clear from the context).

As we have noted above, BS cannot be directly applied to subsets $V \subseteq \mathcal{P}$ with $|V| > n - p_0$. Therefore, in step (c1) of BackAdaSub, if $|V^{(t)}| > n - p_0$ then we first replace $V^{(t)}$ randomly by a subset of $V^{(t)}$ of size $n - p_0$, so that BS can be applied to that reduced subset in step (c2). However, in practice this case will only occur with non-negligible probability if the sample size n is very small or if the criterion C does not enforce enough sparsity. In the following considerations we will therefore assume for simplicity that step (c1) of BackAdaSub is not needed.

The main motivation for replacing the full enumeration approach in AdaSub by greedy stepwise methods is to save computational resources and hereby to enable the application of the method to models where the evaluation of the criterion C takes more time than for normal linear models. Recall that in GLMs like logistic or Poisson regression models the MLEs of the regression coefficients are generally not given in closed form and therefore each evaluation $C(V)$ for some $V \in \mathcal{M}$ requires the solution of a separate (continuous) optimization problem by using numerical methods.

For a rough estimate of the computational complexity assume that it takes at most L operations to compute (an approximation to) $C(V)$ for a subset $V \in \mathcal{M}$. Furthermore, assume that we have $|V^{(t)}| \leq U_C \leq n - p_0$ for all iterations t of the algorithms. Then recall that the computational complexity of AdaSub with $T \in \mathbb{N}$ iterations when using full enumeration for the sub-problems is roughly bounded by $2^{U_C} \times L \times T$ (see Section 3.6). On the other hand, under the same assumptions, the computational complexity of FoAdaSub

or BackAdaSub with T iterations is roughly bounded by $U_C^2 \times L \times T$. Since $U_C^2 \ll 2^{U_C}$ (say, for $U_C \geq 10$), using FoAdaSub or BackAdaSub can yield a significant reduction of the computational time in comparison to the original AdaSub method. A more refined analysis of the (average) computational complexity of AdaSub and its variants FoAdaSub and BackAdaSub is certainly desirable, but this is apparently much more difficult due to the stochastic nature of the algorithms (compare Section 3.6). Nevertheless, even the very rough discussion on the complexity clearly indicates that the greedy variants of AdaSub are more efficient than a naive full enumeration for solving the sampled sub-problems.

6.2. Limiting properties of FoAdaSub and BackAdaSub

In this section we investigate the limiting properties of FoAdaSub and BackAdaSub, i.e. the behaviour of the algorithms if the number of iterations T goes to infinity. After presenting a thorough theoretical analysis of these limiting properties in Section 6.2.1, we confirm the findings through simulated data examples in Section 6.2.2.

6.2.1. Theoretical results

The following theorem shows that FoAdaSub converges against the model selected by Forward Stepwise Selection (FS2). This implies that (in the limit $T \to \infty$) we do not gain any statistical advantages if we use FS2 for the solutions of the sub-problems in the original AdaSub algorithm, since we could just apply FS2 directly to the original high-dimensional problem. In addition, we note that FoAdaSub does not yield significant computational advantages over the already quite efficient FS2 algorithm, so that we do not recommend to use FoAdaSub in place of FS2.

Theorem 6.1. *FoAdaSub converges against the model $\hat{S}_F$ selected by Forward Stepwise Selection (FS2). In more detail, consider the selection criterion $C : \mathcal{M} \to \mathbb{R}$ and let $\hat{S}_F = h_C(\mathcal{P})$ be the model selected by FS2. Then for all $j \in \mathcal{P}$ it holds (for Algorithm 6.1)*

$$r_j^{(t)} \xrightarrow{a.s.} \begin{cases} 1 & , \text{ if } j \in \hat{S}_F, \\ 0 & , \text{ if } j \notin \hat{S}_F, \end{cases} \quad \text{for } t \to \infty\,. \tag{6.3}$$

Proof. Let $\hat{S}_F = h_C(\mathcal{P}) = \{j_1, \ldots, j_{s_F}\}$ be the subset selected by FS2 of size $|\hat{S}_F| = s_F$.

We define $k_1 := \arg\max_{j \in \mathcal{P}} C(\{j\})$ and for $i = 2, \ldots, s_F$ we inductively define

$$k_i := \underset{j \in \mathcal{P} \setminus \{k_1, \ldots, k_{i-1}\}}{\arg\max} C(\{k_1, \ldots, k_{i-1}, j\}) \,.$$

By the definition of the map h_C we have

$$\hat{S}_F = h_C(\mathcal{P}) = \{j_1, \ldots, j_{s_F}\} = \{k_1, \ldots, k_{s_F}\}$$

and for all $i = 1, \ldots, s_F$ it holds

$$k_i \in h_C(V) \quad \text{for all } V \subseteq \mathcal{P} \text{ with } \{k_1, \ldots, k_i\} \subseteq V \,. \tag{6.4}$$

Furthermore, for all $j \in \mathcal{P} \setminus \hat{S}_F$ it holds

$$j \notin h_C(V) \quad \text{for all } V \subseteq \mathcal{P} \text{ with } \hat{S}_F \subseteq V \,. \tag{6.5}$$

The assertion follows by applying Theorem 4.5 and using arguments along the lines of the proof of Theorem 4.8 in Section 4.1: From equation (6.4) we successively conclude that $r_{k_i}^{(t)} \overset{\text{a.s.}}{\to} 1$, $t \to \infty$, for $i = 1, \ldots, s_F$ and from equation (6.5) we obtain $r_j^{(t)} \overset{\text{a.s.}}{\to} 0$, $t \to \infty$, for all $j \in \mathcal{P} \setminus \hat{S}_F$. □

In Theorem 4.8 of Section 4.1 we have shown that the original AdaSub algorithm "converges correctly" against the best model $S^* = f_C(\mathcal{P})$ according to the used criterion C, provided that the ordered importance property (OIP) is satisfied. Recall that by "correct convergence" it is meant that we almost surely have $r_j^{(t)} \to 1$ if $j \in S^*$ and $r_j^{(t)} \to 0$ if $j \in \mathcal{P} \setminus S^*$, as $t \to \infty$ (see Definition 4.2 in Section 4.1). It is desirable to find sufficient conditions similar to OIP under which the correct convergence against S^* of BackAdaSub can be guaranteed. In the next definition we introduce the modified OIP (MOIP) which adjusts the original OIP such that it yields a sufficient condition for the correct convergence of BackAdaSub. For convenience, we also recall the definition of the original OIP (see Definition 4.4 in Section 4.1).

Definition 6.3. Given that data $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$ is observed, let $C_{\mathcal{D}} \equiv C : \mathcal{M} \to \mathbb{R}$ be a selection criterion (with respect to maximization) with C-optimal model $S^* = f_C(\mathcal{P}) = \arg\max_{S \in \mathcal{M}} C(S) = \{j_1, \ldots, j_{s^*}\}$ of size $s^* = |S^*|$.

(a) The selection criterion C is said to fulfil the *ordered importance property (OIP)* for the

sample $\mathcal{D}$, if there exists a permutation $(k_1, \ldots, k_{s^*})$ of $(j_1, \ldots, j_{s^*})$ such that for each $i = 1, \ldots, s^* - 1$ it holds

$$k_i \in f_C(V) \quad \text{for all } V \subseteq \mathcal{P} \setminus N_{i-1} \text{ with } \{k_1, \ldots, k_i\} \subseteq V\,, \tag{6.6}$$

where

$$N_0 := \{j \in \mathcal{P};\ j \notin f_C(V) \text{ for all } V \subseteq \mathcal{P}\} \tag{6.7}$$

and

$$N_i := \{j \in \mathcal{P};\ j \notin f_C(V) \text{ for all } V \subseteq \mathcal{P} \setminus N_{i-1} \text{ with } \{k_1, \ldots, k_i\} \subseteq V\}\,. \tag{6.8}$$

(b) The selection criterion C is said to fulfil the *modified ordered importance property (MOIP)* for the sample $\mathcal{D}$, if there exists a permutation $(k_1, \ldots, k_{s^*})$ of $(j_1, \ldots, j_{s^*})$ such that for all $i = 1 \ldots, s^*$ and for all $V \subseteq \mathcal{P} \setminus \tilde{N}_{i-1}$ with $\{k_1, \ldots, k_i\} \subseteq V$ it holds

$$C(V \setminus \{k_i\}) < \max_{l \in V \setminus \{k_1, \ldots, k_i\}} C(V \setminus \{l\}) \quad \textbf{or} \quad f_C(V) = V\,, \tag{6.9}$$

where

$$\tilde{N}_0 := \{j \in \mathcal{P};\ C(V) < C(V \setminus \{j\}) \text{ for all } V \subseteq \mathcal{P} \text{ with } j \in V\} \tag{6.10}$$

and

$$\tilde{N}_i := \{j \in \mathcal{P};\ C(V) < C(V \setminus \{j\}) \text{ for all } V \subseteq \mathcal{P} \setminus \tilde{N}_{i-1} \text{ with } \{k_1, \ldots, k_i, j\} \subseteq V\}\,. \tag{6.11}$$

Theorem 6.2. *Suppose that the selection criterion C satisfies the MOIP for a given sample $\mathcal{D}$ with C-optimal set $S^* = f_C(\mathcal{P}) = \{j_1, \ldots, j_{s^*}\}$ of size $|S^*| = s^*$. Then we have:*

(a) The criterion C satisfies the OIP for the sample $\mathcal{D}$.

(b) BackAdaSub converges correctly against S^*, i.e. for all $j \in \mathcal{P}$ it holds

$$r_j^{(t)} \overset{a.s.}{\to} \begin{cases} 1 & \text{, if } j \in S^*, \\ 0 & \text{, if } j \notin S^*, \end{cases} \qquad \text{for } t \to \infty\,. \tag{6.12}$$

Proof. (a) Suppose that MOIP is satisfied with corresponding permutation $(k_1, \ldots, k_{s^*})$. We want to show that OIP is also satisfied with the same corresponding permutation $(k_1, \ldots, k_{s^*})$.

In particular, we shall prove that

$$\tilde{N}_i \subseteq N_i \quad , \text{ for } i = 0, 1, \ldots, s^* - 1 \,, \tag{6.13}$$

and that (6.6) holds for each $i = 1, \ldots, s^* - 1$.

First we show (6.13) for $i = 0$, i.e. $\tilde{N}_0 \subseteq N_0$: Let $j \in \tilde{N}_0$ and suppose that $j \notin N_0$. Then by equation (6.7) there exists a subset $V \subseteq \mathcal{P}$ with $j \in f_C(V)$. But then, by the definition of the map f_C, we have $C(f_C(V)) > C(f_C(V) \setminus \{j\})$, which contradicts the definition of $\tilde{N}_0$ in equation (6.10).

Suppose that (6.6) does not hold for $i = 1$, i.e. suppose that there exists a subset $V \subseteq \mathcal{P} \setminus N_0 \subseteq \mathcal{P} \setminus \tilde{N}_0$ with $k_1 \in V$ and $k_1 \notin f_C(V)$. Let $W := f_C(V) \cup \{k_1\} \subseteq V \setminus \tilde{N}_0$. Then, by the definition of the map f_C, we have $C(W \setminus \{k_1\}) > \max_{l \in W \setminus \{k_1\}} C(W \setminus \{l\})$ and $f_C(W) = f_C(V) \neq W$ which contradicts equation (6.9).

Now we show (6.13) for $i = 1$, i.e. $\tilde{N}_1 \subseteq N_1$: Let $j \in \tilde{N}_1$ and suppose that $j \notin N_1$. Then by equation (6.8) there exists a subset $V \subseteq \mathcal{P} \setminus N_0 \subseteq \mathcal{P} \setminus \tilde{N}_0$ with $\{k_1\} \subseteq V$ and $j \in f_C(V) \subseteq \mathcal{P} \setminus \tilde{N}_0$. By the definition of f_C we have $C(f_C(V)) > C(f_C(V) \setminus \{j\})$ which yields a contradiction to the definition of $\tilde{N}_1$ in equation (6.11), since we have already shown that $k_1 \in f_C(V)$.

Suppose that (6.6) does not hold for $i = 2$, i.e. suppose that there exists a subset $V \subseteq \mathcal{P} \setminus N_1 \subseteq \mathcal{P} \setminus \tilde{N}_1$ with $\{k_1, k_2\} \subseteq V$ and $k_2 \notin f_C(V)$. Let $W := f_C(V) \cup \{k_2\} \subseteq V \setminus \tilde{N}_1$. Then, by the definition of the map f_C, we have $C(W \setminus \{k_2\}) > \max_{l \in W \setminus \{k_1, k_2\}} C(W \setminus \{l\})$ and $f_C(W) = f_C(V) \neq W$ which contradicts equation (6.9), since $k_1 \in f_C(V)$ (as shown before) and therefore $k_1 \in W$.

For $i = 3, \ldots, s^* - 1$, (6.13) and (6.6) are proved analogously. Thus we conclude that OIP is satisfied.

(b) Suppose that MOIP is satisfied with corresponding permutation $(k_1, \ldots, k_{s^*})$. We want to show that the "corresponding OIP" (with the same permutation $(k_1, \ldots, k_{s^*})$) is satisfied if the map f_C is replaced by the map g_C, i.e. for each $i = 1, \ldots, s^*$ it holds

$$k_i \in g_C(V) \quad \text{for all } V \subseteq \mathcal{P} \setminus \overline{N}_{i-1} \text{ with } \{k_1, \ldots, k_i\} \subseteq V \,, \tag{6.14}$$

where

$$\overline{N}_0 := \{j \in \mathcal{P};\ j \notin g_C(V) \text{ for all } V \subseteq \mathcal{P}\} \tag{6.15}$$

and

$$\overline{N}_i := \{j \in \mathcal{P};\ j \notin g_C(V) \text{ for all } V \subseteq \mathcal{P} \setminus \overline{N}_{i-1} \text{ with } \{k_1, \ldots, k_i\} \subseteq V\}\,. \tag{6.16}$$

Furthermore, we have to show that $\overline{N}_{s^*} = \mathcal{P} \setminus S^*$. Then we can use an analogous argumentation as in Theorem 4.8 of Section 4.1 in order to conclude that BackAdaSub converges correctly in the sense that $r_j^{(t)} \overset{\text{a.s.}}{\to} 1$ for $j \in S^*$ and $r_j^{(t)} \overset{\text{a.s.}}{\to} 0$ for $j \in \mathcal{P} \setminus S^*$. We proceed as in part (a) by considering the sets $\overline{N}_i$ instead of N_i as well as the map g_C instead of f_C. By this, we can successively show that $\tilde{N}_i \subseteq \overline{N}_i$ for $i = 0, \ldots, i-1$ and that equation (6.14) holds for $i = 1, \ldots, s^*$. Since the proof is very similar to the proof of part (a), we describe only the first step in detail:
In order to show that $\tilde{N}_0 \subseteq \overline{N}_0$, let $j \in \tilde{N}_0$ and suppose that $j \notin \overline{N}_0$. Then there exists a subset $V \subseteq \mathcal{P}$ with $j \in g_C(V)$. But then, by the definition of the map g_C, we have $C(g_C(V)) > C(g_C(V) \setminus \{j\})$, which contradicts the definition of $\tilde{N}_0$ in equation (6.10). Again, using proof by contradiction, suppose that for $i = 1$ there exists a subset $V \subseteq \mathcal{P} \setminus \overline{N}_0 \subseteq \mathcal{P} \setminus \tilde{N}_0$ with $k_1 \in V$ and $k_1 \notin g_C(V)$. Then, by the definition of g_C, there exists a subset $W \subseteq V \subseteq \mathcal{P} \setminus \tilde{N}_0$ with $g_C(V) \subset W$ and $k_1 \in W$ such that $C(W \setminus \{k_1\}) > \max_{l \in W \setminus \{k_1\}} C(W \setminus \{l\})$ and $f_C(W) \neq W$. This contradicts equation (6.9). We continue in the same way in order to show that equation (6.14) holds for $i = 3, \ldots, s^*$.
Finally, we have to show that $\overline{N}_{s^*} = \mathcal{P} \setminus S^*$. For this, we need to prove that $g_C(V) = S^*$ for all $V \subseteq \mathcal{P} \setminus \overline{N}_{s^*-1}$ with $S^* \subseteq V$. Let $V \subseteq \mathcal{P} \setminus \overline{N}_{s^*-1}$ with $S^* \subseteq V$ and let $v := |V|$ denote the size of V. Furthermore, let $(V^{(v)}, V^{(v-1)}, \ldots, V^{(1)}, V^{(0)})$ be the sequence of subsets generated by BS when starting with the subset $V = V^{(v)}$ (note that the sequence of subsets is unique due to assumption (3.2)). In particular, we have $|V^{(i)}| = i$ for all $i = 0, \ldots, v$. Since equation (6.9) is satisfied for each $k_1, k_2, \ldots, k_{s^*}$, we have $V^{(s^*)} = S^*$. Now by the definition of $S^* = f_C(\mathcal{P})$ we have $C(S^*) > C(V^{(i)})$ for all $i \neq s^*$, and thus we conclude that $g_C(V) = S^*$. This completes the proof.

□

Part (b) of Theorem 6.2 shows that MOIP is a sufficient condition for the correct convergence of BackAdaSub, while part (a) shows that MOIP is a "stronger" condition than the original OIP. In particular, if MOIP is satisfied, then the original AdaSub algorithm

converges correctly as well (by Theorem 4.8 of Section 4.1).

We now try to explain the idea behind MOIP in some more detail. For simplicity, we first disregard the sets $\tilde{N}_i$ in our explanation. Then MOIP requires that there exists an "importance" reordering $(k_1, \ldots, k_{s^*})$ of the variables inside the C-optimal model $S^* = \{j_1, \ldots, j_{s^*}\}$ with the following properties: If one considers any subset $V \subseteq \mathcal{P}$ with $k_1 \in V$, then either the subset V cannot be further improved by removing any set of variables from it (i.e. $f_C(V) = V$) or, if it can be improved, then the variable X_{k_1} is not the "worst" variable inside the model V, i.e. there is at least one other variable X_l with $l \in V$, that yields a larger value of the criterion C when it is deleted: $C(V \setminus \{k_1\}) < \max_{l \in V \setminus \{k_1\}} C(V \setminus \{l\})$. Therefore, X_{k_1} might be viewed as the "most important" variable.

Now, for the "second most important" variable X_{k_2}, we require a similar property: If one considers any subset $V \subseteq \mathcal{P}$ with $k_2 \in V$ **and** $k_1 \in V$, then either the subset V cannot be further improved by removing any set of variables from it (i.e. $f_C(V) = V$) or, if it can be improved, then the variable X_{k_2} is not the "worst" variable inside the model V. We can continue in the same way for $k_3, \ldots, k_{s^*}$.

It turns out that we can weaken the explained assumptions by introducing the sets $\tilde{N}_i$ for $i = 0, 1, \ldots, s^* - 1$. The set $\tilde{N}_0$ contains all the variables X_j with the following property: Whenever X_j is considered in any model V, then the deletion of that variable improves the criterion, i.e. $C(V) < C(V \setminus \{j\})$. Clearly, such variables will never be selected by BS when starting from any subset V, so we do not have to consider them further. Similarly, the set $\tilde{N}_1$ contains all the variables X_j with the following property: Whenever X_j is considered in any model V which includes also the "most important variable" X_{k_1} (but no variables from the set $\tilde{N}_0$), then the deletion of X_j improves the criterion. We can continue in the same way with the interpretation of the sets $\tilde{N}_2, \ldots, \tilde{N}_{s^*-1}$.

We have shown that if MOIP holds, then BackAdaSub converges correctly against the C-optimal model S^*. But what can we say if this assumption is not satisfied? Then the thresholded model selected by BackAdaSub (for a sufficiently large number of iterations) will contain at least those variables in S^* which are included in a maximal learning path in the sense of MOIP (compare also the similar result for AdaSub in Theorem 4.12). The precise statement can be found in the next theorem.

Theorem 6.3. *Let $S^* = \{j_1, \ldots, j_{s^*}\}$ be the C-optimal model of size $|S^*| = s^*$ and let $D = \{l_1, \ldots, l_d\} \subseteq S^*$ be of maximal cardinality $|D| = d$ such that there exists a permutation $(k_1, \ldots, k_d)$ of $(l_1, \ldots, l_d)$ such that for all $i = 1, \ldots, d$ and for all $V \subseteq \mathcal{P} \setminus \tilde{N}_{i-1}$ with $\{k_1, \ldots, k_i\} \subseteq V$ we have*

$$C(V \setminus \{k_i\}) < \max_{l \in V \setminus \{k_1, \ldots, k_i\}} C(V \setminus \{l\}) \quad \textbf{\textit{or}} \quad f_C(V) = V\,, \tag{6.17}$$

where the sets $\tilde{N}_0, \ldots, \tilde{N}_d$ are defined as in Definition 6.3. In particular, we have

$$\tilde{N}_d = \{j \in \mathcal{P};\ C(V) < C(V \setminus \{j\}) \text{ for all } V \subseteq \mathcal{P} \setminus \tilde{N}_{d-1} \text{ with } \{k_1, \ldots, k_d, j\} \subseteq V\}\,. \tag{6.18}$$

Then, when using BackAdaSub we obtain: For all $j \in D$ we have $r_j^{(t)} \overset{a.s.}{\to} 1$, $t \to \infty$ and for all $j \in \tilde{N}_d$ we have $r_j^{(t)} \overset{a.s.}{\to} 0$, $t \to \infty$.

Proof. The proof is along the lines of the proof of Theorem 6.2 (b), using the (partial) permutation $(k_1, \ldots, k_d)$ of variables in $D \subseteq S^*$ instead of the (full) permutation $(k_1, \ldots, k_{s^*})$ of all variables in S^*. □

6.2.2. Empirical analysis of limiting properties

In this section we want to confirm the obtained theoretical results in a simulation study where the main target is to compare the performance of the original AdaSub algorithm with the proposed BackAdaSub method. For this, we consider simulated data from normal linear models, so that it is computationally feasible to apply the original AdaSub algorithm.

We make use of the Simulation Setup 5.1 described in Section 5.1, using a Toeplitz-correlation structure for the design matrix $\boldsymbol{X}$ with constant $c \in (-1, 1)$ (which controls the correlation between the explanatory variables). For a given choice of the constant c, the sample size n is increased from 40 to 200 in steps of size 20 and for each value of n we simulate 100 different datasets according to Simulation Setup 5.1. For each simulated dataset, we compute the thresholded model $\hat{S}_\rho$ with threshold $\rho = 0.9$ selected by AdaSub, BackAdaSub and FoAdaSub, respectively. In all algorithms we consider the choices $q = 5$, $K = n$ and $T = 3000$. Furthermore, in the comparisons we include the performance of the models selected by usual Forward Stepwise Selection (FS2) and usual Backward Stepwise Selection (BS).

We first consider a relatively low-dimensional scenario with $p = 30$ explanatory variables, so that we can compute the C-optimal model $S^* = f_C(\mathcal{P})$ by making use of the "leaps and bounds" algorithm (Lumley and Miller, 2009). Figure 6.1 summarizes the results for the low-dimensional scenario when using the (negative) BIC as the selection criterion C. Furthermore, Figure 6.2 depicts the results of a moderately high-dimensional scenario with $p = 100$ explanatory variables when using the (negative) $\text{EBIC}_{0.5}$ as the selection criterion C. In both scenarios we consider a setting with large correlations between the covariates (Toeplitz-correlation structure with $c = 0.9$), in order to generate sufficiently challenging variable selection problems so that the differences between the methods are more pronounced.

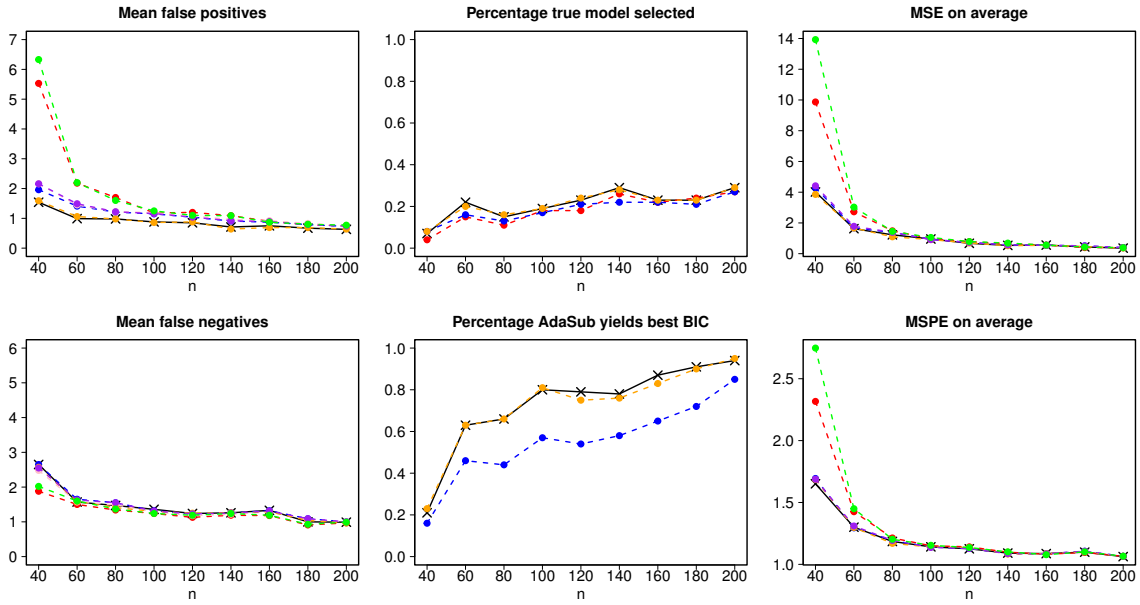


Figure 6.1.: Low-dimensional linear model examples ($p = 30$) with Toeplitz-correlation structure ($c = 0.9$): Comparison of $\hat{S}_{0.9}$ from AdaSub (black), $\hat{S}_{0.9}$ from BackAdaSub (orange), $\hat{S}_{0.9}$ from FoAdaSub (blue), FS2 model (purple), BIC-optimal model S^* (red) and BS model (green) in terms of mean number of false positives/ false negatives, percentage of selecting the true model S_0, percentage of agreement between thresholded AdaSub models and S^*, Mean Squared Error (MSE) and Mean Squared Prediction Error (MSPE) on independent test set with sample size 100.

In the low-dimensional scenario, the BIC-optimal models S^* and even more the models selected by BS tend to include many false positives for small sample sizes. On the other hand, the thresholded models $\hat{S}_{0.9}$ selected by AdaSub and BackAdaSub are usually sparser and often reduce the number of falsely selected variables in a situation where the BIC is too liberal. Both in the low- and high-dimensional scenario, the models selected by AdaSub and BackAdaSub show similar statistical performance, with the tendency that BackAdaSub favours

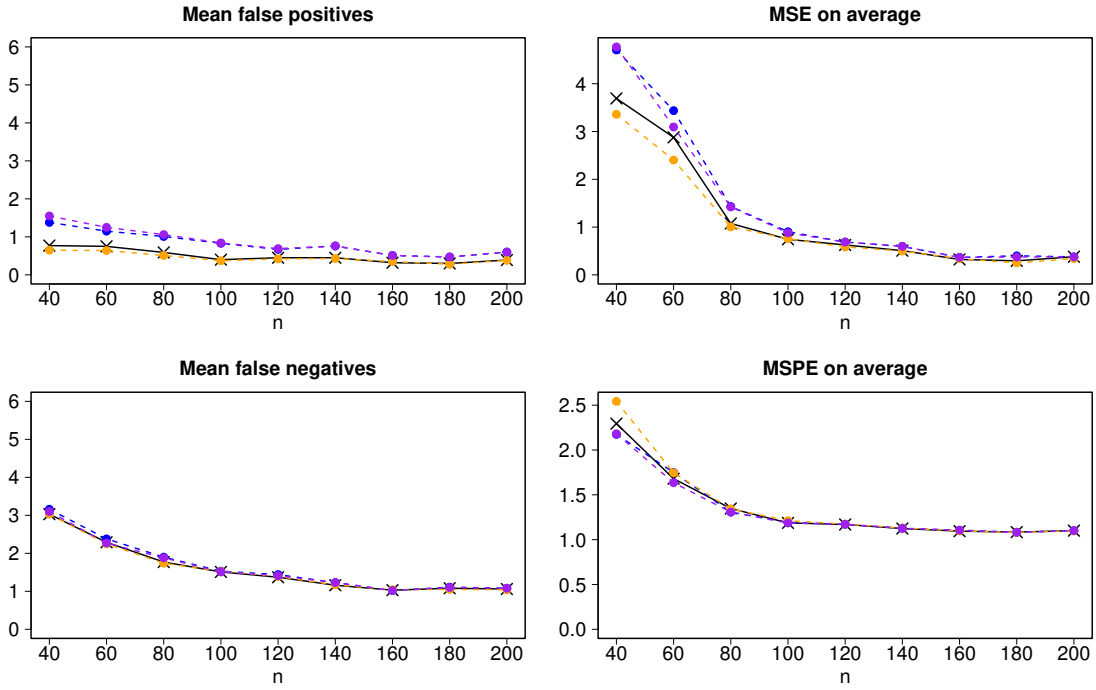


Figure 6.2.: High-dimensional linear model examples ($p = 100$) with Toeplitz-correlation structure ($c = 0.9$): Comparison of $\hat{S}_{0.9}$ from AdaSub (black), $\hat{S}_{0.9}$ from BackAdaSub (orange), $\hat{S}_{0.9}$ from FoAdaSub (blue) and FS2 model (purple) in terms of mean number of false positives/ false negatives, percentage of selecting the true model S_0, percentage of agreement between AdaSub models and S^*, Mean Squared Error (MSE) and Mean Squared Prediction Error (MSPE) on independent test set with sample size 100.

slightly sparser models with even less numbers of false positives in the high-dimensional scenario (at the prize of potentially loosing some predictive power). On the other hand, the models selected by FoAdaSub and usual Forward Stepwise Selection (FS2) show very similar performance and tend to agree as the sample size n increases. The reason that there are some few situations where the two models selected by FoAdaSub and FS2 do not agree is that in such situations FoAdaSub has not yet "converged" against the model selected by FS2 after $T = 3000$ iterations. This observation does not contradict the convergence result in Theorem 6.1. If the number of iterations T of AdaSub is increased in these examples, we observe that FoAdaSub and FS2 tend to agree, although the "convergence" can be quite slow in some situations. Recall that due to the stochastic nature of the Adaptive Subspace methods it is intrinsically difficult to know in advance how many iterations T are sufficient for the "convergence" of the algorithms. Finally, note that when the sample size n increases, all considered methods tend to perform more similarly.

All in all, the empirical behaviour of FoAdaSub and BackAdaSub in the considered simu-

lated data examples supports the theoretical results obtained in the previous subsection. In particular, AdaSub and BackAdaSub perform similarly, so that BackAdaSub may be used as an efficient surrogate algorithm for the original AdaSub method.

6.3. Simulation study for high-dimensional logistic regression

In this section we compare the performance of the proposed BackAdaSub method to other prominent variable selection methods in the challenging scenario of high-dimensional logistic regression models with a binary response variable. For this, we will make use of the following simulation setup.

Simulation Setup 6.1. We consider the same setting as in Simulation Setup 5.1 described in Section 5.1, but with the modification that the "true" underlying regression coefficients $\beta_{0,j} \overset{\text{ind.}}{\sim} \mathcal{U}(-5,5)$, $j \in S_0$, are simulated from the uniform distribution on $[-5,5]$ and that the response vector $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^T$ is simulated according to a logistic regression model via $Y_i \overset{\text{ind.}}{\sim} \text{Bernoulli}(\pi_i)$, where

$$\pi_i = \frac{\exp(\boldsymbol{X}_{i,*}\boldsymbol{\beta}_0)}{\exp(1 + \boldsymbol{X}_{i,*}\boldsymbol{\beta}_0)} \tag{6.19}$$

is the success probability corresponding to observation $i \in \{1, \ldots, n\}$.

In order to compare different classification methods with respect to their variable selection properties, we make use of the same performance measures as described in Notation 5.1 of Section 5.1 in the framework of linear regression models. However, for measuring the predictive accuracy of the final classifier, we compute the well-known *Area Under the Receiver Operating Characteristic (ROC) Curve* (*AUC*, Hanley and McNeil, 1982). As the name suggests, the AUC is the area under the ROC curve, which plots the true positive rate (sensitivity) against the false positive rate ($1 -$ specificity) for different thresholds of a given classifier. If for example $y_i = 0$ has the interpretation that subject i is healthy, while $y_i = 1$ means that subject i has a particular disease, then the AUC can be interpreted as "the probability that a randomly chosen diseased subject is (correctly) rated or ranked with greater suspicion than a randomly chosen non-diseased subject" (Hanley and McNeil, 1982, p. 30). The AUC takes values in the interval $[0,1]$, with the interpretation that larger AUC values indicate better predictive performance of the classifiers and that an AUC value of 0.5 corresponds to "random guessing".

In the following, we will use the (negative) EBIC_γ as the selection criterion C for different regularization constants $\gamma \in [0, 1]$. For the computation of $\text{EBIC}_\gamma(S)$ for models $S \in \mathcal{M}$ we make use of a very fast C++ implementation for ML-estimation in logistic regression models via a limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm, which is available in the R-package `RcppNumerical` (Qiu et al., 2016). Furthermore, after the final model has been selected, in order to stabilize the resulting estimates in the possible case of (quasi-)complete separation (see e.g. Albert and Anderson, 1984), we employ the "weakly informative default prior distribution" for logistic regression models as recommended by Gelman et al. (2008), using an independent Cauchy prior for the regression coefficients (see Gelman et al., 2008, Section 2 for details). For this, we make use of the R-function `bayesglm` (with default choices), which is available in the `arm` package (Gelman and Su, 2016). Note that due to the use of weakly informative priors, the resulting Bayesian estimates are generally very close to the MLEs, with the additional benefit of providing robust estimates of all regression coefficients even in the case of (quasi-)complete separation.

6.3.1. Simulations with fixed number of variables

We first consider a moderately high-dimensional setting where the number of explanatory variables $p = 100$ is fixed and the sample size n is increased from 60 to 140 in steps of size 20. In this setting we make use of the EBIC_γ with constant $\gamma = 0.5$ as the criterion C. For each value of n we simulate 100 datasets according to the Simulation Setup 6.1. For each dataset we apply BackAdaSub, Forward Stepwise Selection (FS2), the Lasso, the Adaptive Lasso, the SCAD and Stability Selection. In BackAdaSub we set $q = 5$, $K = n$ and run the algorithm for $T = 1000$ iterations; we report the performance of the thresholded model $\hat{S}_\rho$ with threshold $\rho = 0.9$ and the best model $\hat{S}_\text{b}$ found by BackAdaSub. The other methods are applied with the same specifications as described in the beginning of Section 5.1.2 (for the linear regression setting).

Figure 6.3 depicts the results for a Toeplitz correlation structure with $c = 0$ (i.e. independent predictors), while Figure 6.4 shows the results for a Toeplitz correlation structure with large correlation $c = 0.9$. In both scenarios, the SCAD tends to select too large models with many false positives, while Stability Selection with the Lasso tends to select too small models with many false negatives (in order to keep the expected number of false positives

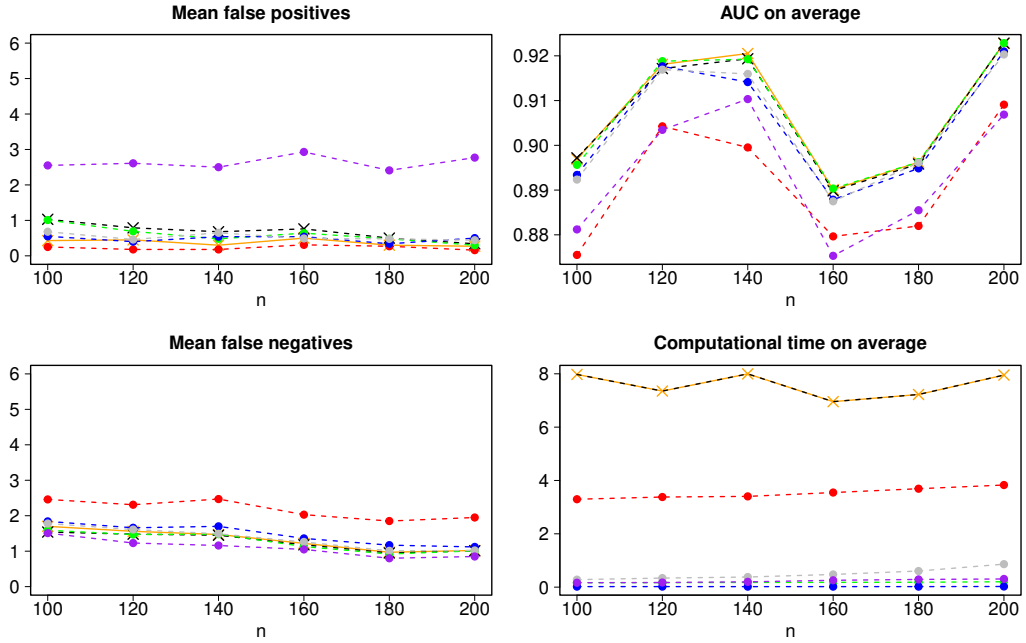


Figure 6.3.: High-dimensional example with fixed $p = 100$ and independent explanatory variables ($c = 0$): Comparison of $\hat{S}_{0.9}$ (orange) and $\hat{S}_{\mathrm{b}}$ (black) from BackAdaSub, as well as FS2 (green), Lasso (blue), Adaptive Lasso (gray), SCAD (purple) and Stability Selection (red) in terms of mean number of false positives/ false negatives, mean area under the ROC curve (AUC) for independent test set of sample size 100 and mean computational time (in s).

below 1). In the independent predictor scenario, the Lasso and the Adaptive Lasso perform only slightly worse than the thresholded model from BackAdaSub with respect to variable selection and prediction. However, in the high correlation scenario, the Lasso and the Adaptive Lasso select on average significantly larger numbers of false positives and false negatives in comparison to the thresholded model from BackAdaSub.

In the independence scenario, Forward Stepwise Selection (FS2), the thresholded model as well as the "best" model from BackAdaSub show the best predictive performance in terms of mean AUC (closely followed by the Lasso and the Adaptive Lasso), although FS2 and the "best" model from BackAdaSub select on average larger models with slightly more false positives and slightly less false negatives than the thresholded model. The situation is significantly different in the high correlation case: Here, the thresholded model shows arguably the best variable selection performance, tightly controlling the number of false positives without a substantial increase in the mean number of false negatives. The SCAD, the Adaptive Lasso and the "best" model from BackAdaSub do lead to smaller mean numbers of false negatives, but only at the prize of increasing the mean number of false positives.

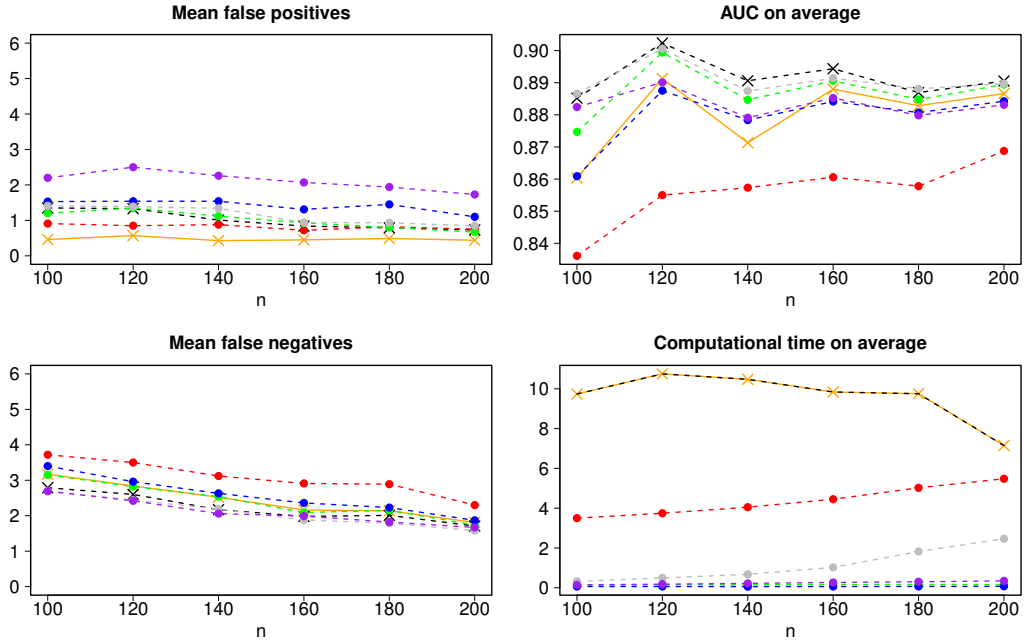


Figure 6.4.: High-dimensional logistic regression example with fixed $p = 100$ and Toeplitz-correlation structure ($c = 0.9$): Comparison of $\hat{S}_{0.9}$ (orange) and $\hat{S}_{\mathrm{b}}$ (black) from BackAdaSub, as well as FS2 (green), Lasso (blue), Adaptive Lasso (gray), SCAD (purple) and Stability Selection (red) in terms of mean number of false positives/ false negatives, mean area under the ROC curve (AUC) for independent test set of sample size 100 and mean computational time (in s).

Regarding the predictive performance (in terms of AUC) for the high correlation scenario, the "best" model from BackAdaSub seems to yield the best results, closely followed by the Adaptive Lasso. Interestingly, FS2 also shows a very good predictive performance. The thresholded model from BackAdaSub performs significantly worse than the "best" model from BackAdaSub as well as FS2. This is due to the fact that "thresholding" the output from BackAdaSub leads to smaller models with less false positives, which is not beneficial for prediction if there exists large positive correlation between the predictors.

In order to assess the variability of the results, Figure 6.5 exemplarily depicts boxplots of the performance measures for the specific case of $n = 140$ (with 100 replicates) in the high correlation scenario. It is apparent that the thresholded model from BackAdaSub selects only a very small number of false positives for all simulated datasets, while the variability in the numbers of false negatives is comparable to the other methods (except for the SCAD). The SCAD tends to select larger models than the other methods with significantly larger numbers of false positives. While the AUC values for the "best" model from BackAdaSub are on average (slightly) larger than for the other methods, the AUC values for the Adaptive

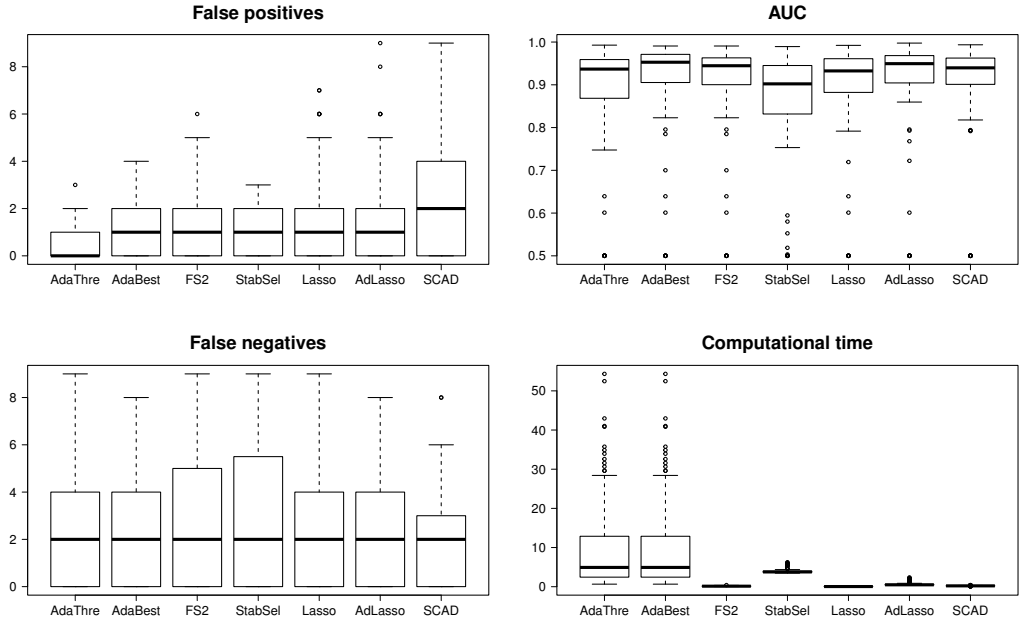


Figure 6.5.: High-dimensional logistic regression example with fixed $p = 100$ and Toeplitz-correlation structure ($c = 0.9$): Boxplots for the case $n = 140$, for $\hat{S}_{0.9}$ (AdaThre) and $\hat{S}_{\mathrm{b}}$ (AdaBest) from BackAdaSub, as well as for FS2, Stability Selection (StabSel), Lasso, Adaptive Lasso (AdLasso) and SCAD.

Lasso show (slightly) less variability over the simulated datasets in the considered setting.

We have also included a comparison of the average computation time for the considered methods (note that the models $\hat{S}_{0.9}$ and $\hat{S}_{\mathrm{b}}$ are derived from a single run of BackAdaSub, so that the computational times for the BackAdaSub models agree). FS2, the SCAD, the Lasso and the Adaptive Lasso are very efficient and often take less than a second to be applied on a particular dataset. Stability Selection and BackAdaSub are computationally more intensive, but still take only a couple of seconds to be applied for most datasets (with BackAdaSub requiring more computational time for those datasets with larger underlying "true" model). Since BackAdaSub shows favourable statistical properties in comparison to the other methods (and in particular in comparison to Stability Selection), we are convinced that the moderate additional computation time for BackAdaSub is worth it.

6.3.2. Simulations with increasing number of variables

We now consider a high-dimensional asymptotic setting where the number of explanatory variables p increases with the sample size n. In particular, we consider $p = 10 \times n$, where n is increased from 60 to 140 in steps of size 20 (this means that p is increased from 600 to 1400

in steps of size 200). For each value of n we simulate 50 datasets according to the Simulation Setup 6.1. Here, we make use of the (negative) EBIC$_\gamma$ with constant $\gamma = 0.6$ as the selection criterion C, which yields a variable selection consistent criterion under suitable conditions in the given high-dimensional asymptotic setting (see Chen and Chen, 2012; note that $p = \mathcal{O}(n^k)$ with $k = 1$ and $\gamma > 1 - \frac{1}{2k}$). We consider the same methods as described above, but due to the larger space of models we increase the number of iterations in BackAdaSub to $T = 3000$.

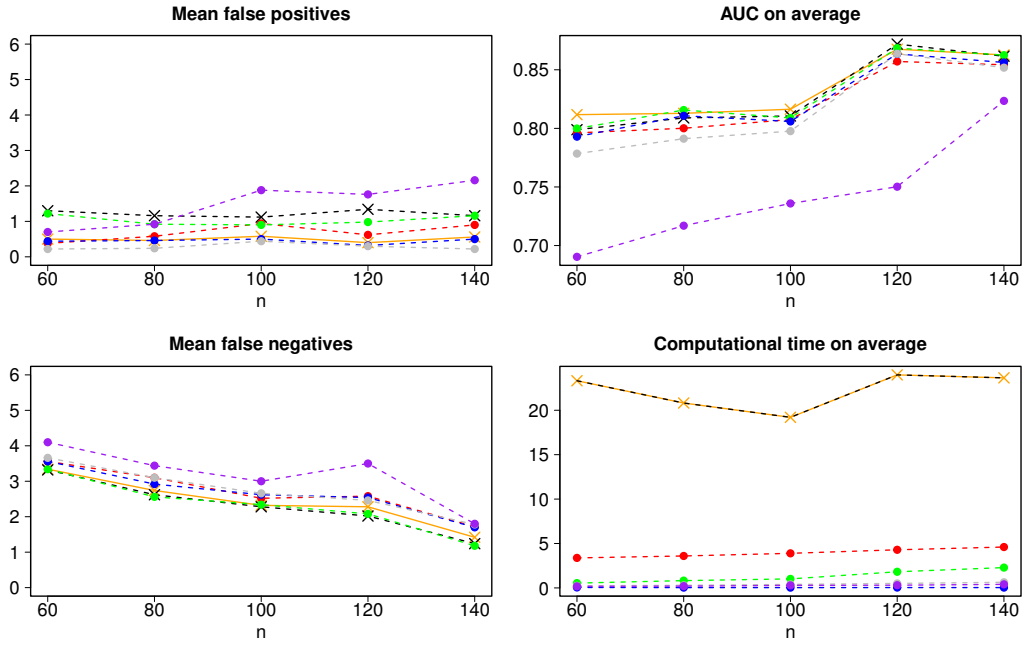


Figure 6.6.: High-dimensional logistic regression example with increasing p ($p = 10 \times n$) and independent explanatory variables ($c = 0$): Comparison of $\hat{S}_{0.9}$ (orange) and $\hat{S}_\text{b}$ (black) from BackAdaSub, as well as FS2 (green), Lasso (blue), Adaptive Lasso (gray), SCAD (purple) and Stability Selection (red) in terms of mean number of false positives/ false negatives, mean area under the ROC curve (AUC) for independent test set of sample size 100 and mean computational time (in s).

In this high-dimensional scenario, Figure 6.6 depicts the results for a Toeplitz correlation structure with $c = 0$ (i.e. independent predictors), while Figure 6.7 shows the results for a Toeplitz correlation structure with large correlation $c = 0.9$. The observations in these high-dimensional scenarios are similar to the ones described for the previous setting (compare Figure 6.3 and Figure 6.4). The thresholded models from BackAdaSub show the best variable selection performance, since they tightly control the number of false positives while not paying a lot in terms of false negatives; on the other hand, especially in the high correlation setting, the "best" sampled models from BackAdaSub show the best predictive performance

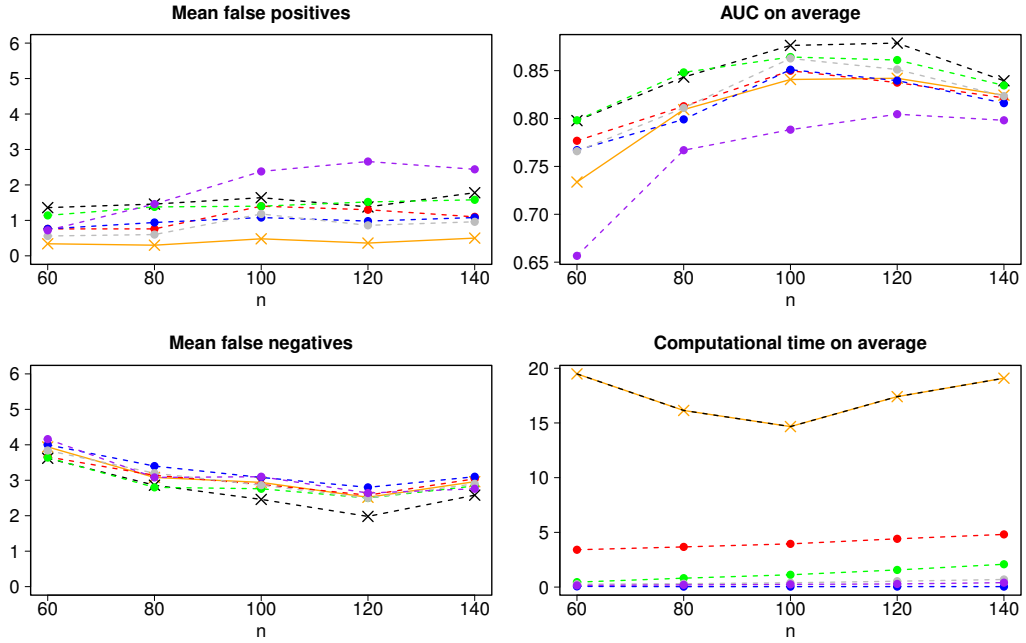


Figure 6.7.: High-dimensional logistic regression example with increasing p ($p = 10 \times n$) and Toeplitz-correlation structure ($c = 0.9$): Comparison of $\hat{S}_{0.9}$ (orange) and $\hat{S}_{\mathrm{b}}$ (black) from Back-AdaSub, as well as FS2 (green), Lasso (blue), Adaptive Lasso (gray), SCAD (purple) and Stability Selection (red) in terms of mean number of false positives/ false negatives, mean area under the ROC curve (AUC) for independent test set of sample size 100 and mean computational time (in s).

(yielding the largest values of AUC). This general tendency is also confirmed in further simulations with different correlation structures (like global or block correlation structures, compare Simulation Setup 5.1 of Section 5.1). Due to brevity, we do not report all the additional simulations here.

6.3.3. Simulation based on real design matrix

With regard to the analysis of real data examples in the subsequent section, we want to conclude this section with results from a simulation setup based on a real design matrix from the field of genomics. In order to be able to compare different methods with respect to their ability in recovering the underlying "truth", we do not make use of the observed responses from the real dataset, but instead generate the responses according to a given logistic regression model where the underlying sparse vector of regression coefficients $\boldsymbol{\beta}_0 \in \mathbb{R}^p$ is assumed to be known. Similar simulation setups based on real design matrices have also been used in Meinshausen and Bühlmann (2010) for the investigation of the performance of Stability Selection.

Simulation Setup 6.2. We consider the same setting as in Simulation Setup 6.1, but with the modification that we make use of a given real design matrix $\boldsymbol{X}^{\text{real}} \in \mathbb{R}^{n \times p}$ and that the sparsity level s_0 is increased in steps from 0 to 6. For each sparsity level s_0 we generate 50 datasets in the following way: We randomly draw a reduced index set of explanatory variables $\mathcal{P}' \subseteq \mathcal{P} = \{1, \ldots, p\}$ of size $|\mathcal{P}'| = p'$, where p' is specified in advance. We randomly draw the true active set $S_0 \subseteq \mathcal{P}'$ of size $|S_0| = s_0$. Then the non-zero regression coefficients $\beta_{0,j} \overset{\text{ind.}}{\sim} \mathcal{U}(-5, 5)$, $j \in S_0$, of $\boldsymbol{\beta}_0 \in \mathbb{R}^p$ are again simulated from the uniform distribution on $[-5, 5]$ and the response vector $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^T$ is simulated according to a logistic regression model via $Y_i \overset{\text{ind.}}{\sim} \text{Bernoulli}(\pi_i)$ with

$$\pi_i = \frac{\exp(\boldsymbol{X}^{\text{real}}_{i,*} \boldsymbol{\beta}_0)}{\exp(1 + \boldsymbol{X}^{\text{real}}_{i,*} \boldsymbol{\beta}_0)} \,. \tag{6.20}$$

As the real design matrix $\boldsymbol{X}^{\text{real}}$ in Simulation Setup 6.2 we consider the preprocessed leukemia data from Golub et al. (1999), where the design matrix consists of gene expression measurements of $p = 3571$ genes for $n = 72$ patients (see Section 6.4.2 for a detailed description and analysis of this particular dataset). For convenience, we have standardized each column of $\boldsymbol{X}^{\text{real}}$ in order to have mean zero and variance one. Furthermore, in the repeated simulations we set $p' = 500$, so that each simulated dataset actually consists of $p' = 500$ randomly chosen genes out of the $p = 3571$ total number of genes. The employed selection criterion C is the (negative) EBIC_γ with constant $\gamma = 0.6$ and we consider the same variable selection methods as described above; in particular we set $q = 5$, $K = n$ and $T = 3000$ in BackAdaSub.

Figure 6.8 depicts the results for the Simulation Setup 6.2 using the leukemia dataset. It does not come as a surprise that the mean number of false negatives increases with the sparsity level s_0 for all considered methods. Furthermore, note that for $s_0 = 0$, the true underlying model is the "null" model $S_0 = \emptyset$ and the AUC values are approximately 0.5 for all methods, meaning that each considered classifier is "as good as random guessing". For sparsity levels $s_0 \geq 1$, the "best" model found by BackAdaSub shows again the best predictive performance with the largest AUC values, at the prize of a relatively large number of mean false positives. On the other hand, the thresholded model $\hat{S}_\rho$ with $\rho = 0.9$ from BackAdaSub yields favourable variable selection properties: The mean number of false positives is controlled for all considered sparsity levels, while the mean number of false negatives is

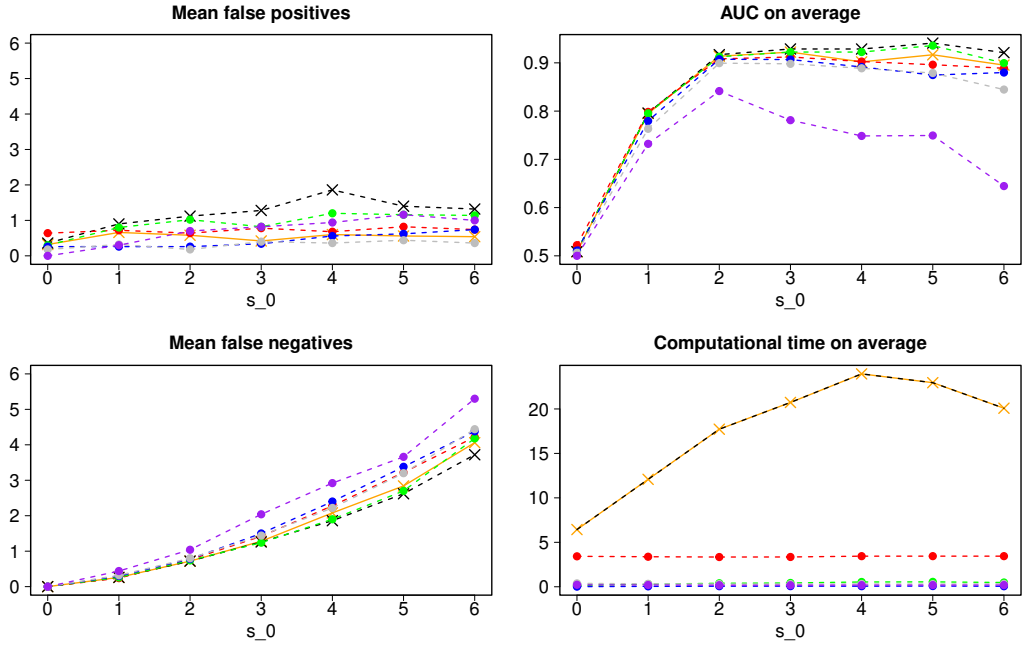


Figure 6.8.: High-dimensional logistic regression example based on leukemia dataset with increasing sparsity level $s_0 = 0, \ldots, 6$: Comparison of $\hat{S}_{0.9}$ (orange) and $\hat{S}_{\mathrm{b}}$ (black) from BackAdaSub, as well as FS2 (green), Lasso (blue), Adaptive Lasso (gray), SCAD (purple) and Stability Selection (red) in terms of mean number of false positives/ false negatives, mean area under the ROC curve (AUC) for independent test set of sample size 72 (using the same design matrix) and mean computational time (in s).

relatively small in comparison to the other methods. In particular, the thresholded model uniformly outperforms Stability Selection for all sparsity levels. Interestingly, the Lasso and the Adaptive Lasso perform relatively well for small sparsity levels in the considered setting, with small mean numbers of false positives. However, when the sparsity level increases, the Lasso and the Adaptive Lasso tend to miss more truly important variables than the models selected by BackAdaSub.

The computation time for BackAdaSub increases with the sparsity level s_0, since the sampled sub-problems $V^{(t)}$ tend to get larger (which is also due to empirical correlations between "noise" and "signal" variables) and thus the computation of the models $h_C(V^{(t)})$ selected by Backward Selection takes more time. Nevertheless, as mentioned above, we are convinced that the extra computational effort of BackAdaSub in comparison to the other methods is well spent time.

On a final note, the results from the controlled simulation study in Figure 6.8 already indicate that the variable selection problem for the given real dataset is really challenging and that we cannot expect to exactly recover the "truly important" genes. In the subsequent

section we confirm this observation by applying BackAdaSub and its competitors on the original dataset of Golub et al. (1999) (with the large number of $p = 3571$ explanatory variables), as well as on a further real dataset described in Alon et al. (1999). The results of a simulation study based on the real design matrix from Alon et al. (1999) are very similar to the one presented for the dataset of Golub et al. (1999), so we omit the detailed results here.

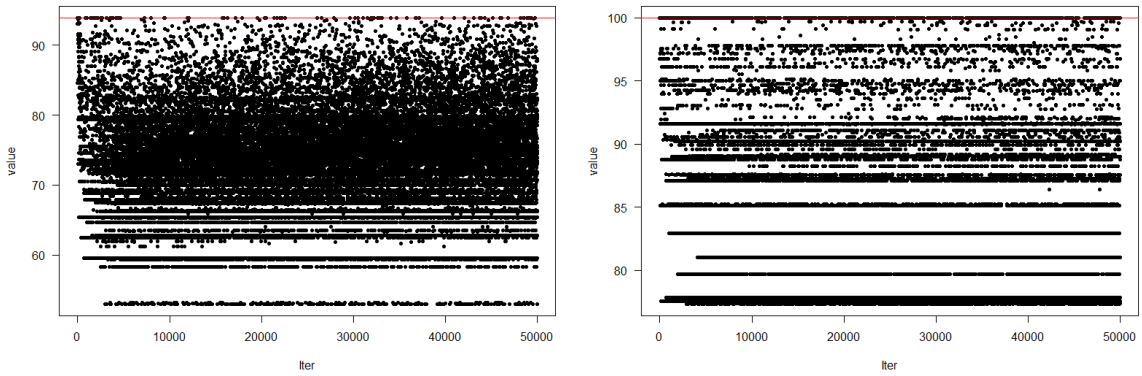
6.4. Real data examples

In this section we illustrate the application of BackAdaSub on real datasets from the field of genomics. We want to emphasize that a lot of applied work in genomics has focused on computational efficient screening methods (or filter methods) which for example consider only the ranking of the marginal correlations between the response variable and each gene (compare e.g. Golub et al., 1999, as well as Section 2.6 of this thesis). These methods often do not take proper account of the complex correlation structure among the genes and therefore more sophisticated multivariate methods might be preferred. Since in many practical situations, as in the considered real data examples below, the response variable is categorical or even binary (e.g. the disease status of a patient), it is natural to employ a logistic regression model.

Here we want to consider two classical datasets from genomics: Section 6.4.1 contains the analysis of the colon cancer dataset introduced in Alon et al. (1999), while Section 6.4.2 addresses the leukemia dataset described in Golub et al. (1999).

6.4.1. Colon cancer dataset

The first real data example we consider is the colon cancer dataset of Alon et al. (1999), which is publicly available at `http://microarray.princeton.edu/oncology/affydata/index.html` and can be conveniently loaded from the R-package `cepp` (Dayal, 2016). The data consists of gene expression measurements of $p = 2000$ human genes for $n = 62$ colon tissues, of which 40 are tumorous and 22 are normal tissues. The gene expression data is first preprocessed in the following way: The logarithm of basis 10 is applied to each measured expression level and then each row of the design matrix (corresponding to each tissue) is standardized to have mean zero and variance one.



(a) $\gamma = 0.6$ (b) $\gamma = 1$

Figure 6.9.: BackAdaSub for colon cancer data. Plots of the evolution of $\text{EBIC}_\gamma(S^{(t)})$ along the iterations t for (a) $\gamma = 0.6$ and (b) $\gamma = 1$. The red lines indicate the EBIC_γ-values of the thresholded model $\hat{S}_{0.9}$.

We apply BackAdaSub on the preprocessed data using $q = 5$, $K = n$ and $T = 50,0000$. We consider the (negative) EBIC_γ as the selection criterion C with two different choices of the constant $\gamma \in \{0.6, 1\}$. The evolution of the values $\text{EBIC}_\gamma(S^{(t)})$, for $\gamma \in \{0.6, 1\}$, along the iterations (t) is given in Figure 6.9.

For $\gamma = 1$, no genes are selected to be in the thresholded model $\hat{S}_\rho$ with threshold $\rho = 0.9$ of BackAdaSub (computation time approximately 15 minutes), which indicates that no variable (gene) is "stable" in the sense of MOIP (compare Definition 6.3 and Theorem 6.3 of Section 6.2). However, two genes are selected to be in the thresholded model $\hat{S}_{0.5}$, namely H06524 and D14812. These genes are also selected by the Bayesian variable selection method considered in Ai-Jun and Xin-Yuan (2009), having the third and fifth largest posterior marginal inclusion probabilities in their Bayesian probit regression analysis. For $\gamma = 1$, the "best" model $\hat{S}_\text{b}$ found by BackAdaSub consists of the genes D14812, T51849, X86693. Note that D14812 is both included in the models $\hat{S}_{0.5}$ and $\hat{S}_\text{b}$, while T51849 and X86693 are not. Interestingly, T51849 and X86693 are also not among the top 18 genes selected by the Bayesian approach of Ai-Jun and Xin-Yuan (2009), which underpins that these genes are not "stable" and are likely to be just incidental findings.

For $\gamma = 0.6$, no genes are selected to be in the thresholded models $\hat{S}_\rho$ with thresholds $\rho = 0.9$ and $\rho = 0.5$ of BackAdaSub (computation time approximately 1 hour). Again, this indicates that no genes are "stable" in the sense of MOIP. For $\gamma = 0.6$, the "best" model $\hat{S}_\text{b}$ of BackAdaSub consists of the genes H06524, H63354 and H64807. Note that the gene

H06524 is also selected by the thresholded model $\hat{S}_{0.5}$ for $\gamma = 1$.

We furthermore report the models selected by other variable selection methods: For $\gamma = 1$ and $\gamma = 0.6$, the Lasso as well as the Adaptive Lasso select the single gene J02854. For $\gamma = 1$, Forward Stepwise Selection yields a model with the single gene Z50753, while for $\gamma = 0.6$ it selects the two genes Z50753 and H22579. Stability Selection with error control of one false positive selects the single gene Z50753. Note that in the Bayesian analysis of Ai-Jun and Xin-Yuan (2009), the genes Z50753 and J02854 have respectively the first and seventh largest posterior marginal inclusion probability, while H22579 is not among the top scoring genes.

The variety of different models selected by the different methods (with no gene selected uniformly by all methods) supports the finding of BackAdaSub that there is no truly "stable" gene on the basis of the given data. Although this might look like a negative result on the first sight, we think that it provides very valuable information: Identifying significant genes with high certainty is simply not possible based on the given data of very small sample size ($n = 62$) and, if feasible, one should aim at collecting a larger sample. Note that this finding does not address the (possibly good) predictive abilities of the selected models, but instead the identification of the underlying data generating mechanism.

Finally, we briefly comment on the predictive performance of BackAdaSub for the considered dataset. Note that a fair comparison of models of different sizes requires an independent test set (which is not available unless the already small training sample size of $n = 62$ is further reduced) or an external cross-validation, in which each method has to be applied separately for each subsample of the original data (yielding possibly different models). For a detailed discussion of issues related to the "selection bias" of variable selection methods we refer to Ambroise and McLachlan (2002).

Table 6.1 shows the results from an external leave-one-out-cross-validation (LOOCV) for the "best" model from BackAdaSub when using the EBIC_γ with $\gamma = 1$ and $\gamma = 0.6$. Here, for computational convenience, the number of iterations for each of the $n = 62$ runs of BackAdaSub is set to the relatively small value of $T = 2000$. We focus on the "best" model from BackAdaSub, since it yielded favourable predictive performance in the simulation studies of Section 6.3. Furthermore, for comparison reasons, we also report the results for the boosting approach called "LogitBoost" with 100 iterations (see Friedman et al., 2000

and Dettling and Bühlmann, 2003 for details), as well as the Bayesian approach (gsg-SSVS) of Ai-Jun and Xin-Yuan (2009) which was shown to outperform several other classification methods for the considered dataset (see Ai-Jun and Xin-Yuan, 2009, Table 2 for details).

Table 6.1.: Results from LOOCV for colon cancer data. Results for Bayesian approach (gsg-SSVS) are taken from Ai-Jun and Xin-Yuan (2009) and results for LogitBoost are taken from Dettling and Bühlmann (2003).

	$\hat{S}_{\text{b}}$ for EBIC_1	$\hat{S}_{\text{b}}$ for $\text{EBIC}_{0.6}$	gsg-SSVS	LogitBoost
Mean model size (sd)	1.10 (0.35)	2.57 (0.53)	6	10
LOOCV error rate	0.210	0.194	0.129	0.145

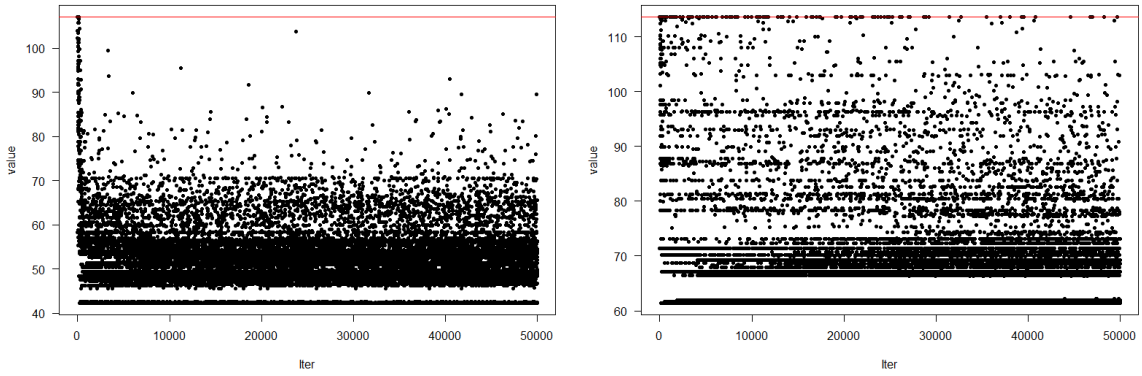
In Table 6.1, the LOOCV misclassification error rate (number of misclassified samples divided by total number of samples) is reported for the different classifiers, when using the common threshold of 0.5 for class assignment. The results indicate that the "best" models from BackAdaSub show a reasonable predictive performance despite the small number of selected genes. The Bayesian approach of Ai-Jun and Xin-Yuan (2009) and the boosting approach of Dettling and Bühlmann (2003) yield smaller LOOCV error rates, but at the "cost" of including more genes in the final models (note that, in contrast to BackAdaSub, the respective model sizes in gsg-SSVS and LogitBoost were chosen to be fixed). Again, we want to emphasize that the primary aim of BackAdaSub is not optimal predictive performance, but the identification of "truly important" genes associated with the disease status of a patient.

6.4.2. Leukemia dataset

As the second real data example we consider the leukemia dataset of Golub et al. (1999), which is available at `http://portals.broadinstitute.org/cgi-bin/cancer/datasets.cgi` and can be loaded from the R-package `golubEsets` (Golub, 2017). The original dataset consists of gene expression measurements of 6817 human genes for $n = 72$ patients, of which 47 suffer from acute lymphoblastic leukemia (ALL) and 25 suffer from acute myeloid leukemia (AML). We first apply the preprocessing steps described in Dudoit et al. (2002) (compare also Ai-Jun and Xin-Yuan, 2009): In a first thresholding step, all measured expression levels below 100 are set to 100 ("floor") and all expression levels above 16,000 are set to 16,000

("ceiling"). In the second step, those genes (i.e. columns of the desing matrix) are excluded from the analysis, for which max/min ≤ 5 and (max $-$ min) ≤ 500, where max and min denote to the maximum and minimum expression levels of a single gene, respectively. This procedure results in a restricted design matrix with $p = 3571$ columns (genes). Finally, the logarithm of basis 10 is applied to the remaining expression levels and then each row of the design matrix is standardized to have mean zero and variance one.

As before, we apply BackAdaSub on the preprocessed data using $q = 5$, $K = n$ and $T = 50,0000$. We consider the (negative) EBIC_γ as the selection criterion C with two different choices of the constant $\gamma \in \{0.6, 1\}$. The evolution of the values $\text{EBIC}_\gamma(S^{(t)})$, for $\gamma \in \{0.6, 1\}$, along the iterations (t) is given in Figure 6.10.



(a) $\gamma = 0.6$ (b) $\gamma = 1$

Figure 6.10.: BackAdaSub for leukemia data. Plots of the evolution of $\text{EBIC}_\gamma(S^{(t)})$ along the iterations t for (a) $\gamma = 0.6$ and (b) $\gamma = 1$. The red lines indicate the EBIC_γ-values of the thresholded model $\hat{S}_{0.9}$.

For $\gamma = 1$ (computation time approximately 10 minutes) as well as for $\gamma = 0.6$ (computation time approximately 22 minutes), no genes are selected to be in the thresholded model $\hat{S}_\rho$ with threshold $\rho = 0.9$ of BackAdaSub, which again indicates that no variable (gene) is "stable" in the sense of MOIP (compare Definition 6.3 and Theorem 6.3 of Section 6.2). However, for $\gamma = 1$, the single gene M23197_at is selected to be in the thresholded model $\hat{S}_{0.5}$ of BackAdaSub, while for $\gamma = 0.6$ the thresholded model $\hat{S}_{0.5}$ consists of the two genes X95735_at and M27891_at. Note that the genes X95735_at and M27891_at have the first and second largest posterior marginal inclusion probabilities in the Bayesian analysis of Ai-Jun and Xin-Yuan (2009), while M23197_at has the fourth largest posterior marginal inclusion probability. For $\gamma = 1$, the "best" model $\hat{S}_\text{b}$ found by BackAdaSub con-

sists of the single gene M23197_at and for $\gamma = 0.6$ it consists of the two genes L07633_at and M27891_at. It is again remarkable, that the gene L07633_at is not among the top 18 genes with the largest posterior marginal inclusion probability in the study of Ai-Jun and Xin-Yuan (2009), indicating that this is likely to be just an incidental finding.

We furthermore report the models selected by other variable selection methods: For $\gamma = 1$ and $\gamma = 0.6$, the Lasso selects the single gene M27891_at. The Adaptive Lasso selects also the single gene M27891_at for $\gamma = 1$, while for $\gamma = 0.6$ it selects the two genes M19507_at and M27891_at. For $\gamma = 1$, Forward Stepwise Selection yields a model with the single gene M23197_at, while for $\gamma = 0.6$ it selects the two genes M23197_at and M55150_at. Stability Selection with error control of one false positive selects a model including the following six genes: HG1612–HT1612_at, L07633_at, M23197_at, M27891_at, U82759_at and X95735_at. Note that the genes L07633_at and U82759_at are not among the 18 genes selected by Ai-Jun and Xin-Yuan (2009).

Table 6.2.: Results from LOOCV for leukemia data. Results for LogitBoost are taken from Dettling and Bühlmann (2003).

	$\hat{S}_{\text{b}}$ for EBIC_1	$\hat{S}_{\text{b}}$ for $\text{EBIC}_{0.6}$	LogitBoost
Mean model size (sd)	1.04 (0.20)	1.99 (0.20)	10
LOOCV error rate	0.125	0.056	0.056

Finally, in Table 6.2 we report the results from an external leave-one-out-cross-validation (LOOCV) for the "best" model from BackAdaSub when using the EBIC_γ with $\gamma = 1$ and $\gamma = 0.6$, as well as the LogitBoost algorithm with 100 iterations (Dettling and Bühlmann, 2003). The methods are applied in the same way as described at the end of Section 6.4.1 for the colon cancer dataset. Note that the results for the Bayesian approach of Ai-Jun and Xin-Yuan, 2009 are not included in Table 6.2 since they are not directly comparable due to the consideration of a separate training and test set. The results in Table 6.2 indicate that the "best" model from BackAdaSub shows a very good predictive performance. In particular, when using the $\text{EBIC}_{0.6}$, BackAdaSub yields the same LOOCV misclassification error as LogitBoost, even though the sizes of the selected models are significantly smaller.

All in all, the main conclusions for the leukemia dataset are very similar to the ones for

the colon cancer dataset described in the previous subsection: In order to obtain reliable results regarding the identification of sets of genes which are associated with different types of cancer, the given datasets of very small sample sizes are not sufficient and, although this might be costly, one should strive for collecting larger samples.

6.5. Conclusions

In this chapter we have introduced two natural variants of the original AdaSub method which make use of greedy stepwise methods for "solving" the sampled variable selection subproblems. Interestingly, FoAdaSub (Algorithm 6.1), the version based on Forward Stepwise Selection (FS2), resembles usual Forward Stepwise Selection, i.e. FoAdaSub converges (a.s.) against the model selected by FS2 as the number of iterations tends to infinity (see Theorem 6.1).

On the other hand, BackAdaSub (Algorithm 6.2), the version based on Backward Stepwise Selection (BS), generally does **not** resemble usual Backward Stepwise Selection (compare e.g. Figure 6.1), which could also not be applied in high-dimensional scenarios with $p > n$. Instead, we have argued that BackAdaSub can be used as a computationally efficient surrogate algorithm for the original AdaSub method. In particular, we have shown that BackAdaSub is guaranteed to converge against the best model according to the employed criterion, provided that the modified ordered importance property (MOIP) is satisfied (see Theorem 6.2). Even though the MOIP implies the OIP and thus BackAdaSub requires stronger conditions for the "correct convergence" than the original AdaSub method, results from simulation studies in normal linear models indicate that BackAdaSub performs similarly to AdaSub in many cases.

Furthermore, through simulated and real data examples in the setting of logistic regression models we have demonstrated that BackAdaSub shows desirable statistical properties in comparison to other variable selection methods such as Forward Stepwise Regression, the Lasso, the Adaptive Lasso, the SCAD and Stability Selection. Although BackAdaSub is computationally more "expensive" than convex methods like the Lasso, we have argued that the beneficial statistical performance of BackAdaSub can outweigh the (moderately) increased computational costs in many practical situations. In particular, we have concluded that the thresholded model from BackAdaSub shows favourable variable selection properties (with

generally small numbers of false positives), while the "best" model selected by BackAdaSub (with possibly larger numbers of false positives) may be preferred when the main target is good predictive performance.

7. Metropolized AdaSub for Bayesian variable selection

In the preceding chapters we have mainly focused on variable selection from a classical point of view, i.e. we have aimed at solving discrete optimization problems induced by different selection criteria which, under certain conditions, have desirable frequentist properties (like variable selection consistency). The Bayesian approach to the variable selection problem is fundamentally different, in the sense that we assign prior probabilities to each of the models in a pre-specified model space as well as prior distributions for all unknown parameters in the different models. After observing some data, Bayes' theorem is employed in order to "update" the prior through the likelihood of the observed data. The resulting posterior model distribution captures all the currently available information concerning the model uncertainty. In particular, posterior marginal inclusion probabilities provide "importance" measures for the individual explanatory variables; furthermore, the median probability model (Barbieri and Berger, 2004) or Bayesian model averaging (Raftery et al., 1997) can be used for predictive inference. However, sampling from the posterior model distribution is usually a very challenging problem, especially in a high-dimensional situation where the number of possible explanatory variables is very large.

In this chapter we present a Metropolized version of the original AdaSub method, called the Metropolized Adaptive Subspace (MAdaSub) algorithm, for efficient sampling from high-dimensional posterior model distributions. After a brief introduction to the Bayesian variable selection setting in Section 7.1, in Section 7.2 we introduce the MAdaSub algorithm, which can be viewed as an adaptive independent Metropolis-Hastings algorithm where the individual sampling probabilities of the explanatory variables are sequentially adjusted after each iteration. The employed updating scheme is inspired by AdaSub and can itself be motivated in a Bayesian way, so that the individual sampling probabilities finally converge against

the true respective posterior marginal inclusion probabilities. By making use of general results for adaptive Markov Chain Monte Carlo (MCMC) algorithms obtained by Roberts and Rosenthal (2007), in Section 7.3 we show that the proposed MAdaSub algorithm is ergodic despite its continuing adaptation. In Section 7.4 we conceptually compare the proposed method with other related approaches for high-dimensional Bayesian variable selection. In Section 7.5 we illustrate the performance of MAdaSub for low- and high-dimensional simulated data examples. Through real data examples with ten thousands of possible explanatory variables, in Section 7.6 we demonstrate that MAdaSub provides an efficient and stable way for sampling from very high-dimensional and multimodal posterior model distributions.

7.1. The Bayesian variable selection setting

In this section we briefly describe the Bayesian variable selection problem in a given GLM setting. Due to brevity, we do not provide a detailed review of Bayesian variable selection methods. We refer to George and McCulloch (1997), O'Hara and Sillanpää (2009) and Mallick and Yi (2013) for comprehensive treatments of Bayesian variable selection and corresponding algorithms.

Notation 7.1. As before, let $\mathcal{P} = \{1, \ldots, p\}$ denote the associated index set for explanatory variables $X_1, \ldots, X_p$ and let $\mathcal{M}_{\text{full}} = \{S;\, S \subseteq \mathcal{P}\}$ denote the full model space (see Notation 2.5 of Section 2.1). In a fully Bayesian approach we assign prior probabilities $\pi(S)$ to each of the models $S \in \mathcal{M}_{\text{full}}$ as well as priors $\pi(\mu_S, \psi_S, \boldsymbol{\beta}_S \,|\, S)$ for the parameters of each model $S \in \mathcal{M}_{\text{full}}$, where μ_S denotes the intercept (which is assumed to be always included in each model), ψ_S denotes a possibly present dispersion parameter (e.g. the variance in a normal linear model) and $\boldsymbol{\beta}_S$ denotes the vector of regression coefficients in the model S (see Notation 2.9 of Section 2.1). After observing some data $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$, with design matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ and response vector $\boldsymbol{y} \in \mathbb{R}^n$, the *posterior model probabilities* are proportional to

$$\pi(S \,|\, \mathcal{D}) \propto \pi(\boldsymbol{y} \,|\, \boldsymbol{X}, S)\, \pi(S)\,, \quad S \in \mathcal{M}_{\text{full}}\,, \tag{7.1}$$

where

$$\pi(\boldsymbol{y} \,|\, \boldsymbol{X}, S) = \int\int\int f(\boldsymbol{y} \,|\, \boldsymbol{X}_S, \mu_S, \psi_S, \boldsymbol{\beta}_S)\, \pi(\mu_S, \psi_S, \boldsymbol{\beta}_S \,|\, S)\, d\mu_S\, d\psi_S\, d\boldsymbol{\beta}_S \tag{7.2}$$

denotes the *marginal likelihood* of the data $\boldsymbol{y}$ under model S and $f(\boldsymbol{y} \,|\, \boldsymbol{X}_S, \mu_S, \psi_S, \boldsymbol{\beta}_S)$ denotes the likelihood of the data $\boldsymbol{y}$ under model S given the parameter values $\mu_S, \psi_S, \boldsymbol{\beta}_S$.

Note that the marginal likelihood $\pi(\boldsymbol{y} \,|\, \boldsymbol{X}, S)$ is generally only available in closed form when conjugate priors are used. A prominent and often used example is the normal-inverse-gamma prior in normal linear models, where the prior on the variance $\psi = \sigma^2$ is given by an inverse-gamma distribution (independent of the model S) and the prior on the vector of coefficients $\boldsymbol{\beta}_S$ in model $S \in \mathcal{M}_{\text{full}}$ is given by a multivariate normal distribution, i.e.

$$\boldsymbol{\beta}_S \,|\, S, \sigma^2 \sim \mathcal{N}_{|S|}(\boldsymbol{\theta}_S, \sigma^2 g\, \boldsymbol{V}_S), \quad \sigma^2 \sim \text{IG}(a, b)\,, \tag{7.3}$$

where $\boldsymbol{\theta}_S \in \mathbb{R}^{|S|}, g > 0, \boldsymbol{V}_S \in \mathbb{R}^{|S| \times |S|}$ and $a, b \in \mathbb{R}$ are hyperparameters. After centering each of the covariates $\boldsymbol{X}_j$, $j \in \mathcal{P}$, the improper prior $\pi(\mu) \propto 1$ is a common choice for the intercept μ (again, independent of the model S). With no specific prior information, the prior mean of $\boldsymbol{\beta}_S$ can be set to the zero vector $\boldsymbol{\theta}_S = \boldsymbol{0}$. The prior variance $\boldsymbol{V}_S$ is often chosen to be the identity matrix of dimension $|S|$ (implying prior independence of the regression coefficients and corresponding to Ridge Regression) or chosen to be $\boldsymbol{V}_S = (\boldsymbol{X}_S^T \boldsymbol{X}_S)^{-1}$ resulting in Zellner's g-prior (Zellner, 1986) given by

$$\boldsymbol{\beta}_S \,|\, S, \sigma^2 \sim \mathcal{N}_{|S|}(\boldsymbol{0}, \sigma^2 g (\boldsymbol{X}_S^T \boldsymbol{X}_S)^{-1}), \quad \pi(\sigma^2) \propto \frac{1}{\sigma^2}\,, \tag{7.4}$$

for which setting $g = n$ corresponds to a unit information prior. Note that in equation (7.4) we employ the common "non-informative" Jeffreys prior for the variance σ^2, which can also be viewed as a limiting case of the inverse-gamma prior in equation (7.3) for $a \to 0$ and $b \to 0$. Furthermore, note that for both prior choices in equations (7.3) and (7.4), the marginal likelihood is given in closed form (see e.g. Liang et al., 2008 and Lamnisos et al., 2013).

If no particular prior information about the explanatory variables is available, it is natural to employ i.i.d. Bernoulli priors for all possible explanatory variables X_j, $j \in \mathcal{P}$, leading to a prior on the model space of the form

$$\pi(S \,|\, \omega) = \omega^{|S|}(1-\omega)^{p-|S|},\ S \in \mathcal{M}_{\text{full}}\,, \tag{7.5}$$

where $\omega = \pi(j \in S)$ is the prior probability that variable X_j is included in the model, for $j \in \mathcal{P}$. One can either set the prior inclusion probability ω to some fixed value or

make use of an additional hyperprior for ω. A convenient choice is the (conjugate) beta prior $\omega \sim \mathcal{B}e(a_\omega, b_\omega)$, where $a_\omega > 0$ and $b_\omega > 0$ can be chosen in order to reflect the prior expectation and prior variance of the model size $s = |S|$, $S \in \mathcal{M}_{\text{full}}$ (see Kohn et al., 2001 for details).

In the general non-conjugate case (and actually for most prior choices in GLMs like logistic regression models) the marginal likelihood is not readily computable and numerical methods may be used in order to derive approximations to the marginal likelihood. Similarly, different information criteria like the Bayesian Information Criterion (BIC, Schwarz, 1978) or the Extended Bayesian Information Criterion (EBIC, Chen and Chen, 2008) can be used directly as asymptotic approximations to fully Bayesian posterior model probabilities under suitable choices of model priors (compare Section 2.2.2 as well as Liang et al., 2013). When employing the model prior underlying the EBIC (see equation (2.25) of Section 2.2.2) and a unit-information prior on the regression coefficients in each model, one can asymptotically approximate the kernel of the posterior $\pi(S \,|\, \mathcal{D}) \propto C(S)$ by

$$C(S) \approx \exp\left(-\frac{1}{2} \times \text{EBIC}_\gamma(S)\right) . \tag{7.6}$$

In this chapter we consider situations where the marginal likelihood $\pi(\boldsymbol{y} \,|\, \boldsymbol{X}, S)$ is available in closed form due to the use of conjugate priors or where an approximation to the kernel of $\pi(S \,|\, \mathcal{D})$ is used (e.g. information criteria like the EBIC). This simplifying assumption allows one to focus on the essential part of efficient sampling in very large model spaces, since additional sampling from the respective model parameters is not necessary. The extension of the presented algorithm to non-conjugate cases without the use of approximations may be possible by incorporating additional reversible-jump moves (compare Green, 1995) or by making use of similar mixture proposals as in the approach of Ji and Schmidler (2013).

Notation 7.2. In order to use a unified and simple notation, in the following let $C(S)$ denote either the exact kernel (i.e. $C(S) = \pi(\boldsymbol{y} \,|\, \boldsymbol{X}, S)\, \pi(S)$) or an approximation (e.g. given by equation (7.6)) to the kernel of the posterior $\pi(S \,|\, \mathcal{D})$, for $S \in \mathcal{M}_{\text{full}}$.

7.2. The MAdaSub algorithm

If we have specified all necessary prior distributions and have observed some data $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$, our aim is to sample from the (approximate) posterior distribution of models $\pi(S \,|\, \mathcal{D}) \propto C(S)$, $S \in \mathcal{M}_{\text{full}}$. Typically, Markov Chain Monte Carlo (MCMC) algorithms are employed in order to sample from the posterior model distribution (see e.g. George and McCulloch, 1993, Madigan et al., 1995, Carlin and Chib, 1995, Kuo and Mallick, 1998 and Dellaportas et al., 2002 for classical references as well as Section 7.4 for a discussion of more recent developments).

A particularly simple way to sample from a given target distribution is to use an independent Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970; Tierney, 1994), for which the proposed model is independent of the current model in each iteration of the algorithm. Clearly, the efficiency of such an independence sampler depends heavily on an appropriate choice of the proposal distribution, which, in the ideal situation, should be the same as the target distribution $\pi(S \,|\, \mathcal{D})$. In that case, one would obtain an independent sample from the target distribution with corresponding acceptance probability one. Of course, this reasoning is somewhat circular since we actually cannot sample directly from the target distribution which is the reason why we use MCMC methods in the first place. Adaptive MCMC algorithms want to break that cycle and update the proposal distribution during the algorithm based on the already obtained samples such that, in the case of the independence sampler, the proposal becomes closer and closer to the target distribution. However, especially in high-dimensional situations it is crucially important that the adaptation of the proposal as well as sampling from the proposal can be carried out efficiently (compare also Schäfer and Chopin, 2013, Section 5). For this reason, we restrict ourselves to proposal distributions which have an independent Bernoulli form, i.e. if $S \in \mathcal{M}_{\text{full}}$ is the current model, then we propose the model $V \in \mathcal{M}_{\text{full}}$ with probability

$$q(V \,|\, S; \boldsymbol{r}) \equiv q(V; \boldsymbol{r}) = \prod_{j \in V} r_j \prod_{j \in \mathcal{P} \setminus V} (1 - r_j)\,, \tag{7.7}$$

for some vector $\boldsymbol{r} = (r_1, \ldots, r_p)^T \in (0,1)^p$ of individual sampling probabilities.

The proposed Metropolized Adaptive Subspace (MAdaSub) method is provided next (given as Algorithm 7.1), while its key parameters are explained in the sequel.

Algorithm 7.1 Metropolized Adaptive Subspace (MAdaSub) method

Input:

- Data $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$.
- (Approximate) kernel of posterior $\pi(S \,|\, \mathcal{D}) \propto C(S)$ for $S \in \mathcal{M}_{\text{full}}$.
- Vector of initial selection probabilities $\boldsymbol{r}^{(0)} = \left(r_1^{(0)}, \ldots, r_p^{(0)}\right)^T \in (0,1)^p$.
- Parameters $L_j > 0$ for $j \in \mathcal{P}$, controlling the adaptation rate of the algorithm (e.g. $L_j = L = p$).
- Constant $\epsilon \in (0, 0.5)$ (chosen to be small, e.g. $\epsilon = 10^{-6}$).
- Number of iterations $T \in \mathbb{N}$.

Algorithm:

(1) Sample $b_j^{(0)} \sim \text{Bernoulli}\left(r_j^{(0)}\right)$ independently for $j \in \mathcal{P}$. Set $S^{(0)} = \{j \in \mathcal{P};\ b_j^{(0)} = 1\}$.

(2) For $t = 1, \ldots, T$:

(a) Compute truncated vector of selection probabilities $\tilde{\boldsymbol{r}}^{(t-1)} = \left(\tilde{r}_1^{(t-1)}, \ldots, \tilde{r}_p^{(t-1)}\right)^T$, i.e. for $j \in \mathcal{P}$ set

$$\tilde{r}_j^{(t-1)} = \begin{cases} r_j^{(t-1)} & , \text{if } r_j^{(t-1)} \in [\epsilon, 1-\epsilon]\,, \\ \epsilon & , \text{if } r_j^{(t-1)} < \epsilon\,, \\ 1-\epsilon & , \text{if } r_j^{(t-1)} > 1-\epsilon\,. \end{cases}$$

(b) Draw $b_j^{(t)} \sim \text{Bernoulli}\left(\tilde{r}_j^{(t-1)}\right)$ independently for $j \in \mathcal{P}$.

(c) Set $V^{(t)} = \{j \in \mathcal{P};\ b_j^{(t)} = 1\}$.

(d) Compute acceptance probability

$$\alpha^{(t)} \equiv \alpha(V^{(t)} \,|\, S^{(t-1)}; \tilde{\boldsymbol{r}}^{(t-1)}) = \min\left\{ \frac{C(V^{(t)})\, q(S^{(t-1)} \,|\, V^{(t)}; \tilde{\boldsymbol{r}}^{(t-1)})}{C(S^{(t-1)})\, q(V^{(t)} \,|\, S^{(t-1)}; \tilde{\boldsymbol{r}}^{(t-1)})},\, 1 \right\}.$$

(e) Set $S^{(t)} = \begin{cases} V^{(t)} & , \text{with probability } \alpha^{(t)}, \\ S^{(t-1)} & , \text{with probability } 1-\alpha^{(t)}. \end{cases}$

(f) Update vector of selection probabilities $\boldsymbol{r}^{(t)} = \left(r_1^{(t)}, \ldots, r_p^{(t)}\right)^T$ via

$$r_j^{(t)} = \frac{L_j r_j^{(0)} + \sum_{i=1}^{t} \mathbb{1}_{S^{(i)}}(j)}{L_j + t}\,, \quad j \in \mathcal{P}\,.$$

Output:

- Approximate sample $S^{(b+1)}, \ldots, S^{(T)}$ from posterior distribution $\pi(\cdot \,|\, \mathcal{D})$, after burn-in period of length b.

The fundamental idea of the MAdaSub algorithm is to sequentially update the individual sampling probabilities according to the currently "estimated" posterior marginal inclusion probabilities. In more detail, after initializing the vector of sampling probabilities $\boldsymbol{r}^{(0)} = \left(r_1^{(0)}, \ldots, r_p^{(0)}\right)^T \in (0,1)^p$, the individual sampling probabilities $r_j^{(t)}$ of variables X_j are updated after each iteration t of the algorithm, such that $r_j^{(t)}$ finally converges to the correct posterior inclusion probability $\pi_j = \pi(j \in S \,|\, \mathcal{D})$, as $t \to \infty$ (see Corollary 7.6 in Section 7.3). This means that, "in the limit", we make use of the proposal

$$q(V; \boldsymbol{r}^*) = \prod_{j \in V} \pi_j \prod_{j \in \mathcal{P} \setminus V} (1 - \pi_j), \quad V \in \mathcal{M}_{\text{full}}\,, \quad \text{with } \boldsymbol{r}^* = (\pi_1, \ldots, \pi_p)^T\,, \tag{7.8}$$

which is the closest distribution (in terms of Kullback-Leibler divergence) to the actual target $\pi(\cdot \,|\, \mathcal{D})$, among all distributions of the independent Bernoulli form (7.7) (see Clyde et al., 2011). For $j \in \mathcal{P}$, the concrete update of $r_j^{(t)}$ after iteration $t \in \mathbb{N}$ is given by

$$r_j^{(t)} = \frac{L_j r_j^{(0)} + \sum_{i=1}^{t} \mathbb{1}_{S^{(i)}}(j)}{L_j + t}\,, \tag{7.9}$$

where, for $j \in \mathcal{P}$, $L_j > 0$ are additional parameters controlling the adaptation rate of the algorithm and $S^{(1)}, \ldots, S^{(t)}$ denote the sampled models in the first t iterations of the MAdaSub algorithm. The updating scheme of the individual selection probabilities is inspired by the AdaSub method and can itself be motivated in a Bayesian way (compare Section 3.4): Since we do not know the true posterior marginal inclusion probability π_j of variable X_j for $j \in \mathcal{P}$, we put an independent beta prior on π_j with the following parametrization

$$\pi_j \sim \mathcal{B}e\left(L_j r_j^{(0)},\, L_j\left(1 - r_j^{(0)}\right)\right)\,, \tag{7.10}$$

where $r_j^{(0)} = E[\pi_j]$ is the prior expectation of π_j and the parameter $L_j > 0$ controls the variance of π_j via

$$\text{Var}(\pi_j) = \frac{1}{L_j + 1} \times r_j^{(0)}\left(1 - r_j^{(0)}\right)\,. \tag{7.11}$$

If $L_j \to 0$, then $\text{Var}(\pi_j) \to r_j^{(0)}\,(1 - r_j^{(0)})$, which is the variance of a Bernoulli random variable with mean $r_j^{(0)}$. If $L_j \to \infty$, then $\text{Var}(\pi_j) \to 0$.

Now, one might view the samples $S^{(1)}, \ldots, S^{(t)}$ obtained from the adaptive MCMC algorithm after iteration t as "new" data and interpret the information learned about π_j as t Bernoulli trials where $j \in S^{(i)}$ corresponds to "success", while $j \notin S^{(i)}$ corresponds to "fail-

ure" ($i = 1, \ldots, t$). When ignoring the dependence of the Bernoulli trials, then the (pseudo) posterior of π_j after iteration t of the algorithm is given by

$$\pi_j \,|\, S^{(1)}, \ldots, S^{(t)} \sim \mathcal{B}e\left(L_j r_j^{(0)} + \sum_{i=1}^{t} \mathbb{1}_{S^{(i)}}(j),\; L_j(1 - r_j^{(0)}) + \sum_{i=1}^{t} \mathbb{1}_{\mathcal{P}\setminus S^{(i)}}(j)\right), \tag{7.12}$$

with posterior expectation

$$E(\pi_j \,|\, S^{(1)}, \ldots, S^{(t)}) = \frac{L_j r_j^{(0)} + \sum_{i=1}^{t} \mathbb{1}_{S^{(i)}}(j)}{L_j + t} = r_j^{(t)} \tag{7.13}$$

and posterior variance

$$\mathrm{Var}(\pi_j \,|\, S^{(1)}, \ldots, S^{(t)}) = \frac{1}{L_j + t + 1} \times r_j^{(t)} \left(1 - r_j^{(t)}\right). \tag{7.14}$$

The interpretation of $r_j^{(0)}$ as the prior expectation of the posterior inclusion probability π_j calls for the choice $r_j^{(0)} = \pi(j \in S)$ as the actual prior inclusion probability of variable X_j, as well as L_j as the actual prior marginal variance of inclusion of variable X_j. If no particular prior information about specific variables is available, but the prior expected model size is $q \in (0, p)$, then we recommend to set $r_j^{(0)} = \frac{q}{p}$ and $L_j = L = p$ for all $j \in \mathcal{P}$. In this particular situation, equation (7.13) reduces to

$$E(\pi_j \,|\, S^{(1)}, \ldots, S^{(t)}) = \frac{q + \sum_{i=1}^{t} \mathbb{1}_{S^{(i)}}(j)}{p + t} = r_j^{(t)}. \tag{7.15}$$

Remark 7.1. Note that equation (7.15) is closely related to the AdaSub update (see equation (3.3) in Section 3.2). An important difference between the AdaSub and the MAdaSub algorithm is that in AdaSub we only learn something about variable X_j in iteration t if $j \in V^{(t)}$ (i.e. if the variable is included in the sampled subspace in which the optimization is carried out), while in MAdaSub we obtain "new knowledge" about variable X_j in each iteration. Referring to the situation corresponding to equation (7.15), this might also explain the empirical observation that it is beneficial to choose a larger adaptation rate in AdaSub (i.e. $K = n$ in equation (3.3) corresponding to the smaller "prior variance" parameter $L = p/n$) in comparison to MAdaSub (i.e. $L = p$, corresponding to $K = 1$ for AdaSub).

Even though it seems natural to choose the parameters $r_j^{(0)}$ and L_j of the algorithm as the respective prior quantities, this choice is not imperative. In fact, simulations indicate that choosing $r_j^{(0)} = \frac{q}{p}$ with $q \in [2, 20]$ and $L_j = p$ for all $j \in \mathcal{P}$ yields a stable and well mixing algorithm in most sparse high-dimensional situations irrespective of the actual

prior. Furthermore, if one has already run and stopped the adaptive MCMC algorithm after some number of iterations T, then one can simply restart the algorithm with the already updated parameters $r_j^{(T)}$ and $L_j + T$ (compare equation (7.14)) as new starting values for the corresponding parameters.

Note that the additional "truncation" step 2 (a) in the MAdaSub algorithm ensures that the truncated individual selection probabilities $\tilde{r}_j^{(t)}$, $j \in \mathcal{P}$, are always included in the compact interval $\mathcal{I} = [\epsilon, 1-\epsilon]$, where $\epsilon \in (0, 0.5)$ is a pre-specified "precision" parameter (chosen to be small, e.g. $\epsilon = 10^{-6}$). This adjustment simplifies the proof of the ergodicity of MAdaSub and can further improve the mixing of the algorithm.

7.3. Ergodicity of MAdaSub

In this section we want to show that the MAdaSub algorithm is ergodic (see Theorem 7.1), meaning that "in the limit" we sample from the targeted posterior model distribution $\pi(\cdot \,|\, \mathcal{D})$, despite the continuing adaptation in MAdaSub. We will make use of a general ergodicity result for adaptive MCMC algorithms which has been obtained by Roberts and Rosenthal (2007). In order to state the result directly for the specific setting of the MAdaSub algorithm, we first introduce some additional notation.

Notation 7.3. (a) In the following, the models $S^{(0)}, S^{(1)}, S^{(2)}, \ldots$ generated by the MAdaSub algorithm should be viewed as random variables with values in the full model space $\mathcal{M}_{\text{full}}$. Furthermore, the (truncated) vectors of selection probabilities $\tilde{\boldsymbol{r}}^{(t)} = \left(\tilde{\boldsymbol{r}}_1^{(t)}, \ldots, \tilde{\boldsymbol{r}}_p^{(t)}\right)^T$, $t \in \mathbb{N}$ should be viewed as random vectors with values in the compact set $\mathcal{I}^p = [\epsilon, 1-\epsilon]^p$.

(b) For a (current) model $S \in \mathcal{M}_{\text{full}}$ and a vector of selection probabilities $\tilde{\boldsymbol{r}} \in [\epsilon, 1-\epsilon]^p$, let $P(\cdot \,|\, S; \tilde{\boldsymbol{r}})$ denote the one-step transition kernel of MAdaSub, i.e. for iteration $t \in \mathbb{N}$ of MAdaSub and a subset of models $A' \subseteq \mathcal{M}_{\text{full}}$ we have

$$P(A' \,|\, S; \tilde{\boldsymbol{r}}) = P\left(S^{(t)} \in A' \,|\, S^{(t-1)} = S, \tilde{\boldsymbol{r}}^{(t-1)} = \tilde{\boldsymbol{r}}\right). \tag{7.16}$$

In particular, for $S' \in \mathcal{M}_{\text{full}}$, let $P(S' \,|\, S; \tilde{\boldsymbol{r}}) \equiv P(\{S'\} \,|\, S; \tilde{\boldsymbol{r}})$ denote the probability that the next state of the MAdaSub chain is $S^{(t)} = S'$, given the current model $S^{(t-1)} = S$ and the current vector of selection probabilities $\tilde{\boldsymbol{r}}^{(t-1)} = \tilde{\boldsymbol{r}}$. Note that for $\tilde{\boldsymbol{r}} \in [\epsilon, 1-\epsilon]^p$

and $S, S' \in \mathcal{M}_{\text{full}}$ with $S \neq S'$ we have

$$P(S' \mid S; \tilde{\boldsymbol{r}}) = q(S'; \tilde{\boldsymbol{r}})\, \alpha(S' \mid S; \tilde{\boldsymbol{r}})\,, \tag{7.17}$$

where $q(S'; \tilde{\boldsymbol{r}})$ is the probability of proposing the model S' and $\alpha(S' \mid S; \tilde{\boldsymbol{r}})$ is the corresponding acceptance probability.

(c) For $t \in \mathbb{N}$, $S \in \mathcal{M}_{\text{full}}$, $A' \subseteq \mathcal{M}_{\text{full}}$ and $\tilde{\boldsymbol{r}} \in [\epsilon, 1-\epsilon]^p$ let

$$P^{(t)}(A' \mid S; \tilde{\boldsymbol{r}}) := P\left(S^{(t)} \in A' \,\middle|\, S^{(0)} = S, \tilde{\boldsymbol{r}}^{(0)} = \ldots = \tilde{\boldsymbol{r}}^{(t-1)} = \tilde{\boldsymbol{r}}\right) \tag{7.18}$$

denote the t-step transition kernel of MAdaSub when the vector of selection probabilities $\tilde{\boldsymbol{r}}$ is fixed (i.e. not adapted during the algorithm). Similarly, let

$$Q^{(t)}(A' \mid S; \tilde{\boldsymbol{r}}) := P\left(S^{(t)} \in A' \,\middle|\, S^{(0)} = S, \tilde{\boldsymbol{r}}^{(0)} = \tilde{\boldsymbol{r}}\right) \tag{7.19}$$

denote the t-step transition kernel for the first t iterations of MAdaSub, given only the initial conditions $S^{(0)} = S$ and $\tilde{\boldsymbol{r}}^{(0)} = \tilde{\boldsymbol{r}}$.

The following theorem provides the ergodicity result of Roberts and Rosenthal (2007, Theorem 1) adjusted to the specific setting of MAdaSub.

Theorem 7.1 (Roberts and Rosenthal, 2007)**.** *Consider the MAdaSub algorithm with initial parameters $\boldsymbol{r}^{(0)} \in (0,1)^p$, $L_j > 0$ and $\epsilon \in (0, 0.5)$. Suppose that for each fixed vector of selection probabilities $\tilde{\boldsymbol{r}} \in [\epsilon, 1-\epsilon]^p$, the one-step kernel $P(\cdot \mid \cdot\,; \tilde{\boldsymbol{r}})$ of MAdaSub is stationary for the target distribution $\pi(\cdot \mid \mathcal{D})$, i.e. for all $S' \in \mathcal{M}_{full}$ we have*

$$\pi(S' \mid \mathcal{D}) = \sum_{S \in \mathcal{M}_{full}} P(S' \mid S; \tilde{\boldsymbol{r}})\, \pi(S \mid \mathcal{D})\,. \tag{7.20}$$

Further suppose that the following two conditions hold:

*(a) The **simultaneous uniform ergodicity** condition is satisfied, i.e. for all $\delta > 0$, there exists an integer $T \in \mathbb{N}$ such that*

$$\left\| P^{(T)}(\cdot \mid S; \tilde{\boldsymbol{r}}) - \pi(\cdot \mid \mathcal{D}) \right\|_{TV} \leq \delta \tag{7.21}$$

for all $S \in \mathcal{M}_{full}$ and $\tilde{\boldsymbol{r}} \in [\epsilon, 1-\epsilon]^p$, where $\|P_1 - P_2\|_{TV} = \sup_{A \in \mathfrak{A}} |P_1(A) - P_2(A)|$ denotes the total variation distance between two distributions P_1 and P_2 defined on some common measurable space $(\Omega, \mathfrak{A})$.

*(b) The **diminishing adaptation** condition is satisfied, i.e. we have*

$$\max_{S\in\mathcal{M}_{full}} \left\| P(\,\cdot\,|\,S;\tilde{\boldsymbol{r}}^{(t)}) - P(\,\cdot\,|\,S;\tilde{\boldsymbol{r}}^{(t-1)}) \right\|_{TV} \xrightarrow{P} 0\,, \quad t\to\infty\,, \tag{7.22}$$

where $\tilde{\boldsymbol{r}}^{(t)}$ and $\tilde{\boldsymbol{r}}^{(t-1)}$ are random vectors of selection probabilities induced by the MAdaSub algorithm (see Notation 7.3).

*Then the MAdaSub algorithm is **ergodic**, i.e. for all $S\in\mathcal{M}_{full}$ and $\tilde{\boldsymbol{r}}\in[\epsilon,1-\epsilon]^p$ we have*

$$\left\| Q^{(t)}(\cdot\,|\,S;\tilde{\boldsymbol{r}}) - \pi(\cdot\,|\,\mathcal{D}) \right\|_{TV} \to 0, \quad t\to\infty\,. \tag{7.23}$$

*Furthermore, the **weak law of large numbers** holds for MAdaSub, i.e. for any function $g:\mathcal{M}_{full}\to\mathbb{R}$ we have*

$$\frac{1}{t}\sum_{i=1}^{t} g(S^{(i)}) \xrightarrow{P} E[g\,|\,\mathcal{D}]\,, \tag{7.24}$$

where $E[g\,|\,\mathcal{D}] = \sum_S g(S)\pi(S\,|\,\mathcal{D})$ denotes the posterior expectation of g.

In the following we will show that MAdaSub satisfies both the simultaneous uniform ergodicity condition and the diminishing adaptation condition, so that Theorem 7.1 can be applied.

Lemma 7.2. *The simultaneous uniform ergodicity condition is satisfied for the MAdaSub algorithm for all choices of $\boldsymbol{r}^{(0)}\in(0,1)^p$, $L_j>0$ and $\epsilon\in(0,0.5)$.*

Proof. Here we make use of a very similar argumentation as in the proof of Lemma 1 in Griffin et al. (2017). We show that $\mathcal{M}_{\text{full}}$ is a *1-small set* (see Roberts and Rosenthal, 2004, Section 3.3), i.e. there exists $\beta>0$ and a probability measure ν on $\mathcal{M}_{\text{full}}$ such that $P(A'\,|\,S;\tilde{\boldsymbol{r}})\geq\beta\nu(A')$ for all $S\in\mathcal{M}_{\text{full}}$, $A'\subseteq\mathcal{M}_{\text{full}}$ and $\tilde{\boldsymbol{r}}\in[\epsilon,1-\epsilon]^p$. Then by Theorem 8 in Roberts and Rosenthal (2004), the simultaneous uniform ergodicity condition is satisfied. In order to prove that $\mathcal{M}_{\text{full}}$ is 1-small (note that $\mathcal{M}_{\text{full}}$ is finite), it suffices to show that there exists a constant $\beta_0>0$ such that $P(S'\,|\,S;\tilde{\boldsymbol{r}})\geq\beta_0$ for all $S,S'\in\mathcal{M}_{\text{full}}$ and all $\tilde{\boldsymbol{r}}\in[\epsilon,1-\epsilon]^p$. Indeed, for $S,S'\in\mathcal{M}_{\text{full}}$ and $\tilde{\boldsymbol{r}}\in[\epsilon,1-\epsilon]^p$ it holds

$$\begin{aligned}
P(S'\,|\,S;\tilde{\boldsymbol{r}}) &\geq q(S';\tilde{\boldsymbol{r}})\,\alpha(S'\,|\,S;\tilde{\boldsymbol{r}}) \\
&= \Bigg(\prod_{j\in S'}\underbrace{\tilde{r}_j}_{\geq\epsilon}\Bigg)\Bigg(\prod_{j\in\mathcal{P}\setminus S'}\underbrace{(1-\tilde{r}_j)}_{\geq\epsilon}\Bigg)\min\left\{\frac{\pi(S'\,|\,\mathcal{D})\,q(S;\tilde{\boldsymbol{r}})}{\pi(S\,|\,\mathcal{D})\,q(S';\tilde{\boldsymbol{r}})},1\right\} \\
&\geq \epsilon^p\pi(S'\,|\,\mathcal{D})\,q(S;\tilde{\boldsymbol{r}}) \geq \epsilon^{2p}\min_{S\in\mathcal{M}_{\text{full}}}\pi(S\,|\,\mathcal{D}) =: \beta_0\,.
\end{aligned}$$

This completes the proof. □

In order to show that the diminishing adaptation condition is satisfied for the MAdaSub algorithm, we will make repeated use of the following simple observation.

Lemma 7.3. *Let $m \in \mathbb{N}$ be fixed. For $j \in \{1, \dots, m\}$ let $\left(a_j^{(t)}\right)_{t\in\mathbb{N}_0}$ be bounded sequences of real numbers $a_j^{(t)} \in \mathbb{R}$ with $|a_j^{(t)} - a_j^{(t-1)}| \to 0$ for $t \to \infty$. Then we have*

$$\left|\prod_{j=1}^{m} a_j^{(t)} - \prod_{j=1}^{m} a_j^{(t-1)}\right| \to 0, \quad t \to \infty\,. \tag{7.25}$$

Proof. Since $\left(a_j^{(t)}\right)_{t\in\mathbb{N}_0}$ are bounded sequences, there are constants $L_j > 0$ so that $|a_j^{(t)}| \leq L_j$ for all $t \in \mathbb{N}_0$ and $j \in \{1, \dots, m\}$. We proceed by induction on $m \in \mathbb{N}$: Equation (7.25) obviously holds for $m = 1$. Now suppose that the assertion holds for $m - 1$ and we want to show that it also holds for m. Then we have

$$\begin{aligned}\left|\prod_{j=1}^{m} a_j^{(t)} - \prod_{j=1}^{m} a_j^{(t-1)}\right| &\leq \left|a_m^{(t)} \prod_{j=1}^{m-1} a_j^{(t)} - a_m^{(t-1)} \prod_{j=1}^{m-1} a_j^{(t)}\right| + \left|a_m^{(t-1)} \prod_{j=1}^{m-1} a_j^{(t)} - a_m^{(t-1)} \prod_{j=1}^{m-1} a_j^{(t-1)}\right| \\ &= \underbrace{\prod_{j=1}^{m-1} \left|a_j^{(t)}\right|}_{\leq \prod_{j=1}^{m-1} L_j} \times \underbrace{\left|a_m^{(t)} - a_m^{(t-1)}\right|}_{\to 0} + \underbrace{\left|a_m^{(t-1)}\right|}_{\leq L_m} \times \underbrace{\left|\prod_{j=1}^{m-1} a_j^{(t)} - \prod_{j=1}^{m-1} a_j^{(t-1)}\right|}_{\to 0} \overset{t\to\infty}{\to} 0\,.\end{aligned}$$

□

Lemma 7.4. *Consider the application of the MAdaSub algorithm on a given dataset $\mathcal{D}$ with some tuning parameter choices $\boldsymbol{r}^{(0)} \in (0,1)^p$, $L_j > 0$ and $\epsilon \in (0, 0.5)$. Then, for $j \in \mathcal{P}$, we have*

$$\left|\tilde{r}_j^{(t)} - \tilde{r}_j^{(t-1)}\right| \overset{a.s.}{\to} 0, \quad t \to \infty\,. \tag{7.26}$$

Furthermore, for all $S, S' \in \mathcal{M}_{full}$ it holds

$$\left|P(S' \mid S; \tilde{\boldsymbol{r}}^{(t)}) - P(S' \mid S; \tilde{\boldsymbol{r}}^{(t-1)})\right| \overset{a.s.}{\to} 0, \quad t \to \infty\,. \tag{7.27}$$

In particular, MAdaSub fulfils the diminishing adaptation condition.

Proof. For $j \in \mathcal{P}$ we have

$$\begin{aligned}\left|\tilde{r}_j^{(t)} - \tilde{r}_j^{(t-1)}\right| &\leq \left|r_j^{(t)} - r_j^{(t-1)}\right| \\ &\leq \left|\frac{L_j r_j^{(0)} + \sum_{i=1}^{t} \mathbb{1}_{S^{(i)}}(j)}{L_j + t} - \frac{L_j r_j^{(0)} + \sum_{i=1}^{t-1} \mathbb{1}_{S^{(i)}}(j)}{L_j + t - 1}\right|\end{aligned}$$

$$
\begin{aligned}
&\leq \left| \frac{L_j r_j^{(0)} + \sum_{i=1}^{t} \mathbb{1}_{S^{(i)}}(j)}{L_j + t} - \frac{L_j r_j^{(0)} + \sum_{i=1}^{t} \mathbb{1}_{S^{(i)}}(j)}{L_j + t - 1} \right| \\
&\quad + \left| \frac{L_j r_j^{(0)} + \sum_{i=1}^{t} \mathbb{1}_{S^{(i)}}(j)}{L_j + t - 1} - \frac{L_j r_j^{(0)} + \sum_{i=1}^{t-1} \mathbb{1}_{S^{(i)}}(j)}{L_j + t - 1} \right| \\
&\leq \underbrace{\frac{L_j r_j^{(0)} + \sum_{i=1}^{t} \mathbb{1}_{S^{(i)}}(j)}{L_j + t}}_{\in(0,1)} \times \underbrace{\frac{1}{L_j + t - 1}}_{\to 0} + \underbrace{\frac{1}{L_j + t - 1}}_{\to 0} \overset{\text{a.s.}}{\to} 0, \quad t \to \infty \,.
\end{aligned}
$$

With Lemma 7.3 (set $m = p$ and note that the number of variables $p = |\mathcal{P}|$ is fixed for the given dataset) we conclude that for $V \in \mathcal{M}_{\text{full}}$ it holds

$$
\left| q(V; \tilde{\boldsymbol{r}}^{(t)}) - q(V; \tilde{\boldsymbol{r}}^{(t-1)}) \right| = \left| \prod_{j \in V} \tilde{r}_j^{(t)} \prod_{j \in \mathcal{P} \setminus V} \left(1 - \tilde{r}_j^{(t)}\right) - \prod_{j \in V} \tilde{r}_j^{(t-1)} \prod_{j \in \mathcal{P} \setminus V} \left(1 - \tilde{r}_j^{(t-1)}\right) \right| \overset{\text{a.s.}}{\to} 0 \,. \tag{7.28}
$$

Let $S, S' \in \mathcal{M}_{\text{full}}$ and suppose that $S \neq S'$. Then we have

$$
\left| P(S' \,|\, S; \tilde{\boldsymbol{r}}^{(t)}) - P(S' \,|\, S; \tilde{\boldsymbol{r}}^{(t-1)}) \right| = \left| q(S'; \tilde{\boldsymbol{r}}^{(t)}) \alpha(S' \,|\, S; \tilde{\boldsymbol{r}}^{(t)}) - q(S'; \tilde{\boldsymbol{r}}^{(t-1)}) \alpha(S' \,|\, S; \tilde{\boldsymbol{r}}^{(t-1)}) \right| . \tag{7.29}
$$

Note that $q(S'; \tilde{\boldsymbol{r}}^{(t)}) \in [\epsilon^p, (1-\epsilon)^p]$ and $\alpha(S' \,|\, S; \tilde{\boldsymbol{r}}^{(t)}) \in [0, 1]$ for all $t \in \mathbb{N}_0$. Furthermore, we have already shown that $\left| q(S'; \boldsymbol{r}^{(t)}) - q(S'; \boldsymbol{r}^{(t-1)}) \right| \overset{\text{a.s.}}{\to} 0$ for all $S' \in \mathcal{M}_{\text{full}}$. Therefore, we also have

$$
\begin{aligned}
\left| \alpha(S' \,|\, S; \tilde{\boldsymbol{r}}^{(t)}) - \alpha(S' \,|\, S; \tilde{\boldsymbol{r}}^{(t-1)}) \right| &\leq \left| \frac{C(S')\, q(S; \tilde{\boldsymbol{r}}^{(t)})}{C(S)\, q(S'; \tilde{\boldsymbol{r}}^{(t)})} - \frac{C(S')\, q(S; \tilde{\boldsymbol{r}}^{(t-1)})}{C(S)\, q(S'; \tilde{\boldsymbol{r}}^{(t-1)})} \right| \\
&= \frac{C(S')}{C(S)} \left| \frac{q(S; \tilde{\boldsymbol{r}}^{(t)})}{q(S'; \tilde{\boldsymbol{r}}^{(t)})} - \frac{q(S; \tilde{\boldsymbol{r}}^{(t-1)})}{q(S'; \tilde{\boldsymbol{r}}^{(t-1)})} \right| \overset{\text{a.s.}}{\to} 0 \,,
\end{aligned} \tag{7.30}
$$

where we made use of Lemma 7.3 with $m = 2$ and

$$
a_1^{(t)} = q(S; \tilde{\boldsymbol{r}}^{(t)}) \in [\epsilon^p, (1-\epsilon)^p] \quad \text{and} \quad a_2^{(t)} = \frac{1}{q(S'; \tilde{\boldsymbol{r}}^{(t)})} \in [(1-\epsilon)^{-p}, \epsilon^{-p}] \,, \quad t \in \mathbb{N}_0 \,,
$$

noting that

$$
\left| a_2^{(t)} - a_2^{(t-1)} \right| = \underbrace{\frac{1}{q(S'; \tilde{\boldsymbol{r}}^{(t)}) q(S'; \tilde{\boldsymbol{r}}^{(t-1)})}}_{\leq \epsilon^{-2p}} \left| q(S'; \tilde{\boldsymbol{r}}^{(t)}) - q(S'; \tilde{\boldsymbol{r}}^{(t-1)}) \right| \overset{\text{a.s.}}{\to} 0 \,.
$$

Again by using Lemma 7.3 and combining equations (7.28), (7.29) and (7.30) we conclude

that

$$\left|P(S' \mid S; \tilde{\boldsymbol{r}}^{(t)}) - P(S' \mid S; \tilde{\boldsymbol{r}}^{(t-1)})\right| \overset{\text{a.s.}}{\to} 0\,.$$

Finally, we consider the case $S = S'$. Then it holds

$$\begin{aligned}\left|P(S \mid S; \tilde{\boldsymbol{r}}^{(t)}) - P(S \mid S; \tilde{\boldsymbol{r}}^{(t-1)})\right| &= \left|1 - \sum_{S' \neq S} P(S' \mid S; \tilde{\boldsymbol{r}}^{(t)}) - \left(1 - \sum_{S' \neq S} P(S' \mid S; \tilde{\boldsymbol{r}}^{(t-1)})\right)\right| \\ &\leq \sum_{S' \neq S} \left|P(S' \mid S; \tilde{\boldsymbol{r}}^{(t)}) - P(S' \mid S; \tilde{\boldsymbol{r}}^{(t-1)})\right| \overset{\text{a.s.}}{\to} 0\,.\end{aligned}$$

Thus we have shown that equation (7.27) holds for all $S, S' \in \mathcal{M}_{\text{full}}$. In particular, we conclude that the diminishing adaptation condition is satisfied for MAdaSub (recall that almost sure convergence implies convergence in probability). □

Theorem 7.5. *The MAdaSub algorithm (Algorithm 7.1) is ergodic for all choices of* $\boldsymbol{r}^{(0)} \in (0,1)^p$, $L_j > 0$ *and* $\epsilon \in (0, 0.5)$ *and fulfils the weak law of large numbers.*

Proof. The MAdaSub algorithm fulfils the simultaneous uniform ergodicity condition (see Lemma 7.2) and the diminishing adaptation condition (see Lemma 7.4). Furthermore, for each fixed $\tilde{\boldsymbol{r}} \in [\epsilon, 1-\epsilon]^p$, the corresponding transition kernel $P(\cdot \mid \cdot\,; \tilde{\boldsymbol{r}})$ is induced by a simple Metropolis-Hastings step and therefore has the desired target distribution $\pi(\cdot \mid \mathcal{D})$ as its stationary distribution. Therefore, by Theorem 7.1 the MAdaSub algorithm is ergodic and fulfils the weak law of large numbers. □

Corollary 7.6. *For all choices of* $\boldsymbol{r}^{(0)} \in (0,1)^p$, $L_j > 0$ *and* $\epsilon \in (0, 0.5)$*, the ("untruncated") selection probabilities* $r_j^{(t)}$ *of the explanatory variables* X_j, $j \in \mathcal{P}$ *in MAdaSub converge (in probability) to the respective posterior marginal inclusion probabilities* $\pi_j = \pi(j \in S \mid \mathcal{D})$*, i.e. for all* $j \in \mathcal{P}$ *it holds that* $r_j^{(t)} \overset{P}{\to} \pi_j$ *as* $t \to \infty$*.*

Proof. Since MAdaSub fulfils the weak law of large numbers (Theorem 7.5), for $j \in \mathcal{P}$ it holds that

$$\frac{1}{t}\sum_{i=1}^{t} \mathbb{1}_{S^{(i)}}(j) \overset{\text{P}}{\to} \pi_j, \quad t \to \infty\,.$$

Hence, for $j \in \mathcal{P}$, we also have

$$r_j^{(t)} = \frac{L_j r_j^{(0)} + \sum_{i=1}^{t} \mathbb{1}_{S^{(i)}}(j)}{L_j + t} \overset{\text{P}}{\to} \pi_j, \quad t \to \infty\,.$$

□

Remark 7.2. Note that even though we have shown that the MAdaSub algorithm is ergodic for all possible choices of $\boldsymbol{r}^{(0)} \in (0,1)^p$, $L_j > 0$ and $\epsilon \in (0, 0.5)$, the appropriate choice of these tuning parameters is very important. In fact, for "bad" choices of these parameters the resulting adaptive MCMC algorithm might actually converge slower against the target distribution than a simple non-adaptive MCMC algorithm. However, as already indicated in Section 7.2, simulation studies show that the choices $r_j^{(0)} = \frac{q}{p}$ with $q \in [2, 20]$, $L_j = p$ for all $j \in \mathcal{P}$ as well as $\epsilon = 10^{-6}$ lead to a well-mixing algorithm in most situations (see also Sections 7.5 and 7.6). It would be an interesting topic for future research to derive theoretical results concerning the mixing time of MAdaSub, i.e. the "speed of convergence" of MAdaSub against the target distribution, in terms of the choices of its tuning parameters.

7.4. Comparison to existing methodology

In this section we conceptually compare the proposed MAdaSub algorithm with other approaches for high-dimensional Bayesian variable selection. Here, we mainly focus on adaptive MCMC algorithms that are most closely related to the new algorithm.

In a pioneering work, Nott and Kohn (2005) propose an adaptive sampling algorithm for Bayesian variable selection: After an initial period of applying a non-adaptive MCMC algorithm in order to obtain initial estimates of the posterior inclusion probabilities and of the corresponding posterior covariances, in the second phase of the algorithm a version of a Metropolized Gibbs sampler is used where the sampling probability of a variable X_j is given by the best current linear approximation to the posterior inclusion probability of X_j. During the second phase, the current estimated posterior inclusion probabilities and the posterior covariances are sequentially adapted. Nott and Kohn (2005) empirically show that their adaptive MCMC algorithm outperforms different non-adaptive algorithms in terms of efficiency per iteration. However, since their approach requires the computation of the inverses of estimated covariance matrices, it does not scale well to high-dimensional settings with many explanatory variables p. Furthermore, the Gibbs sampling nature of the algorithm leads to a principally local exploration of the model space, which might not be optimal in a situation where multiple "distant" modes of the posterior exist.

Certain variants and extensions of the adaptive algorithm of Nott and Kohn (2005) have been proposed, like an adaptive Metropolis-Hastings algorithm by Lamnisos et al. (2013),

where the expected number of variables to be changed by the proposal is adapted during the algorithm. Furthermore, different individual adaptation algorithms have been developed in Griffin et al. (2014) as well as in the corresponding follow-up work Griffin et al. (2017). In order to describe the differences between these methods and the MAdaSub algorithm, we summarize the two variants of individual adaptation proposed in Griffin et al. (2017) in some more detail. Both strategies are based on an adaptive Metropolis-Hastings algorithm where the adaptation of the proposal distributions aims at achieving a desirable pre-specified acceptance rate $\tau \in (0,1)$. The proposal distributions are of the following form: If $S \in \mathcal{M}_{\text{full}}$ is the current model, then the probability of proposing $V \in \mathcal{M}_{\text{full}}$ is given by

$$\tilde{q}(V \mid S; \boldsymbol{\eta}) = \prod_{j \in V \setminus S} A_j \prod_{j \in S \setminus V} D_j \prod_{j \in \mathcal{P} \setminus (S \cap V)} (1 - A_j) \prod_{j \in S \cap V} (1 - D_j)\,, \tag{7.31}$$

where $\boldsymbol{\eta} = (\boldsymbol{A}^T, \boldsymbol{D}^T)^T = (A_1, \ldots, A_p, D_1, \ldots, D_p)^T \in (0,1)^{2p}$ is a vector of tuning parameters with the following interpretation: For $j \in \mathcal{P}$, A_j is the probability of adding variable X_j if it is not included in the current model S and D_j is the probability of deleting variable X_j if it is included in the current model S. Note that this proposal has also an independent Bernoulli form and corresponds to the independence proposal in (7.7) when setting $r_j = A_j = 1 - D_j$ for all $j \in \mathcal{P}$. However, in general the proposal is fundamentally different from the independence proposal in (7.7) since the proposed model V is generally **not** independent from the current model S.

The first strategy in Griffin et al. (2017) is called "exploratory individual adaptation" and adapts each of the $2p$ tuning parameters $\boldsymbol{\eta}$ after iteration t via

$$\operatorname{logit}_\epsilon(A_j^{(t)}) = \operatorname{logit}_\epsilon(A_j^{(t-1)}) + \mathbb{1}_{V^{(t)} \setminus S^{(t-1)}}(j) \times \phi^{(t-1)} \left(\tilde{\alpha}^{(t)} - \tau\right), \quad j \in \mathcal{P}\,, \tag{7.32}$$

and

$$\operatorname{logit}_\epsilon(D_j^{(t)}) = \operatorname{logit}_\epsilon(D_j^{(t-1)}) + \mathbb{1}_{S^{(t-1)} \setminus V^{(t)}}(j) \times \phi^{(t-1)} \left(\tilde{\alpha}^{(t)} - \tau\right), \quad j \in \mathcal{P}\,. \tag{7.33}$$

Here, $\operatorname{logit}_\epsilon : (\epsilon, 1-\epsilon) \to \mathbb{R}$ is defined by $\operatorname{logit}_\epsilon(x) = \log((x-\epsilon)/(1-x-\epsilon))$ for $x \in (\epsilon, 1-\epsilon)$, where $\epsilon \in (0, 0.5)$ is an additional constant which ensures that $A_j^{(t)}, D_j^{(t)} \in (\epsilon, 1-\epsilon)$ for all iterations t (similar as in Algorithm 7.1). Furthermore,

$$\tilde{\alpha}^{(t)} = \min\left\{ \frac{C(V^{(t)})\, \tilde{q}(S^{(t-1)} \mid V^{(t)}; \boldsymbol{\eta}^{(t-1)})}{C(S^{(t-1)})\, \tilde{q}(V^{(t)} \mid S^{(t-1)}; \boldsymbol{\eta}^{(t-1)})}, 1 \right\} \tag{7.34}$$

denotes the corresponding acceptance probability in iteration t. The parameter $\phi^{(t)}$ controls the "adaptation rate" and is assumed to decay sufficiently fast, i.e. $\phi^{(t)} = \mathcal{O}(1/t^{\lambda})$ with some constant $0.5 < \lambda \leq 1$.

Note that "exploratory individual adaptation" requires to update $2p$ tuning parameters after each iteration, which can be computationally expensive if p is large. Therefore, the second strategy proposed in Griffin et al. (2017), which is called "adaptive scale individual adaptation", adapts only one tuning parameter $\zeta \in [0, 1]$ after iteration t via

$$\text{logit}_{\epsilon}(\zeta^{(t)}) = \text{logit}_{\epsilon}(\zeta^{(t-1)}) + \phi^{(t-1)} \left(\tilde{\alpha}^{(t)} - \tau \right) , \tag{7.35}$$

where $\phi^{(t-1)}$ and τ are specified as above. The corresponding proposal in iteration $t+1$ is again given by equation (7.31) with $\boldsymbol{\eta}^{(t)} = (A_1^{(t)}, \ldots, A_p^{(t)}, D_1^{(t)}, \ldots, D_p^{(t)})^T$ defined by

$$A_j^{(t)} = \zeta^{(t)} \min \left\{ \frac{\hat{\pi}_j^{(t)}}{1 - \hat{\pi}_j^{(t)}}, 1 \right\}, \quad D_j^{(t)} = \zeta^{(t)} \min \left\{ \frac{1 - \hat{\pi}_j^{(t)}}{\hat{\pi}_j^{(t)}}, 1 \right\}, \quad j \in \mathcal{P} , \tag{7.36}$$

where $\hat{\pi}_j^{(t)}$ is a Rao-Blackwellized estimate (see Ghosh and Clyde, 2011) of the posterior inclusion probability of variable X_j after iteration t.

Griffin et al. (2017) motivate their choice of the proposal parameters $A_j^{(t)}$ and $D_j^{(t)}$ by considering the idealized situation of a target distribution which itself has an independent Bernoulli form of (7.7) and showing that, in such a situation, the expected squared jumping distance (compare Pasarica and Gelman, 2010) of the resulting (non-adaptive) MCMC algorithm is maximized when the true marginal probabilities π_j are known and used as (fixed) estimates $\hat{\pi}_j^{(t)} = \pi_j$ for $j \in \mathcal{P}$ in (7.36), as well as $\zeta^{(t)} = 1$. Furthermore, they argue that this proposal leads to a small asymptotic variance when estimating linear transformations of marginal quantities (like posterior inclusion probabilities). They thus conclude that the proposal (7.31) with $A_j^{(t)}$ and $D_j^{(t)}$ as in (7.36) should be preferred over an independence sampler of the form (7.7) (which is used in the MAdaSub algorithm).

This argument may not be applicable in general, since the definition (7.36) of $A_j^{(t)}$ and $D_j^{(t)}$ can lead to large negative autocorrelations between succeeding samples of the resulting chain: If $\hat{\pi}_j^{(t)} \approx 0.5$ for some $j \in \mathcal{P}$ and $\zeta^{(t)} \approx 1$, then $A_j^{(t)} \approx D_j^{(t)} \approx 1$, i.e. with large probability the chain will "oscillate" between models including the variable X_j and excluding the variable X_j (note that if $\hat{\pi}_j^{(t)} = 0.5$ we might actually loose the aperiodicity of the sampler). Of course, this behaviour of the algorithm is beneficial for estimating the marginal inclusion

probabilities π_j of the target, but only given that the estimated inclusion probabilities are already close to the true ones ($\hat{\pi}_j^{(t)} \approx \pi_j$) and, even more importantly, given that the target has an independent Bernoulli form. However, posterior model distributions will rarely have an exact independent Bernoulli form and thus the proposed algorithm based on $A_j^{(t)}$ and $D_j^{(t)}$ as in (7.36) might have problems to quickly explore the full posterior model distribution. We want to illustrate this point with a simple toy example.

Example 7.1. Let $\mathcal{P} = \{1, 2\}$ and suppose that the target π is given by $\pi(\emptyset) = 0.5 - 2\delta$, $\pi(\{1\}) = \pi(\{2\}) = \delta$ and $\pi(\{1, 2\}) = 0.5$ for some small constant $\delta \in (0, 0.25)$. Then the true marginal inclusion probabilities are $\pi_1 = 0.5 + \delta$ and $\pi_2 = 0.5 + \delta$. Suppose that we use the non-adaptive MCMC algorithm based on the proposal (7.31) with $A_j^{(t)}$ and $D_j^{(t)}$ as in (7.36) and $\hat{\pi}_j^{(t)} = \pi_j$ for $j = 1, 2$ and $t \in \mathbb{N}$. Then, for $\delta \approx 0$ and $\zeta^{(t)} \approx 1$, we have $A_j^{(t)} \approx 1$ and $D_j^{(t)} \approx 1$. Now if we start the chain with $S^{(0)} = \{1\}$, then with large probability we will oscillate between the two models with only a single variable (i.e. $S^{(1)} = \{2\}$, $S^{(2)} = \{1\}$, $S^{(3)} = \{2\}$, ...) and for a long time we will not be able to visit the other two models which have the largest probability. On the other hand, the independent Bernoulli proposal (7.7) (used in MAdaSub) with $r_j = \pi_j \approx 0.5$ for $j = 1, 2$, has probability $(1 - r_1)(1 - r_2) \approx 0.25$ of proposing the "null" model $V = \emptyset$ and probability $r_1 r_2 \approx 0.25$ of proposing the "full" model $V = \{1, 2\}$. Thus, we avoid being stuck at models with low target probability.

Another related adaptive method for Bayesian variable selection has been proposed by Ji and Schmidler (2013). They consider an adaptive independence Metropolis-Hastings algorithm for sampling directly from the posterior distribution of the regression coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$, assuming that the prior of β_j for $j \in \mathcal{P}$ is given by a mixture of a point-mass at zero (indicating that the corresponding variable X_j is not included in the model) and a continuous normal distribution (indicating that variable X_j is "relevant"). Mixtures of normal distributions are used as proposals in the Metropolis-Hastings step, which are adapted during the algorithm in order to minimize the Kullback-Leibler divergence from the target distribution. The considered family of mixture distributions should ideally have sufficiently many mixture components in order to be able to approximate the generally highly multimodal posterior distribution of $\boldsymbol{\beta}$. In comparison, MAdaSub focuses on sampling from the discrete model distribution and makes use of independent Bernoulli distributions as approximations to the targeted posterior model distribution, while the updating scheme is motivated

in a Bayesian way. It would certainly be interesting to combine Algorithm 7.1 and the approach of Ji and Schmidler (2013) in order to address non-conjugate and non-approximate situations in a similar way.

Clyde et al. (2011) propose a "Bayesian adaptive sampling" (BAS) algorithm which is based on sampling without replacement from the posterior model distribution, where the individual sampling probabilities of the variables are adapted during the algorithm such that they converge against the posterior marginal inclusion probabilities. By construction of BAS, if the number of iterations is equal to the number of possible models, the algorithm enumerates all possible models. However, since BAS samples without replacement, it has to be ensured that no model is sampled twice and therefore, after each iteration of the algorithm, the sampling probabilities of some of the remaining models have to be renormalized which can be computationally expensive.

Shotgun stochastic search (SSS, Hans et al., 2007) is a (non-adaptive) stochastic optimization algorithm which aims at a fast identification of models with large posterior probability based on local moves of three different types: "Addition" (add one variable to the current model), "replacement" (replace one variable in the current model with another variable not in the current model) and "deletion" (delete one variable from the current model). Evolutionary stochastic search (ESS, Bottolo and Richardson, 2010) is an evolutionary Monte Carlo algorithm (Liang and Wong, 2000) which makes use of a parallel tempering strategy, running multiple Markov chains in parallel with different associated "temperatures", similar as in the simulated annealing approach for optimization (Kirkpatrick et al., 1983). The algorithm combines local moves (based on adaptive Gibbs-like updates within the different chains) with global moves (based on partial or full swaps of the current states between different chains) in order to avoid being stuck in local modes of the posterior. Bottolo et al. (2011) provide a computationally optimized implementation of ESS in C++, which can be used through the R-package `R2GUESS`. A major difference of MAdaSub in comparison to ESS is that it operates on a single Markov chain only, without loosing the ability to make "good" global moves. Furthermore, Algorithm 7.1 is apparently much easier to understand and implement.

Finally, Schäfer and Chopin (2013) propose sequential Monte Carlo algorithms (see also Del Moral et al., 2006) using model proposals which are not of an independent Bernoulli

form and thus can take correlations between different explanatory variables into account. In contrast, recall that MAdaSub is an adaptive MCMC algorithm which is based on independent Bernoulli proposals. While similar extensions of MAdaSub (i.e. including proposal distributions which allow for dependencies between different variables) might be desirable in order to better approximate the target posterior model distribution, this may come at the prize of a significantly larger computational cost for updating and sampling from the proposal. Furthermore, results from simulation studies indicate that even the simple proposals of independent Bernoulli form result in a well-mixing algorithm with sufficiently large acceptance probabilities in most high-dimensional but sparse situations.

7.5. Simulated data examples

In this section we want to illustrate the performance of MAdaSub via simulated data examples in the setting of normal linear models. In Section 7.5.1 we consider a relatively low-dimensional situation with $p = 20$ explanatory variables, so that it is computationally feasible to derive the exact posterior model distribution via a full model enumeration and to compare it with the estimates obtained from MAdaSub. In Section 7.5.2 we consider a simulated data example with $p = 1000$ explanatory variables and demonstrate that the MAdaSub algorithm provides stable estimates of posterior marginal inclusion probabilities even in high-dimensional situations.

While we have investigated the performance of the MAdaSub algorithm in various different scenarios, an extensive simulation study comparing the effectiveness of the different adaptive and non-adaptive MCMC algorithms described in Section 7.4 is beyond the scope of this thesis and will be an interesting topic for future work.

7.5.1. Low-dimensional setting

In this section our main aim is to show that the MAdaSub algorithm provides an efficient way to sample from the correct target distribution in a relatively low-dimensional setting. For this, we consider exactly the same simulation setup as described in Section 4.2 for the illustration of the AdaSub method, but with $p = 20$ (instead of $p = 1000$) explanatory

variables. In particular, the sample size is given by $n = 60$ and

$$\boldsymbol{\beta}_0 = (0.4, 0.8, 1.2, 1.6, 2.0, 0, \ldots, 0)^T \in \mathbb{R}^p$$

is the true vector of regression coefficients with active set $S_0 = \{1, \ldots, 5\}$. We employ the g-prior with $g = n$ as given in equation (7.4) of Section 7.1. Furthermore, we make use of an independent Bernoulli model prior with inclusion probability of $\omega = 0.5$ (see equation (7.5) of Section 7.1), resulting in a uniform prior over the model space.

In the MAdaSub algorithm we set $r_j^{(0)} = \frac{1}{2}$ for all $j \in \mathcal{P}$, i.e. we use the prior marginal inclusion probabilities as the initial sampling probabilities in MAdaSub. We first consider the choice $L_j = p$ (for $j \in \mathcal{P}$) for the variance parameters of MAdaSub, corresponding to the update in equation (7.15). Furthermore, we set $\epsilon = 10^{-6}$ and run the MAdaSub algorithm for $T = 50,000$ iterations.

In order to compare the results of MAdaSub with the true posterior model distribution, we also conduct a full model enumeration using the Bayesian Adaptive Sampling (BAS) algorithm (compare Section 7.4) which is available in the R-package `BAS` (Clyde, 2017). By this, we obtain the exact posterior inclusion probabilities $\pi(j \in S \,|\, \mathcal{D})$ for each $j \in \mathcal{P}$.

For illustrating the efficiency of MAdaSub, we compare it with independent Metropolis-Hastings algorithms where the individual selection probabilities are **not** adapted during the algorithm, i.e. we set $r_j^{(t)} = r_j^{(0)}$ for all $t \in \mathbb{N}$ and $j \in \mathcal{P}$. In particular, we consider the choice $r_j^{(t)} = r_j^{(0)} = 0.5$, corresponding to the initial proposal distribution in MAdaSub, and the choice $r_j^{(t)} = r_j^{(0)} = \pi(j \in S \,|\, \mathcal{D})$, corresponding to the targeted proposal distribution, which is, as stated above, the closest independent Bernoulli proposal to the target $\pi(\cdot \,|\, \mathcal{D})$ in terms of Kullback-Leibler divergence (Clyde et al., 2011).

Figure 7.1 shows the evolution of the values of the targeted (log-)kernel

$$\log(C(S^{(t)})) = \log(\pi(\boldsymbol{y} \,|\, \boldsymbol{X}, S^{(t)})) + \log(\pi(S^{(t)})) \tag{7.37}$$

for the sampled models $S^{(t)}$ along the iterations t of MAdaSub as well as for the non-adaptive independent Metropolis-Hastings samplers with prior and posterior marginal inclusion probabilities as (fixed) proposal sampling probabilities, respectively. Furthermore, Figure 7.2 depicts the sizes $|V^{(t)}|$ of the proposed models and the sizes $|S^{(t)}|$ of the sampled models, while Figure 7.3 shows the evolution of the acceptance rates along the iterations t.

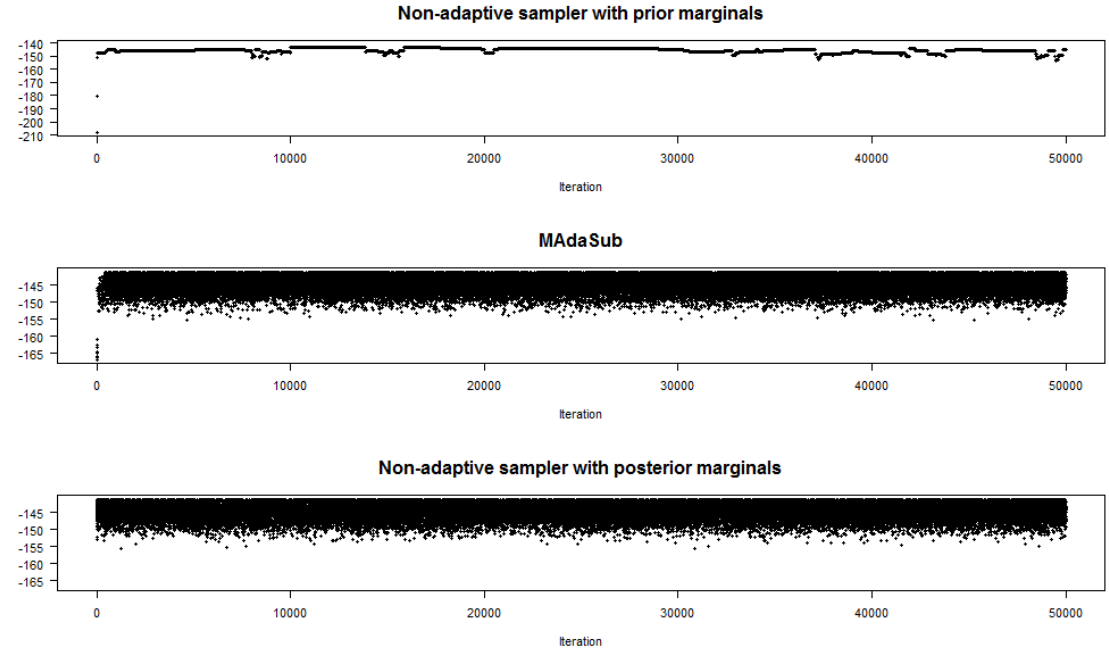


Figure 7.1.: Low-dimensional example with g-prior: Evolution of $\log(C(S^{(t)}))$ along the iterations (t) for non-adaptive sampler with prior marginals as selection probabilities, for MAdaSub (with $L_j = p$) and for non-adaptive sampler with posterior marginals as selection probabilities.

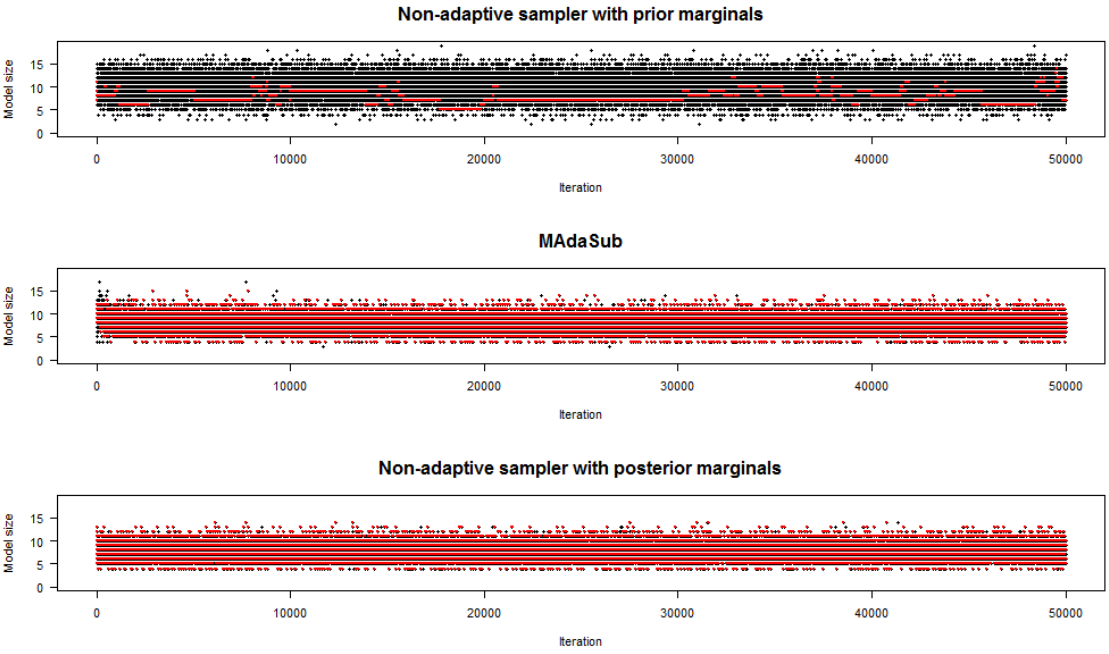


Figure 7.2.: Low-dimensional example with g-prior: Evolution of the sizes $|V^{(t)}|$ of the proposed models (black) and of the sizes $|S^{(t)}|$ of the sampled models (red) along the iterations (t) for non-adaptive sampler with prior marginals as selection probabilities, for MAdaSub (with $L_j = p$) and for non-adaptive sampler with posterior marginals as selection probabilities.

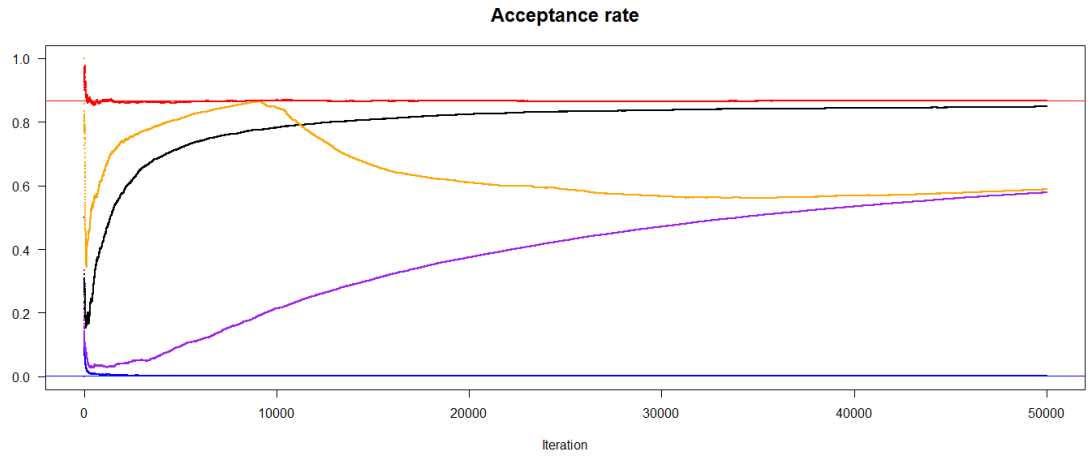


Figure 7.3.: Low-dimensional example with g-prior: Evolution of the acceptance rates along the iterations for non-adaptive sampler with prior marginals as selection probabilities (blue) and for non-adaptive sampler with posterior marginals as selection probabilities (red), as well as for MAdaSub with $L_j = p$ (black), $L_j = p/n$ (orange) and $L_j = 100p$ (purple) for $j \in \mathcal{P}$.

As might have been expected, it is apparent that the non-adaptive sampler with prior marginals as selection probabilities performs poorly with a very slow exploration of the model space and a small acceptance rate which remains close to zero. On the other hand, the non-adaptive sampler with posterior marginals as selection probabilities leads to fast mixing with corresponding acceptance rate of approximately 0.87. Even though the MAdaSub algorithm starts with exactly the same "initial configuration" as the non-adaptive sampler with prior marginals, it quickly adjusts the selection probabilities accordingly, so that the resulting acceptance rate approaches the "target" value of 0.87 from the non-adaptive sampler with posterior marginals. In particular, when inspecting the evolutions of the sampled models through Figures 7.1 and 7.2, the MAdaSub algorithm is very difficult to distinguish from the sampler with posterior marginals (after a very short "burn-in" period).

In order to illustrate the behaviour of the MAdaSub algorithm with respect to the choice of the variance parameters L_j, additionally to the choice $L_j = p$ we have conducted two further runs of MAdaSub with the same specifications as before but with $L_j = p/n$ (corresponding to the learning rate $K = n$ in AdaSub) and with $L_j = 100p$, respectively. Figure 7.3 indicates that the choice $L_j = p$ is favourable, yielding a fast and "sustainable" increase of the acceptance rate. On the other hand, for $L_j = 100p$ the sampling probabilities in MAdaSub are slowly adapted, while for $L_j = p/n$ the sampling probabilities are adapted very quickly by the algorithm resulting in initially large acceptance rates; however, this increase is only

due to a premature focus of the proposal on certain parts of the model space and thus the acceptance rate significantly decreases at some point when the algorithm "identifies" other areas of high posterior probability that have not been covered by the proposal. Interestingly, this example confirms the general observation that the optimal "speed" of adaptation seems to be slower for MAdaSub in comparison to the AdaSub method (see the discussion in Remark 7.1 of Section 7.2).

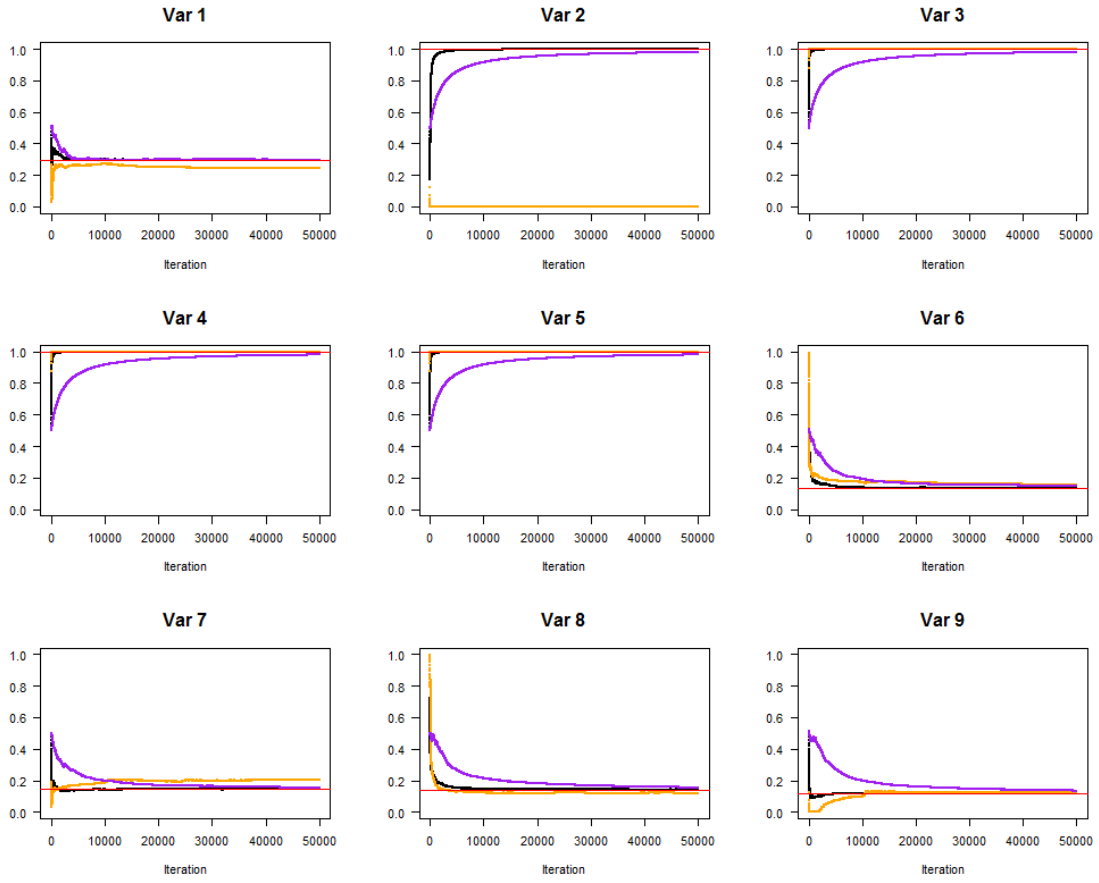


Figure 7.4.: Low-dimensional example with g-prior: Evolution of the selection probabilities $r_j^{(t)}$, for $j = 1, \ldots, 9$, along the iterations (t) of MAdaSub with $L_j = p$ (black), $L_j = p/n$ (orange) and $L_j = 100p$ (purple) for $j \in \mathcal{P}$. The red horizontal lines indicate the true posterior inclusion probabilities.

Figure 7.4 shows the evolution of the selection probabilities $r_j^{(t)}$ corresponding to variables X_j, with $j = 1, \ldots, 9$, along the iterations t of MAdaSub for the three different choices of the variance parameter L_j. For $L_j = p$ the sampling probabilities quickly converge to the correct posterior inclusion probabilities. For $L_j = 100p$, the sampling probabilities also "converge correctly", though at a significantly slower pace. For $L_j = p/n$, the algorithm "converges" very slowly against the correct posterior distribution, as for example the sampling probability

$r_2^{(t)}$ is still very far away from the correct posterior inclusion probability of variable X_2.

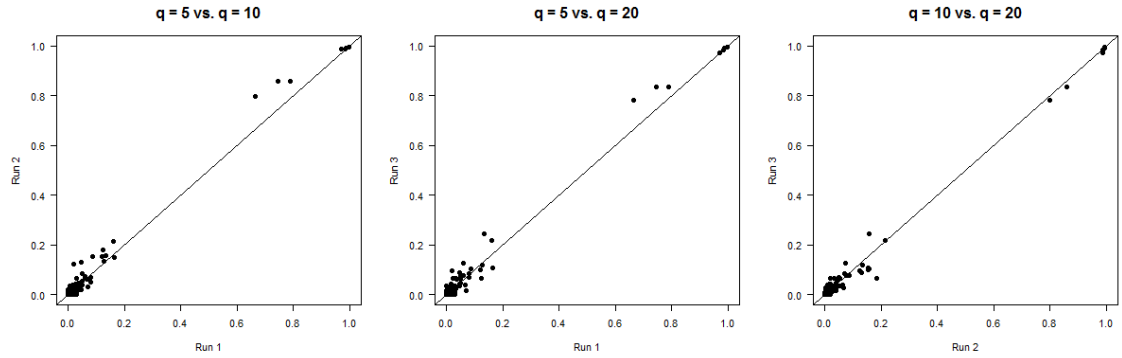
Note that this illustrative example strongly supports the argumentation in Remark 7.2 of Section 7.3: Despite the ergodicity of the MAdaSub algorithm for all choices of its tuning parameters, the "speed of convergence" against the target distribution crucially depends on a proper choice of these parameters. Although a more detailed discussion of the influence of the tuning parameters on the mixing behaviour of the algorithm is desirable, we observe that the choices $L_j = p$ and $r_j^{(0)} = \frac{q}{p}$ with $q \in [2, 20]$ work well for most sparse situations in practice. Similar as for the AdaSub algorithm, we note that the choice of the initial selection probabilities $r_j^{(0)}$ seems not to be as crucial as the choice of the variance parameters L_j. In the subsequent section we will further illustrate this point in the context of a high-dimensional simulated data example.

7.5.2. High-dimensional setting

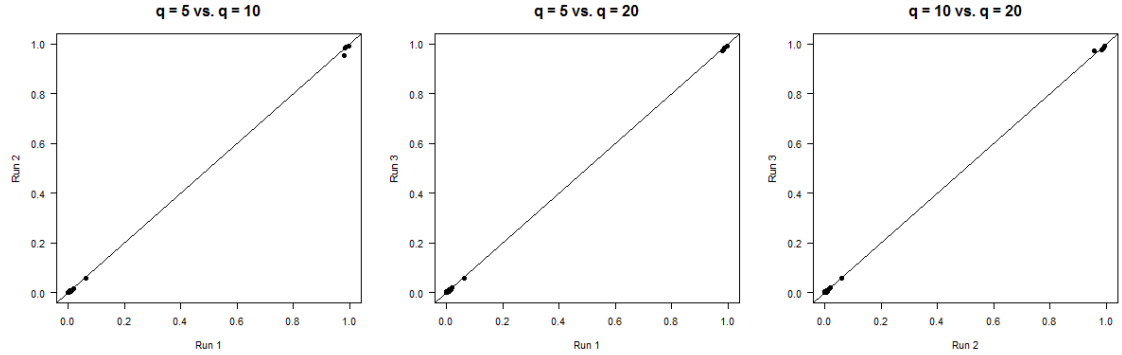
In order to investigate the performance of MAdaSub in a high-dimensional setting, we consider the same simulated dataset as in Section 4.2 with $p = 1000$ explanatory variables. Recall that even though the explanatory variables are generated independently, due to the small sample size of $n = 60$, the pairwise empirical correlations between the variables are quite diverse (between -0.45 and 0.45). However, the detailed investigation of MAdaSub in the presence of severe multicollinearity will be an important topic for further research.

In Section 4.2 we have applied the AdaSub method on this particular dataset in combination with the EBIC_γ as the selection criterion, using $\gamma = 0.6$ and $\gamma = 1$. For comparison reasons, we consider the corresponding Bayesian setting where the kernel of the posterior model distribution is given by $\pi(S \mid \mathcal{D}) \propto \exp\{-\frac{1}{2}\,\text{EBIC}_\gamma(S)\}$ (see equation (7.6) of Section 7.1). In the MAdaSub algorithm we set $L_j = p$ for $j \in \mathcal{P}$ and $\epsilon = 10^{-6}$. We conduct three different runs of MAdaSub, each with $T = 200,000$ iterations, setting $r_j^{(0)} = \frac{q}{p}$ as the initial selection probabilities with different expected search sizes q: For the first run we set $q = 5$, for the second run $q = 10$ and for the third run $q = 20$. On a general note, the proper choice of a sufficiently large number of iterations T, yielding a decent convergence behaviour of MAdaSub, depends largely on the considered setting and should be based on the inspection of appropriate "diagnostic plots" (see below). Although we have not aimed for the most efficient implementation, the computation time for each of the different runs

with $T = 200,000$ iterations is only between five and six minutes on a 3.2-GHz processor.



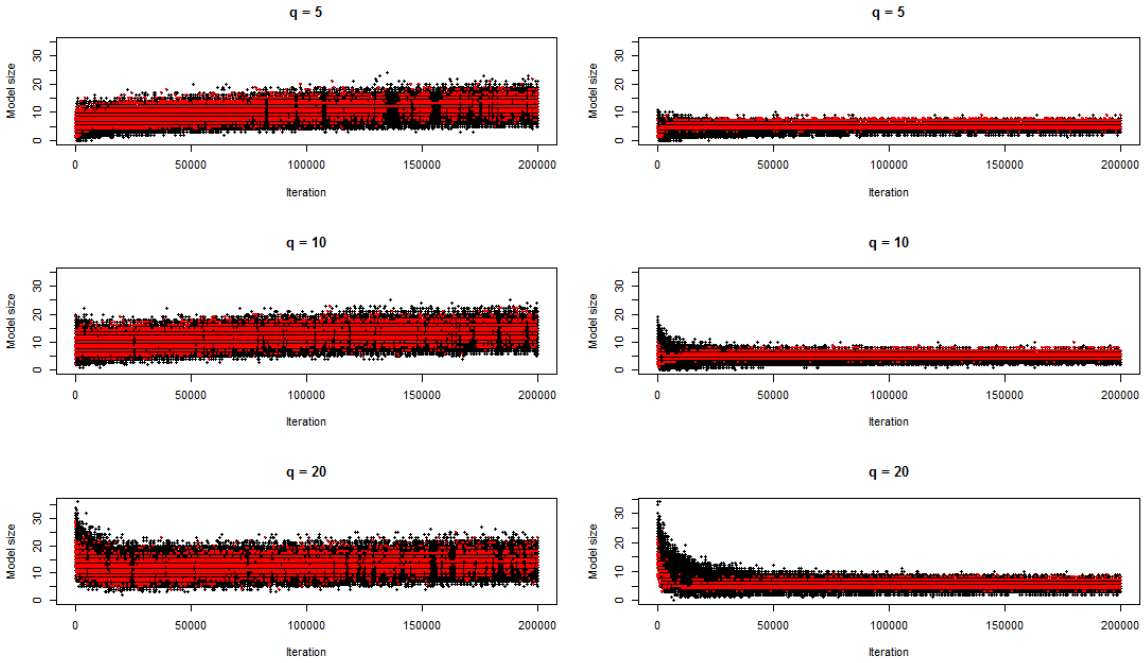
(a) EBIC_γ with $\gamma = 0.6$



(b) EBIC_γ with $\gamma = 1$

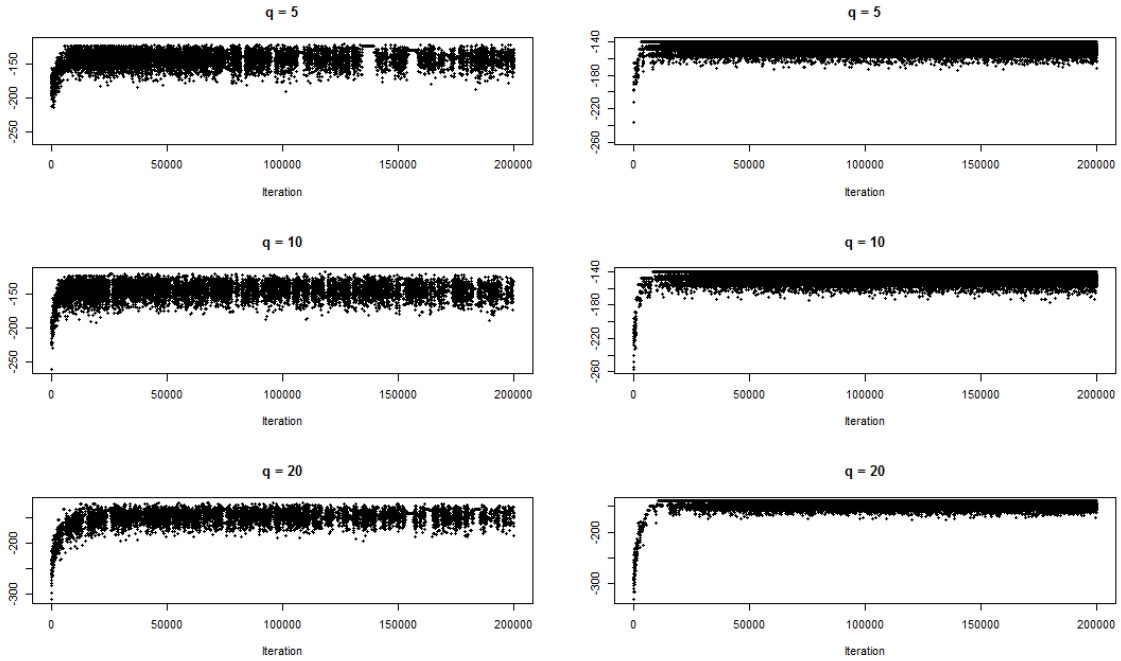
Figure 7.5.: High-dimensional example: Scatterplots of estimated posterior inclusion probabilities for the different runs of MAdaSub ($q = 5$, $q = 10$ and $q = 20$).

Figure 7.5 shows scatterplots of the estimated posterior marginal inclusion probabilities for the three different runs of the MAdaSub algorithm. For the situation with $\gamma = 1$, the estimates are very stable across the different runs of the algorithm. For $\gamma = 0.6$, the respective estimates are similar but show larger variability, which is due to the fact that this corresponds to a more challenging situation where the model prior underlying the criterion enforces less sparsity. This observation is confirmed by Figure 7.6 and Figure 7.7, which depict the evolutions of the sizes of the sampled and proposed models as well as the log-values of the targeted kernel: For $\gamma = 1$, most of the sampled models are of small size, while for $\gamma = 0.6$, the sampled models tend to be larger. Note that for the different initial search sizes $q \in \{5, 10, 20\}$, the sizes of the proposed models are adapted accordingly along the number of the iterations of the MAdaSub algorithm.



(a) EBIC$_\gamma$ with $\gamma = 0.6$ (b) EBIC$_\gamma$ with $\gamma = 1$

Figure 7.6.: High-dimensional example: Evolution of the sizes $|V^{(t)}|$ of the proposed models (black) and of the sizes $|S^{(t)}|$ of the sampled models (red) along the iterations (t) of MAdaSub for $q = 5$, $q = 10$ and $q = 20$.



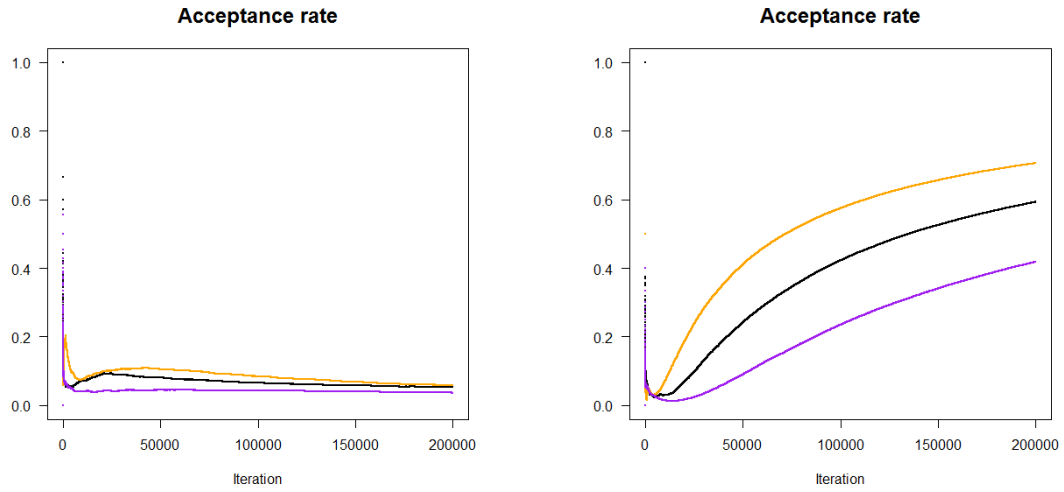
(a) EBIC$_\gamma$ with $\gamma = 0.6$ (b) EBIC$_\gamma$ with $\gamma = 1$

Figure 7.7.: High-dimensional example: Evolution of $\log(C(S^{(t)}))$ along the iterations (t) of MAdaSub for $q = 5$, $q = 10$ and $q = 20$.

For the situation with $\gamma = 0.6$, the estimated median probability model from MAdaSub is given by

$$\hat{S}_{\mathrm{MP}} = \left\{ j \in \mathcal{P};\, r_j^{(T)} \geq 0.5 \right\} = \{2, 3, 4, 5, 519, 731, 950\}\,,$$

while for $\gamma = 1$, it is given by $\hat{S}_{\mathrm{MP}} = \{2, 3, 4, 5\}$, which is closer to the true underlying model $S_0 = \{1, \ldots, 5\}$. This indicates that the prior corresponding to the EBIC_γ with $\gamma = 1$ is favourable for the given situation of a sparse underlying truth. Recall the results from Section 4.2 where the AdaSub method has been applied on the same dataset: For $\gamma = 1$, the "best" and thresholded model (with threshold $\rho = 0.9$) selected by AdaSub are both given by $\hat{S}_{0.9} = \hat{S}_{\mathrm{b}} = \{2, 3, 4, 5\}$, which also agrees with the median probability model estimated by MAdaSub. However, the case $\gamma = 0.6$ corresponds to an "unstable" situation where the "best" model and the thresholded model of AdaSub do not agree; the "best" model from AdaSub is given by $\hat{S}_{\mathrm{b}} = \{2, 3, 4, 5, 519, 731, 950\}$, which coincides with the median probability model from MAdaSub, while the thresholded model is given by $\hat{S}_{0.9} = \{2, 3, 4, 5, 950\}$, which is again closer to the true underlying model S_0.



(a) EBIC_γ with $\gamma = 0.6$ (b) EBIC_γ with $\gamma = 1$

Figure 7.8.: High-dimensional example: Evolution of the acceptance rates along the iterations of MAdaSub for $q = 5$ (orange), $q = 10$ (black) and $q = 20$ (purple).

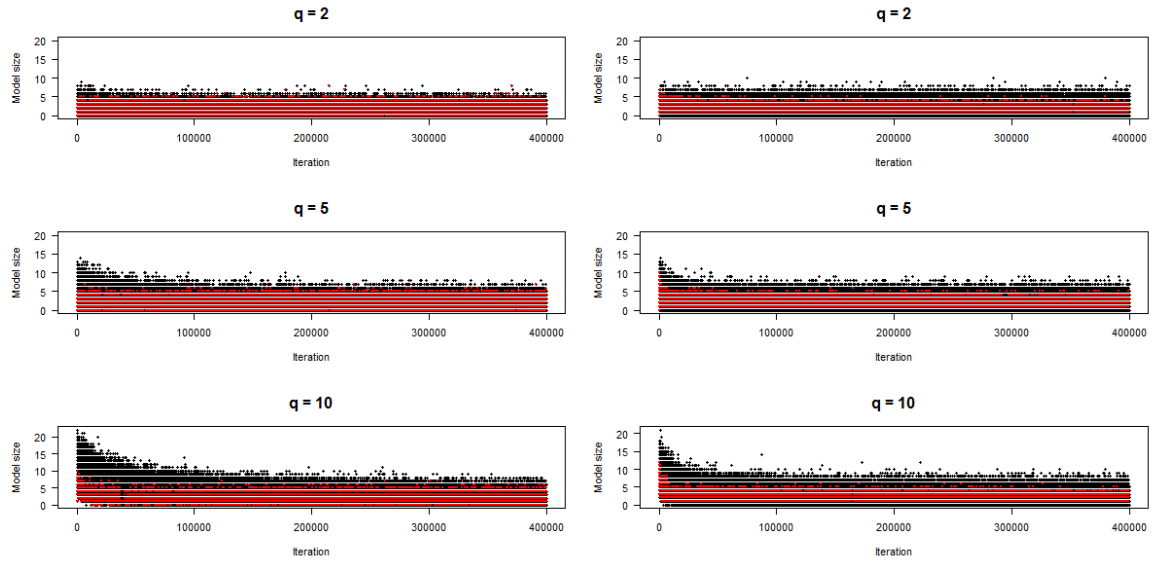
Finally, Figure 7.8 shows the evolution of the acceptance rates: For $\gamma = 1$, the acceptance rates quickly increase along the number of iterations with the fastest increase for $q = 5$, yielding a final acceptance rate of approximately 0.71. However, for $\gamma = 0.6$, the acceptance rates remain relatively small and are below 10% for all runs of MAdaSub, indicating slower

mixing of the algorithm. Interestingly, one may observe that the statistically less "favourable" situation corresponding to the EBIC_γ with $\gamma = 0.6$ does also lead to a slower convergence behaviour of the MAdaSub algorithm. In this light, we also refer to the recent work of Yang et al. (2016), which suggests that contraction properties of the posterior distribution (i.e. posterior concentration around the "true" generating model) come with both desirable statistical properties and faster mixing MCMC algorithms.

7.6. Real data examples

In this section we illustrate the efficiency of the MAdaSub algorithm when applied on high-dimensional real data. In particular, we consider two datasets which have already been analysed by AdaSub and BackAdaSub in previous chapters of this thesis: First, the polymerase chain reaction (PCR) dataset of Lan et al. (2006) with $p = 22,575$ explanatory variables, sample size $n = 60$ and continuous response data (see Section 5.2.2 for details and the analysis with AdaSub) and second, the leukemia dataset of Golub et al. (1999) with $p = 6817$, $n = 72$ and binary response data (see Section 6.4.2 for details and the analysis with BackAdaSub). For the PCR dataset we face the problem of variable selection in a linear regression framework, while for the leukemia dataset we consider variable selection in a logistic regression framework. Since the EBIC has already been employed as a selection criterion for the analysis of these datasets via AdaSub and BackAdaSub, for comparison reasons we consider the corresponding Bayesian setting where the kernel of the posterior model distribution is given by $\pi(S \,|\, \mathcal{D}) \propto \exp\{-\frac{1}{2}\,\text{EBIC}_\gamma(S)\}$ (see equation (7.6) of Section 7.1). Here, we focus on the choice $\gamma = 1$ in EBIC_γ, which enforces more sparsity than the choice $\gamma = 0.6$.

We make use of the preprocessed datasets as described in Sections 5.2.2 and 6.4.2. In the MAdaSub algorithm we set $L_j = p$ for $j \in \mathcal{P}$ and $\epsilon = 10^{-6}$. Similar to the analysis of the simulated data example in Section 7.5.2, we conduct three different runs of MAdaSub, each with $T = 400,000$ iterations, setting $r_j^{(0)} = \frac{q}{p}$ as initial selection probabilities with different expected search sizes q: For the first run we set $q = 2$, for the second run $q = 5$ and for the third run $q = 10$. The computation time for a single run of MAdaSub on a 3.2-GHz processor are between 43 and 48 minutes for the PCR dataset and approximately 25 minutes for the leukemia dataset.

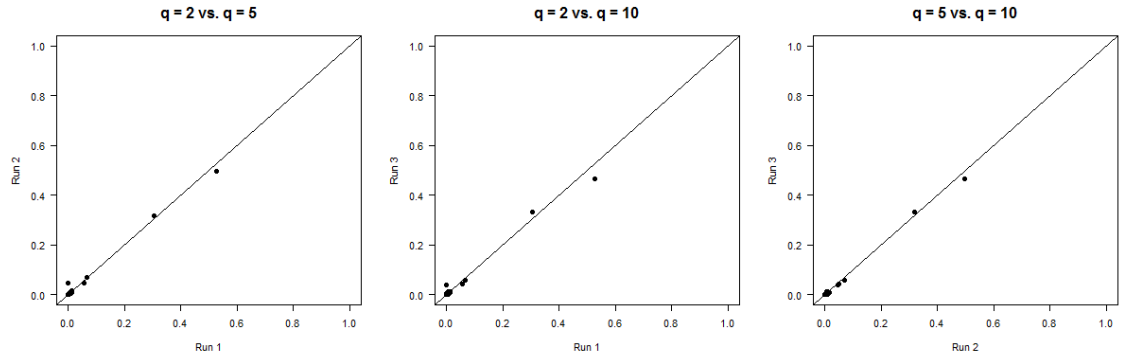


(a) PCR dataset (b) Leukemia dataset

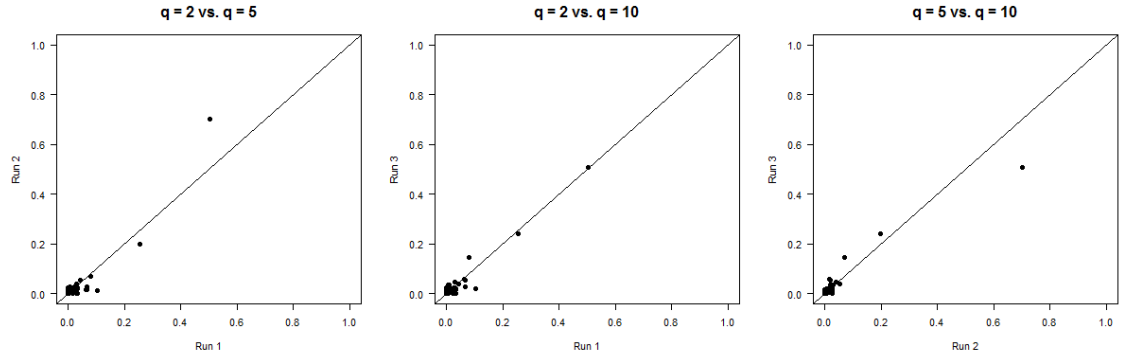
Figure 7.9.: Real data examples: Evolution of the sizes $|V^{(t)}|$ of the proposed models (black) and of the sizes $|S^{(t)}|$ of the sampled models (red) along the iterations (t) of MAdaSub for $q = 2$, $q = 5$ and $q = 10$.

Figure 7.9 depicts the evolution of the sizes of the sampled and proposed models, while Figure 7.10 shows scatterplots of the estimated posterior inclusion probabilities for the different runs of MAdaSub. It is apparent that for both datasets the MAdaSub algorithm provides quite stable estimates of the posterior marginal inclusion probabilities despite the (ultra-)high-dimensional model space, despite the different initial search sizes q and despite the moderate number of iterations $T = 400,000$ (in relation to the huge number of possible models). The acceptance rates of MAdaSub for the PCR dataset are between 0.14 (for $q = 10$) and 0.33 (for $q = 2$), while the acceptance rates for the leukemia dataset are between 0.06 (for $q = 10$) and 0.13 (for $q = 5$). The smaller acceptance rates for the leukemia dataset indicate that this corresponds to a more challenging scenario (i.e. the targeted posterior model distribution seems to be "further away" from an independent Bernoulli form). This observation is also reflected in the larger variability of the estimates of the posterior marginal inclusion probabilities for the leukemia dataset.

Note that due to the very large model spaces in both considered examples, posterior probabilities of individual models are generally very small and corresponding MCMC estimates will typically not be reliable. Therefore, we have focused on the estimation of posterior marginal inclusion probabilities. Figure 7.10 (a) shows that for the PCR dataset two variables (genes)



(a) PCR dataset



(b) Leukemia dataset

Figure 7.10.: Real data examples: Scatterplots of estimated posterior inclusion probabilities for the different runs of MAdaSub ($q = 2$, $q = 5$ and $q = 10$).

stand out with respect to their estimated posterior inclusion probabilities, namely the genes 1438937_x_at (with estimated probability around 0.5) and 1438936_s_at (with estimated probability around 0.3). Recall from Section 5.2.2 that for the PCR dataset, the thresholded model $\hat{S}_{0.9}$ (with threshold $\rho = 0.9$) and the "best" model selected by AdaSub both consist of the single gene 1438937_x_at, which has the largest posterior marginal inclusion probability in the corresponding Bayesian analysis.

Similarly, Figure 7.10 (b) indicates that for the leukemia dataset two or three genes stand out with respect to their estimated posterior inclusion probabilities, namely the genes M23197_at (with estimated prob. between 0.5 and 0.7), X95735_at (with estimated prob. between 0.2 and 0.25) and M31523_at (with estimated prob. between 0.07 and 0.15). Again, recall from Section 6.4.2 that for the leukemia dataset, the thresholded model $\hat{S}_{0.5}$ (with threshold $\rho = 0.5$) and the "best" model selected by AdaSub both consist of the single gene M23197_at, having also the largest posterior inclusion probability in the Bayesian analysis.

8. Conclusions and future work

In this final chapter we give a compact summary of the main results presented in this thesis. We conclude with a discussion of possible directions for further research.

8.1. Summary of main results

In Chapter 2 we have provided a selective overview of classical approaches to the variable selection problem in the high-dimensional GLM setup. In particular, we have contrasted ℓ_0- and ℓ_1-type regularization methods and concluded that ℓ_0-type methods like the EBIC have desirable theoretical properties in comparison to ℓ_1-type methods like the Lasso, but lead to generally NP-hard discrete optimization problems.

Motivated by these insights, in **Chapter 3** we have introduced the Adaptive Subspace (AdaSub) method which aims at providing a solution to the natural ℓ_0-regularized optimization problem for high-dimensional variable selection. We have shown that the evolution of AdaSub can be viewed as a Markov chain and that AdaSub can be interpreted as a form of Bayesian learning. We have illustrated the behaviour of the algorithm with respect to its tuning parameters — the initial expected search size q and the learning rate K — and provided recommendations for a sensible choice of these tuning parameters in practice. In addition, we have discussed the computational complexity of AdaSub and remarked that it is apparently difficult to derive general and exact computational complexity results due to the stochastic nature of the algorithm.

In **Chapter 4** we have investigated the limiting properties of AdaSub. The three main results from this theoretical chapter are:

(i) If the ordered importance property (OIP) is satisfied, then AdaSub converges against the optimal solution of the generally NP-hard problem (see Theorem 4.8).

(ii) AdaSub provides a stable thresholded model even when OIP is not guaranteed to hold (see Theorem 4.12 and related discussions).

(iii) If the employed criterion is (quasi-)consistent under some conditions (see Definition 4.9), then AdaSub yields a variable selection consistent procedure under the same conditions, provided that the population ordered importance property (POIP) is satisfied and that only a bounded number of explanatory variables is "related" with the response (see Theorem 4.13).

In **Chapter 5** we have demonstrated that AdaSub can provide more "stable" models with less false positives in a situation where the best model according to an ℓ_0-type criterion would lead to "overfitting" and failure of the OIP condition. Furthermore, in the framework of linear regression models we have shown through simulated and real data examples that the performance of AdaSub is very competitive for high-dimensional variable selection in comparison to state-of-the-art methods like the Adaptive Lasso, the SCAD or the Bayesian split-and-merge (SAM) approach. It is notable that AdaSub significantly outperforms Stability Selection with the Lasso in many situations, which underpins the argument that usual subsampling in combination with an ℓ_1-type method might not be optimal in a high-dimensional situation. On the contrary, the application of adaptive "subsampling" in the space of explanatory variables (i.e. the feature space) can efficiently reduce the intractable ℓ_0-type high-dimensional problem to solvable low-dimensional sub-problems even in very high-dimensional situations with ten thousands of possible explanatory variables.

Since the computation of the exact solution to the sampled low-dimensional sub-problems in AdaSub can be quite expensive for GLMs different than the normal linear model, in **Chapter 6** we have proposed two variants of AdaSub which obtain greedy solutions to the sub-problems via Forward Stepwise Selection (FoAdaSub) and Backward Stepwise Selection (BackAdaSub), respectively. By using similar ideas as in Chapter 4, we have shown that FoAdaSub converges with the number of iterations against the model selected by usual Forward Stepwise Selection (see Theorem 6.1). On the other hand, we have argued that BackAdaSub closely resembles the original AdaSub method. We have provided a sufficient condition — the modified ordered importance property (MOIP) — for the convergence of BackAdaSub against the best model according to the used criterion (see Theorem 6.2). As might have been expected, it can be shown that MOIP implies OIP, indicating that

the greedy method BackAdaSub requires stronger conditions for the "correct convergence" than the computationally more challenging original AdaSub method. Nevertheless, through simulated and real data examples we have demonstrated that BackAdaSub shows a very competitive performance for variable selection in high-dimensional logistic regression models compared to other prominent methods like Forward Stepwise Selection, the Adaptive Lasso and Stability Selection.

In **Chapter 7** we have presented a Metropolized version of AdaSub, called the MAdaSub algorithm, which is an adaptive MCMC algorithm for sampling from posterior model distributions in the Bayesian variable selection context for situations where conjugate priors or approximations to the posterior are employed. The updating scheme for the individual selection probabilities of the variables in MAdaSub is very similar to the one used in AdaSub and can itself be interpreted as a form of Bayesian learning. We have shown that MAdaSub is ergodic and satisfies the weak law of large numbers for all choices of its tuning parameters; thus, it is guaranteed that, "in the limit", MAdaSub samples from the correct target distribution despite its continuing adaptation (see Theorem 7.5). As a consequence, the selection probabilities of the explanatory variables converge (in probability) against the true respective posterior marginal inclusion probabilities (see Corollary 7.6). Finally, through simulated and real data examples, we have demonstrated that MAdaSub can efficiently sample from very high-dimensional posterior model distributions.

8.2. Directions for further research

Future research motivated by this thesis can be directed towards various different directions, as outlined next.

(i) While Theorem 4.8 of Section 4.1 states that the ordered importance property (OIP) is a sufficient condition for the "correct convergence" of AdaSub, it would be desirable to address the question, whether it is also a necessary condition or whether it can be relaxed in order to provide weaker sufficient conditions. Furthermore, for the variable selection consistency of AdaSub under the population ordered importance property (POIP), in Theorem 4.13 of Section 4.3 we have additionally assumed that the number of explanatory variables "related" to the response is bounded. As indicated in Remark

4.8 of Section 4.3, it may be feasible to relax this assumption by making use of specific rates for the consistency of certain selection criteria. We have further discussed the relationship between the POIP and the partial faithfulness (PF) assumption underlying the PC-simple algorithm (Bühlmann et al., 2010) and we have argued that the POIP is a weaker condition than the PF. In future work, it would be interesting to investigate the connections between POIP and further conditions which are imposed for showing the consistency of other variable selection methods, as for example the stepwise detectable condition for the "SODA" method proposed by Li and Liu (2017).

(ii) In Remark 4.9 of Section 4.3 we have emphasized that the consistency results regarding AdaSub are with respect to the divergence of both the sample size n and the number of iterations T. It would be desirable to derive results concerning the "speed of convergence" of AdaSub as well as non-asymptotic bounds on the number of false positive and false negative selections in the final models of AdaSub. The development of further tools for assessing the convergence of AdaSub is particularly relevant for its application in practice.

(iii) In the context of the proposed modifications of AdaSub, similar theoretical questions arise as for the original AdaSub method: The modified ordered importance property (MOIP) is a sufficient condition for the "correct convergence" of BackAdaSub (Theorem 6.2 of Section 6.2.1), but is it also necessary? While we have shown that MOIP is a weaker condition than OIP, it would be desirable to characterize those situations in which OIP is satisfied, but MOIP does not hold. The theoretical investigation of the variable selection consistency properties of BackAdaSub is another interesting topic for future research.

(iv) In simulated and real data examples we have focused on the application of AdaSub in combination with the EBIC (Chen and Chen, 2008) as the variable selection criterion. Extensive simulation studies concerning the performance of AdaSub in combination with other selection criteria is an important topic for further research. Furthermore, it would be interesting to investigate the application of different variable selection methods like the Lasso or random forests (see e.g. Genuer et al., 2010) for "solving" the sub-problems of AdaSub. Note that the theoretical results for AdaSub presented in

this thesis are crucially based on the assumption that one aims at optimizing a given selection criterion $C : \mathcal{M} \to \mathbb{R}$ (which is not directly the case for variable selection methods like the Lasso or random forests). Another avenue for further research would be the combination of "adaptive subsampling" in the feature space (as in AdaSub) with "usual subsampling" in the sample space (as in Stability Selection, compare Meinshausen and Bühlmann, 2010 and Beinrucker et al., 2016).

(v) We have argued that the AdaSub method can lead to more "stable" models with less false positive selections in situations where the employed selection criterion is prone to overfitting. In Remark 5.1 of Section 5.1.1 we have conjectured that the AdaSub method has significantly less "search degrees of freedom" (compare Tibshirani, 2015) than a full model enumeration (i.e. best subset selection). The theoretical and empirical investigation of this conjecture would be an interesting challenge. Very recently, there has been new research activity in applying modern discrete optimization methods to ℓ_0-type variable selection problems, which are based on mixed-integer programming (see e.g. Bertsimas et al., 2016). In further research we would like to compare the performance of AdaSub with such optimization methods, both with respect to their computational efficiency as well as their statistical properties. Preliminary results from on-going work indicate that, from a statistical point of view, it may actually not be desirable to solve ℓ_0-type problems exactly in all situations, but that a "reasonable" approximate solution (like the thresholded model from AdaSub) can show favourable variable selection properties in finite sample regimes. In a similar vein, we also refer to Hastie et al. (2017) for a recent comparison of the predictive performances of the Lasso, Forward Stepwise Selection and Best Subset Selection in high-dimensional linear regression models.

(vi) Note that for ℓ_0-type selection criteria like the EBIC, the proper choice of the penalty parameter (i.e. the choice of $\gamma \in [0, 1]$ in EBIC_γ) is as crucial as for ℓ_1-type regularization methods like the Lasso. In this thesis we have focused on the case where the level of penalization induced by the criterion is prespecified and fixed. In future work it would be interesting to develop methods which "automatically" adapt the penalty parameter of the criterion, depending on the observed sample; for example, one might be interested in the smallest possible choice of γ in EBIC_γ (i.e. the smallest level of pe-

nalization), so that the EBIC_γ-optimal model is "stable" in the sense of OIP. Note that in a fully Bayesian setting one may alternatively make use of an appropriate hyperprior on γ.

(vii) Regarding further modifications of the AdaSub algorithm, one might consider alternative heuristic algorithms for solving the sampled sub-problems, like combinations of Forward and Backward Stepwise Selection or genetic algorithms (compare Section 2.3.2). However, we expect that more complex heuristics would complicate the derivation of convergence results similar to those derived for BackAdaSub. Another option for modifying the original AdaSub algorithm would be to consider "softer" voting schemes for the sampled sub-problems, e.g. by making use of the information from the best k models (for some integer $k \geq 2$) instead of only the single best model in each subspace. Further, note that the updating scheme for the individual selection probabilities in AdaSub "ignores" the dependencies between the explanatory variables. One may derive alternative updating schemes which make also use of multivariate information about different subsets of variables obtained from the previous iterations. However, such modifications are likely to come with a significant additional computational cost; speculatively, such modifications have the potential to lead to increased "overfitting" of discrete ℓ_0-type selection criteria.

(viii) Concerning the proposed MAdaSub algorithm for Bayesian variable selection, in future work we want to conduct extensive simulation studies investigating the efficiency of MAdaSub in comparison with several other adaptive and non-adaptive MCMC methods for Bayesian variable selection (compare Section 7.4). As already indicated, the extension of MAdaSub to settings with non-conjugate priors may be possible by incorporating additional reversible-jump moves (Green, 1995) or by using similar ideas as in Ji and Schmidler (2013). While we have shown that the MAdaSub algorithm is ergodic for all choices of its tuning parameters, in Remark 7.2 of Section 7.3 we have noted that the "speed of convergence" of MAdaSub against the targeted posterior model distribution depends crucially on a proper choice of its tuning parameters. Deriving theoretical results about the mixing time of MAdaSub is an important but challenging open problem for further research.

(ix) In this thesis we have focused on variable selection in regression models with main effects. The proper application of the proposed methods in the specific case of interaction selection with respect to (weak or strong) hierarchy constraints (compare e.g. Bien et al., 2013) will be another interesting topic for further investigation.

We conclude by emphasizing that the presented Adaptive Subspace methods — AdaSub, BackAdaSub and MAdaSub — are of a general nature and that their application is not restricted to the context of variable selection in GLMs, but could also be valuable for other classes of models. Extensions of the proposed methods for variable selection in high-dimensional generalized linear mixed models (compare e.g. Cai and Dunson, 2006 and Groll and Tutz, 2014) and structure estimation in high-dimensional graphical models (compare e.g. Dobra et al., 2004, Meinshausen and Bühlmann, 2006 and Friedman et al., 2008) may be promising avenues for future research.

A. Implementation in R

In this appendix we briefly describe the implementation of the proposed methods in the R programming language (R Core Team, 2017). Note that we plan to provide an R-package in the near future. Currently, relevant source code is available on request.

Here, we focus on the description of the three most important functions, called *Simdata*, *AdaSub* and *MAdaSub*, which have been newly implemented in R. The function *Simdata* can be used to simulate data in the same way as in the simulation studies presented in this thesis (see Simulation Setups 5.1 and 6.1). The function *AdaSub* can be used to run the original AdaSub method (Algorithm 3.1), as well as its variants FoAdaSub (Algorithm 6.1) and BackAdaSub (Algorithm 6.2). The function *MAdaSub* can be used to run the Metropolized AdaSub algorithm (Algorithm 7.1) in a Bayesian variable selection context.

A.1. R-function Simdata

Description

The function `Simdata` generates data from a generalized linear model (GLM) according to Simulation Setups 5.1 and 6.1. It depends on the R-packages `MASS` and `Matrix`.

Usage

```
Simdata(n, p, beta, family="normal", sigma.normal=1, corr=0,
        corr.type="global", blocks=10)
```

Arguments

`n`	Sample size
`p`	Number of explanatory variables
`beta`	True underlying vector of regression coefficients (without intercept)
`family`	Family of distributions for GLM with canonical link function.

	Options are `"normal"` (normal linear model), `"binomial"` (logistic regression), `"poisson"` (Poisson regression).
`sigma.normal`	Standard deviation of error in normal linear model
`corr`	Value of correlation parameter in the respective correlation structure
`corr.type`	Correlation structure. Options are `"global"`, `"toeplitz"`, `"block"`.
`blocks`	Number of blocks in block correlation structure

Value

A list with components

`x` Design matrix of dimension `n` $\times$ `p`

`y` Response vector of length `n`

Example

```
n = 100
p = 1000
beta = c(2,-2,2,-2,2,rep(0,p-5))
data = Simdata(n=n, p=p, beta=beta, family="binomial", corr=0.9,
               corr.type="toeplitz")
```

A.2. R-function AdaSub

Description

The function `AdaSub` can be used to run the AdaSub method (Algorithm 3.1), as well as its variants FoAdaSub (Algorithm 6.1) and BackAdaSub (Algorithm 6.2). It depends on the R-packages `leaps` and `RcppNumerical`. It makes use of the additional (self-implemented) R-functions `EBIC_glm` (computes the EBIC value for a given model), `FoStep` (FS2 algorithm) and `BackStep_nostop` (BS algorithm). This implementation considers the EBIC_γ with constant $\gamma \in [0, 1]$ as the selection criterion, but it can be easily modified by replacing the function `EBIC_glm` in order to include different selection criteria.

Usage

```
AdaSub(x, y, Iter, K=100, q=10, const=0, family="normal", method="full",
       U_C=100, savings=1, plotting=1000)
```

Arguments

`x`	Design matrix of dimension `n` $\times$ `p`

`y`	Response vector of length `n`
`Iter`	Number of iterations
`K`	Learning rate
`q`	Initial expected search size
`const`	Constant $\gamma \in [0,1]$ in EBIC_γ
`family`	Family of distributions for GLM with canonical link function. Options are `"normal"` (normal linear model), `"binomial"` (logistic regression), `"poisson"` (Poisson regression).
`method`	Options are `"full"` (AdaSub), `"back"` (BackAdaSub), `"forward"` (FoAdaSub). Note that `"full"` should only be used in combination with `family="normal"`.
`U_C`	Maximal size of sampled subspaces (if larger, consider subsample of size U_C)
`savings`	Save history of algorithm only at every "`savings`" iteration
`plotting`	Plot current evolution of algorithm at every "`plotting`" iteration

Value

A list with components

`relfreq.hist`	History of selection probabilities. Matrix of dimension `p` $\times \lfloor$`Iter/savings`$\rfloor$. Each row ($j \in \mathcal{P}$) contains $r_j^{(t)}$, for $t =$ `savings`$, 2 \times$ `savings`$, \ldots$
`relfreq.final`	Vector of final selection probabilities $r_j^{(T)}$, $j \in \mathcal{P}$, for $T =$ `Iter`
`S.size`	Vector of sizes $\lvert S^{(t)}\rvert$ for $t = 1, \ldots,$ `Iter`
`V.size`	Vector of sizes $\lvert V^{(t)}\rvert$ for $t = 1, \ldots,$ `Iter`
`values`	Vector of values $\text{EBIC}_\gamma(S^{(t)})$, for $t = 1, \ldots,$ `Iter`
`best.S`	Vector of indices, corresponding to the best sampled model S_b
`best.models`	List of length `Iter`, with vectors of indices $S^{(t)} \subseteq \mathcal{P}$, for $t = 1, \ldots,$ `Iter`

Examples

```
##### Example 1: AdaSub for linear regression #####
# corresponds to illustrative example in Section 4.2
p = 1000
n = 60
beta1 = numeric(p)
S0 = c(1,2,3,4,5)
beta1[S0] = c(0.4,0.8,1.2,1.6,2)
set.seed(2)
data = Simdata(n=n, p=p, beta=beta1, family="normal")
output = AdaSub(x=data$x, y=data$y, Iter=10000, K=n, q=10, const=0.6,
```

```
                family="normal", method="full")
which(output$relfreq.final>=0.9)   # thresholded model
output$best.S                      # best model

##### Example 2: BackAdaSub for logistic regression #####
p = 1000
n = 200
beta = c(2,-2,2,-2,2,rep(0,p-5))
set.seed(2)
data = Simdata(n=n, p=p, beta=beta, corr=0.5, family="binomial",
               corr.type="toeplitz")
output = AdaSub(x=data$x, y=data$y, Iter=3000, K=n, q=5, const=1,
                family="binomial", method="back")
which(output$relfreq.final>=0.9)   # thresholded model
output$best.S                      # best model
```

A.3. R-function MAdaSub

Description

The function *MAdaSub* can be used to run the Metropolized AdaSub algorithm (Algorithm 7.1) in a Bayesian variable selection context. It depends on the R-packages `BMS` and `RcppNumerical`. It makes use of the additional (self-implemented) R-functions `EBIC_glm` (computes the EBIC value for a given model), `marginal_log_like` (computes the marginal log-likelihood for a given model), `log_prob` (computes the log-probability of proposing a given model), `log_modelprior` (computes the log-prior probability of a given model) and `log_modelposterior` (computes `log_modelprior` + `log_prob`). This implementation considers the prior corresponding to EBIC_γ with constant $\gamma \in [0, 1]$ (default) and the g-prior in combination with a binomial model prior. It can be easily modified by replacing the functions `marginal_log_like` and `log_modelprior` in order to include other prior choices.

Usage

```
MAdaSub(x, y, Iter, priormean, L, const=0, family="normal", epsilon=1e-06,
        prior="EBIC", priorprob=0.5, savings=1, plotting=10000)
```

Arguments

`x`	Design matrix of dimension `n` $\times$ `p`
`y`	Response vector of length `n`
`Iter`	Number of iterations
`priormean`	Vector of initial selection probabilities $r_j^{(0)}$, for $j \in \mathcal{P}$.

	If `priormean` is a scalar, then set $r_j^{(0)} =$ `priormean` for all $j \in \mathcal{P}$.
`L`	Vector of parameters L_j, for $j \in \mathcal{P}$, controlling the adaptation in MAdaSub.
	If `L` is a scalar, then set $L_j^{(0)} =$ `L` for all $j \in \mathcal{P}$.
`const`	Constant $\gamma \in [0, 1]$ in EBIC_γ
`family`	Family of distributions for GLM with canonical link function.
	Options are `"normal"` (normal linear model), `"binomial"` (logistic regression),
	`"poisson"` (Poisson regression).
`epsilon`	"Precision" parameter $\epsilon \in (0, 0.5)$, ensuring that $\tilde{r}_j^{(t)} \in [\epsilon, 1 - \epsilon]$
`prior`	The prior used. If `prior="EBIC"`, then EBIC is used (default).
	If `prior="gprior"`, then g-prior with binomial model prior is used.
`priorprob`	If `prior="gprior"`, vector of prior inclusion probabilities $\pi(j \in S)$, $j \in \mathcal{P}$,
	in binomial model prior.
`savings`	Save history of algorithm only at every "`savings`" iteration
`plotting`	Plot current evolution of algorithm at every "`plotting`" iteration

Value

A list with components

`relfreq.hist`	History of selection probabilities. Matrix of dimension `p` $\times \lfloor$`Iter/savings`$\rfloor$.
	Each row ($j \in \mathcal{P}$) contains $r_j^{(t)}$, for $t =$ `savings`, $2 \times$ `savings`, $\ldots$
`relfreq.final`	Vector of final selection probabilities $r_j^{(T)}$, $j \in \mathcal{P}$, for $T =$ `Iter`
`S.size`	Vector of sizes $\lvert S^{(t)} \rvert$ for $t = 1, \ldots,$ `Iter`
`V.size`	Vector of sizes $\lvert V^{(t)} \rvert$ for $t = 1, \ldots,$ `Iter`
`values`	Vector of sampled (log)-values of posterior kernel
	$\log(\pi(S^{(t)} \mid \mathcal{D})) \propto \log C(S^{(t)})$, for $t = 1, \ldots,$ `Iter`
`best.S`	Vector of indices, corresponding to the best sampled model
`sampled.models`	List of length `Iter`, with vectors of indices $S^{(t)} \subseteq \mathcal{P}$, for $t = 1, \ldots,$ `Iter`
`acc.prob`	Vector of length `Iter`, with current ("running") acceptance rates
`acc`	Total number of accepted moves after `Iter` iterations

Example

```
## This corresponds to high-dimensional simulated example in Section 7.5.2
p = 1000
n = 60
beta1 = numeric(p)
```

```
S0 = c(1,2,3,4,5)
beta1[S0] = c(0.4,0.8,1.2,1.6,2)
set.seed(2)
data = Simdata(n=n, p=p, beta=beta1, family="normal")
Iter = 200000
set.seed(13)
output = MAdaSub(x=data$x, y=data$y, Iter=Iter, priormean=10/p, L=p,
                 const=1, family="normal", prior="EBIC")
which(output$relfreq.final>=0.5) # estimated median probability model
output$best.S                    # estimated MAP model
output$acc/Iter                  # acceptance rate
```

References

Agresti, A. (2015). *Foundations of linear and generalized linear models*. New York: John Wiley & Sons.

Ai-Jun, Y. and S. Xin-Yuan (2009). Bayesian variable selection for disease classification using gene expression data. *Bioinformatics 26*(2), 215–222.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control 19*(6), 716–723.

Albert, A. and J. A. Anderson (1984). On the existence of maximum likelihood estimates in logistic regression models. *Biometrika 71*(1), 1–10.

Alon, U., N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences 96*(12), 6745–6750.

Ambroise, C. and G. J. McLachlan (2002). Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the National Academy of Sciences 99*(10), 6562–6566.

Bach, F. R. (2008). Bolasso: Model consistent lasso estimation through the bootstrap. In *Proceedings of the 25th international conference on Machine learning*, pp. 33–40. ACM.

Baraud, Y., C. Giraud, and S. Huet (2009). Gaussian model selection with an unknown variance. *The Annals of Statistics 37*(2), 630–672.

Barber, R. F. and M. Drton (2015). High-dimensional Ising model selection with Bayesian information criteria. *Electronic Journal of Statistics 9*(1), 567–607.

Barbieri, M. M. and J. O. Berger (2004). Optimal predictive model selection. *The Annals of Statistics 32*(3), 870–897.

Barndorff-Nielsen, O. (2014). *Information and exponential families in statistical theory*. New York: John Wiley & Sons.

Beinrucker, A., Ü. Dogan, and G. Blanchard (2016). Extensions of stability selection using subsamples of observations and covariates. *Statistics and Computing 26*(5), 1059–1077.

Belloni, A. and V. Chernozhukov (2013). Least squares after model selection in high-dimensional sparse models. *Bernoulli 19*(2), 521–547.

Bertsekas, D. P. (1999). *Nonlinear programming*. Belmont: Athena scientific.

Bertsimas, D., A. King, and R. Mazumder (2016). Best subset selection via a modern optimization lens. *The Annals of Statistics 44*(2), 813–852.

Bien, J., J. Taylor, and R. Tibshirani (2013). A lasso for hierarchical interactions. *Annals of Statistics 41*(3), 11111141.

Bogdan, M., J. K. Ghosh, and R. Doerge (2004). Modifying the Schwarz Bayesian information criterion to locate multiple interacting quantitative trait loci. *Genetics 167*(2), 989–999.

Bogdan, M., E. van den Berg, C. Sabatti, W. Su, and E. J. Candès (2015). SLOPE — adaptive variable selection via convex optimization. *The Annals of Applied Statistics 9*(3), 1103–1140.

Bottolo, L., M. Chadeau-Hyam, D. I. Hastie, S. R. Langley, E. Petretto, L. Tiret, D. Tregouet, and S. Richardson (2011). ESS++: A C++ objected-oriented algorithm for Bayesian stochastic search model exploration. *Bioinformatics 27*(4), 587–588.

Bottolo, L. and S. Richardson (2010). Evolutionary stochastic search for Bayesian model exploration. *Bayesian Analysis 5*(3), 583–618.

Bozdogan, H. (1987). Model selection and Akaike's information criterion (AIC): The general theory and its analytical extensions. *Psychometrika 52*(3), 345–370.

Breiman, L. (1996). Heuristics of instability and stabilization in model selection. *The Annals of Statistics 24*(6), 2350–2383.

Brooks, S. P., N. Friel, and R. King (2003). Classical model selection via simulated annealing. *Journal of the Royal Statistical Society, Ser. B 65*(2), 503–520.

Brooks, S. P. and B. J. T. Morgan (1995). Optimization using simulated annealing. *The Statistician 44*(2), 241–257.

Bühlmann, P. (2013a). Causal statistical inference in high dimensions. *Mathematical Methods of Operations Research 77*(3), 357–370.

Bühlmann, P. (2013b). Statistical significance in high-dimensional linear models. *Bernoulli 19*(4), 1212–1242.

Bühlmann, P., M. Kalisch, and M. H. Maathuis (2010). Variable selection in high-dimensional linear models: Partially faithful distributions and the PC-simple algorithm. *Biometrika 97*(2), 261–278.

Bühlmann, P. and S. van de Geer (2011). *Statistics for high-dimensional data: Methods, theory and applications*. Berlin: Springer Science & Business Media.

Bühlmann, P., S. van de Geer, and S. Zhou (2015). High-dimensional inference in misspecified linear models. *Electronic Journal of Statistics 9*(1), 1449–1473.

Buntine, W. L. and A. S. Weigend (1991). Bayesian back-propagation. *Complex Systems 5*(6), 603–643.

Burnham, K. P. and D. R. Anderson (2004). Multimodel inference understanding AIC and BIC in model selection. *Sociological Methods & Research 33*(2), 261–304.

Cai, B. and D. B. Dunson (2006). Bayesian covariance selection in generalized linear mixed models. *Biometrics 62*(2), 446–457.

Candes, E. and T. Tao (2007). The Dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics 35*(6), 2313–2351.

Carlin, B. P. and S. Chib (1995). Bayesian model choice via Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society, Ser. B 57*(3), 473–484.

Chen, J. and Z. Chen (2008). Extended Bayesian information criteria for model selection with large model spaces. *Biometrika 95*(3), 759–771.

Chen, J. and Z. Chen (2012). Extended BIC for small-n-large-P sparse GLM. *Statistica Sinica 22*(2), 555–574.

Chen, K., I. Hu, and Z. Ying (1999). Strong consistency of maximum quasi-likelihood estimators in generalized linear models with fixed and adaptive designs. *The Annals of Statistics 27*(4), 1155–1163.

Chen, Z. and J. Chen (2009). Tournament screening cum EBIC for feature selection with high-dimensional feature spaces. *Science in China Series A: Mathematics 52*(6), 1327–1341.

Cho, H. and P. Fryzlewicz (2012). High dimensional variable selection via tilting. *Journal of the Royal Statistical Society, Ser. B 74*(3), 593–622.

Chrétien, S. and S. Darses (2014). Sparse recovery with unknown variance: A LASSO-type approach. *IEEE Transactions on Information Theory 60*(7), 3970–3988.

Christensen, R. (2011). *Plane answers to complex questions: The theory of linear models*. New York: Springer Science & Business Media.

Claeskens, G. and N. L. Hjort (2008). *Model selection and model averaging*. Cambridge: University Press Cambridge.

Clyde, M. (2017). *BAS: Bayesian Adaptive Sampling for Bayesian model averaging*. R package version 1.4.7.

Clyde, M. A., J. Ghosh, and M. L. Littman (2011). Bayesian adaptive sampling for variable selection and model averaging. *Journal of Computational and Graphical Statistics 20*(1), 80–101.

Dayal, M. (2016). *Cepp: Context driven Exploratory Projection Pursuit*. R package version 1.7.

Del Moral, P., A. Doucet, and A. Jasra (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society, Ser. B 68*(3), 411–436.

Dellaportas, P., J. J. Forster, and I. Ntzoufras (2002). On Bayesian model and variable selection using MCMC. *Statistics and Computing 12*(1), 27–36.

Dettling, M. and P. Bühlmann (2003). Boosting for tumor classification with gene expression data. *Bioinformatics 19*(9), 1061–1069.

Dezeure, R., P. Bühlmann, L. Meier, and N. Meinshausen (2015). High-Dimensional Inference: Confidence Intervals, p-Values and R-Software hdi. *Statistical Science 30*(4), 533–558.

Dobra, A., C. Hans, B. Jones, J. R. Nevins, G. Yao, and M. West (2004). Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis 90*(1), 196–212.

Dubins, L. E. and D. A. Freedman (1965). A sharper form of the Borel-Cantelli lemma and the strong law. *The Annals of Mathematical Statistics 36*(3), 800–807.

Dudoit, S., J. Fridlyand, and T. P. Speed (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association 97*(457), 77–87.

Dumas, M.-E., S. P. Wilder, M.-T. Bihoreau, R. H. Barton, J. F. Fearnside, K. Argoud, L. D'Amato, R. H. Wallis, C. Blancher, and H. C. Keun (2007). Direct quantitative trait locus mapping of mammalian metabolic phenotypes in diabetic and normoglycemic rat

models. *Nature Genetics 39*(5), 666–672.

Eberhart, R. and J. Kennedy (1995). A new optimizer using particle swarm theory. In *MHS'95., Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995.*, pp. 39–43. IEEE.

Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani (2004). Least angle regression. *The Annals of Statistics 32*(2), 407–499.

Fan, J., S. Guo, and N. Hao (2012). Variance estimation using refitted cross-validation in ultrahigh dimensional regression. *Journal of the Royal Statistical Society, Ser. B 74*(1), 37–65.

Fan, J. and R. Li (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association 96*(456), 1348–1360.

Fan, J. and J. Lv (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society, Ser. B 70*(5), 849–911.

Fan, J. and J. Lv (2010). A selective overview of variable selection in high dimensional feature space. *Statistica Sinica 20*(1), 101–148.

Fan, J. and J. Lv (2011). Nonconcave penalized likelihood with NP-dimensionality. *IEEE Transactions on Information Theory 57*(8), 5467–5484.

Fan, J. and H. Peng (2004). Nonconcave penalized likelihood with a diverging number of parameters. *The Annals of Statistics 32*(3), 928–961.

Fan, J. and R. Song (2010). Sure independence screening in generalized linear models with NP-dimensionality. *The Annals of Statistics 38*(6), 3567–3604.

Fan, Y. and C. Y. Tang (2013). Tuning parameter selection in high dimensional penalized likelihood. *Journal of the Royal Statistical Society, Ser. B 75*(3), 531–552.

Flom, P. L. and D. L. Cassell (2007). Stopping stepwise: Why stepwise and similar selection methods are bad, and what you should use. *NorthEast SAS Users Group (NESUG): Statistics and Data Analysis*.

Foster, D. P. and E. I. George (1994). The risk inflation criterion for multiple regression. *The Annals of Statistics 22*(4), 1947–1975.

Foygel, R. and M. Drton (2010). Extended Bayesian information criteria for Gaussian graphical models. In *Advances in Neural Information Processing Systems*, pp. 604–612.

Frank, L. E. and J. H. Friedman (1993). A statistical view of some chemometrics regression

tools. *Technometrics 35*(2), 109–135.

Friedman, J., T. Hastie, H. Höfling, and R. Tibshirani (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics 1*(2), 302–332.

Friedman, J., T. Hastie, and R. Tibshirani (2000). Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics 28*(2), 337–407.

Friedman, J., T. Hastie, and R. Tibshirani (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics 9*(3), 432–441.

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

Frommlet, F., F. Ruhaltinger, P. Twaróg, and M. Bogdan (2012). Modified versions of Bayesian Information Criterion for genome-wide association studies. *Computational Statistics & Data Analysis 56*(5), 1038–1051.

Fu, W. J. (1998). Penalized regressions: The bridge versus the lasso. *Journal of Computational and Graphical Statistics 7*(3), 397–416.

Furnival, G. M. and R. W. Wilson (1974). Regressions by leaps and bounds. *Technometrics 16*(4), 499–511.

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin (2014). *Bayesian data analysis*, Volume 2. Boca Raton: Chapman & Hall/CRC.

Gelman, A., A. Jakulin, M. G. Pittau, and Y.-S. Su (2008). A weakly informative default prior distribution for logistic and other regression models. *The Annals of Applied Statistics 2*(4), 1360–1383.

Gelman, A. and Y.-S. Su (2016). *arm: Data analysis using regression and multilevel/hierarchical models*. R package version 1.9-3.

Genuer, R., J.-M. Poggi, and C. Tuleau-Malot (2010). Variable selection using random forests. *Pattern Recognition Letters 31*(14), 2225–2236.

George, E. I. and R. E. McCulloch (1993). Variable selection via Gibbs sampling. *Journal of the American Statistical Association 88*(423), 881–889.

George, E. I. and R. E. McCulloch (1997). Approaches for Bayesian variable selection. *Statistica sinica 7*, 339–373.

Gheyas, I. A. and L. S. Smith (2010). Feature subset selection in large dimensionality

domains. *Pattern Recognition 43*(1), 5–13.

Ghosh, J. and M. A. Clyde (2011). Rao–blackwellization for bayesian variable selection and model averaging in linear and binary regression: A novel data augmentation approach. *Journal of the American Statistical Association 106*(495), 1041–1052.

Giraud, C., S. Huet, and N. Verzelen (2012). High-dimensional regression with unknown variance. *Statistical Science 27*(4), 500–518.

Givens, G. H. and J. A. Hoeting (2012). *Computational statistics*, Volume 710. New York: John Wiley & Sons.

Glover, F. (1990). Tabu search: A tutorial. *Interfaces 20*(4), 74–94.

Golub, T. (2017). *golubEsets: ExprSets for Golub leukemia data*. R package version 1.20.0.

Golub, T., D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, and M. A. Caligiuri (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science 286*(5439), 531–537.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika 82*(4), 711–732.

Griffin, J., K. Latuszynski, and M. Steel (2014). Individual adaptation: An adaptive MCMC scheme for variable selection problems. *arXiv preprint arXiv:1412.6760*.

Griffin, J., K. Latuszynski, and M. Steel (2017). In Search of Lost (Mixing) Time: Adaptive Markov chain Monte Carlo schemes for Bayesian variable selection with very large p. *arXiv preprint arXiv:1708.05678*.

Groll, A. and G. Tutz (2014). Variable selection for generalized linear mixed models by L_1-penalized estimation. *Statistics and Computing 24*(2), 137–154.

Guyon, I. and A. Elisseeff (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research 3*, 1157–1182.

Hanley, J. A. and B. J. McNeil (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology 143*(1), 29–36.

Hans, C. (2009). Bayesian lasso regression. *Biometrika 96*(4), 835–845.

Hans, C. (2010). Model uncertainty and variable selection in Bayesian lasso regression. *Statistics and Computing 20*(2), 221–229.

Hans, C., A. Dobra, and M. West (2007). Shotgun stochastic search for "large p" regression.

Journal of the American Statistical Association 102(478), 507–516.

Hardin, J. W. and J. M. Hilbe (2007). *Generalized linear models and extensions* (3 ed.). Texas: Stata press.

Hastie, T., R. Tibshirani, and R. J. Tibshirani (2017). Extended comparisons of best subset selection, forward stepwise selection, and the lasso. *arXiv preprint arXiv:1707.08692*.

Hastie, T., R. Tibshirani, and M. Wainwright (2015). *Statistical learning with sparsity*. Boca Raton: CRC press.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika 57*(1), 97–109.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*(8), 832–844.

Hochreiter, S. and J. Schmidhuber (1997). Flat minima. *Neural Computation 9*(1), 1–42.

Hocking, R. R. (1976). A Biometrics invited paper. The analysis and selection of variables in linear regression. *Biometrics 32*(1), 1–49.

Hofmann, M., C. Gatu, and E. J. Kontoghiorghes (2007). Efficient algorithms for computing the best subset regression models for large-scale problems. *Computational Statistics & Data Analysis 52*(1), 16–29.

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Massachusetts: MIT press.

Huang, J., S. Ma, and C.-H. Zhang (2008). Adaptive Lasso for sparse high-dimensional regression models. *Statistica Sinica 18*(4), 1603–1618.

Huo, X. and X. Ni (2007). When do stepwise algorithms meet subset selection criteria? *The Annals of Statistics 35*(2), 870–887.

Javanmard, A. and A. Montanari (2014). Confidence intervals and hypothesis testing for high-dimensional regression. *Journal of Machine Learning Research 15*(1), 2869–2909.

Ji, C. and S. C. Schmidler (2013). Adaptive markov chain Monte Carlo for Bayesian variable selection. *Journal of Computational and Graphical Statistics 22*(3), 708–728.

Jørgensen, B. (1987). Exponential dispersion models. *Journal of the Royal Statistical Society, Ser. B 49*(2), 127–162.

JRSS(B) Datasets, Vol. 77:5 (2015). `https://rss.onlinelibrary.wiley.com/hub/`

journal/14679868/series-b-datasets/pre_2016.

Kapetanios, G. (2007). Variable selection in regression models using nonstandard optimisation of information criteria. *Computational Statistics & Data Analysis 52*(1), 4–15.

Kass, R. E. and A. E. Raftery (1995). Bayes factors. *Journal of the American Statistical Association 90*(430), 773–795.

Kateri, M. (2014). *Contingency table analysis*. New York: Springer.

Kennedy, J. and R. C. Eberhart (1997). A discrete binary version of the particle swarm algorithm. In *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, Volume 5, pp. 4104–4108. IEEE.

Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi (1983). Optimization by simulated annealing. *Science 220*(4598), 671–680.

Knight, K. and W. Fu (2000). Asymptotics for lasso-type estimators. *The Annals of Statistics 28*(5), 1356–1378.

Kohn, R., M. Smith, and D. Chan (2001). Nonparametric regression using linear combinations of basis functions. *Statistics and Computing 11*(4), 313–322.

Konishi, S. and G. Kitagawa (1996). Generalised information criteria in model selection. *Biometrika 83*(4), 875–890.

Kuo, L. and B. Mallick (1998). Variable selection for regression models. *Sankhyā: The Indian Journal of Statistics, Series B 60*(1), 65–81.

Lai, C., M. J. Reinders, and L. Wessels (2006). Random subspace method for multivariate feature selection. *Pattern Recognition Letters 27*(10), 1067–1076.

Lamnisos, D., J. E. Griffin, and M. F. Steel (2013). Adaptive Monte Carlo for Bayesian variable selection in regression models. *Journal of Computational and Graphical Statistics 22*(3), 729–748.

Lan, H., M. Chen, J. B. Flowers, B. S. Yandell, D. S. Stapleton, C. M. Mata, E. T.-K. Mui, M. T. Flowers, K. L. Schueler, and K. F. Manly (2006). Combined expression trait correlations and expression quantitative trait locus mapping. *PLoS Genetics 2*(1), e6.

Lee, J. D., D. L. Sun, Y. Sun, and J. E. Taylor (2016). Exact post-selection inference, with application to the lasso. *The Annals of Statistics 44*(3), 907–927.

Li, Y. and J. S. Liu (2017). Robust variable and interaction selection for logistic regression and general index models. *Journal of the American Statistical Association* (just-accepted).

Liang, F., R. Paulo, G. Molina, M. A. Clyde, and J. O. Berger (2008). Mixtures of g priors for Bayesian variable selection. *Journal of the American Statistical Association 103*(481), 410–423.

Liang, F., Q. Song, and K. Yu (2013). Bayesian subset modeling for high-dimensional generalized linear models. *Journal of the American Statistical Association 108*(502), 589–606.

Liang, F. and W. H. Wong (2000). Evolutionary Monte Carlo: Applications to C_p model sampling and change point problem. *Statistica Sinica 10*(2), 317–342.

Lockhart, R., J. Taylor, R. J. Tibshirani, and R. Tibshirani (2014). A significance test for the lasso. *The Annals of Statistics 42*(2), 413–468.

Loughrey, J. and P. Cunningham (2005). Overfitting in wrapper-based feature subset selection: The harder you try the worse it gets. In *Research and Development in Intelligent Systems XXI*, pp. 33–43. Springer.

Lumley, T. and A. Miller (2009). Leaps: Regression subset selection. R package version 2.9. *See* `http://CRAN.R-project.org/package=leaps`.

Luo, S. and Z. Chen (2013). Extended BIC for linear regression models with diverging number of relevant features and high or ultra-high feature spaces. *Journal of Statistical Planning and Inference 143*(3), 494–504.

Lv, J. and J. S. Liu (2014). Model selection principles in misspecified models. *Journal of the Royal Statistical Society, Ser. B 76*(1), 141–167.

Lykou, A. and I. Ntzoufras (2013). On bayesian lasso variable selection and the specification of the shrinkage parameter. *Statistics and Computing 23*(3), 1–30.

Madigan, D., J. York, and D. Allard (1995). Bayesian graphical models for discrete data. *International Statistical Review / Revue Internationale de Statistique 63*(2), 215–232.

Mallick, H. and N. Yi (2013). Bayesian methods for high dimensional linear models. *Journal of Biometrics & Biostatistics 1*, 005.

Mallows, C. L. (1973). Some comments on C_P *Technometrics 15*(4), 661–675.

Mallows, C. L. (1995). More comments on C_P *Technometrics 37*(4), 362–372.

McCullagh, P. and J. A. Nelder (1989). *Generalized Linear Models (Second edition)*. London: Chapman & Hall.

Meinshausen, N. (2007). Relaxed lasso. *Computational Statistics & Data Analysis 52*(1),

374–393.

Meinshausen, N. and P. Bühlmann (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics 34*(3), 1436–1462.

Meinshausen, N. and P. Bühlmann (2010). Stability selection. *Journal of the Royal Statistical Society, Ser. B 72*(4), 417–473.

Meinshausen, N., L. Meier, and P. Bühlmann (2009). P-values for high-dimensional regression. *Journal of the American Statistical Association 104*(488), 1671–1681.

Meinshausen, N. and B. Yu (2009). Lasso-type recovery of sparse representations for high-dimensional data. *The Annals of Statistics 37*(1), 246–270.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics 21*(6), 1087–1092.

Miller, A. J. (1984). Selection of subsets of regression variables. *Journal of the Royal Statistical Society, Ser. A 147*(3), 389–425.

Narendra, P. M. and K. Fukunaga (1977). A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers 26*(9), 917–922.

Natarajan, B. K. (1995). Sparse approximate solutions to linear systems. *SIAM Journal on Computing 24*(2), 227–234.

Nelder, J. A. and R. W. M. Wedderburn (1972). Generalized Linear Models. *Journal of the Royal Statistical Society, Ser. A 135*(3), 370–384.

Ni, X. S. and X. Huo (2005). Enhanced leaps-and-bounds method in subset selections with additional optimality tests. *INFORMS QSR student paper competition finalist. Available at qsr.section.informs.org/qsr_activities.htm.*

Nikolova, M. (2013). Description of the Minimizers of Least Squares Regularized with ℓ_0-norm. Uniqueness of the Global Minimizer. *SIAM Journal on Imaging Sciences 6*(2), 904–937.

Nishii, R. (1984). Asymptotic properties of criteria for selection of variables in multiple regression. *The Annals of Statistics 12*(2), 758–765.

Nishii, R. (1988). Maximum likelihood principle and model selection when the true model is unspecified. *Journal of Multivariate Analysis 27*(2), 392–403.

Nott, D. J. and R. Kohn (2005). Adaptive sampling for Bayesian variable selection.

Biometrika 92(4), 747–763.

O'Hara, R. B. and M. J. Sillanpää (2009). A review of Bayesian variable selection methods: What, how and which. *Bayesian Analysis 4*(1), 85–117.

Pacheco, J., S. Casado, and L. Núñez (2009). A variable selection method based on Tabu search for logistic regression models. *European Journal of Operational Research 199*(2), 506–511.

Park, M. Y. and T. Hastie (2007). L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society, Ser. B 69*(4), 659–677.

Park, T. and G. Casella (2008). The Bayesian lasso. *Journal of the American Statistical Association 103*(482), 681–686.

Pasarica, C. and A. Gelman (2010). Adaptively scaling the Metropolis algorithm using expected squared jumped distance. *Statistica Sinica 20*(1), 343–364.

Pearl, J. (2009). Causal inference in statistics: An overview. *Statistics Surveys 3*, 96–146.

Pötscher, B. M. and U. Schneider (2009). On the distribution of the adaptive lasso estimator. *Journal of Statistical Planning and Inference 139*(8), 2775–2790.

Qiu, Y., S. Balan, M. Beall, M. Sauder, N. Okazaki, and T. Hahn (2016). *RcppNumerical: 'Rcpp' integration for numerical computing libraries*. R package version 0.3-1.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Raftery, A. E., D. Madigan, and J. A. Hoeting (1997). Bayesian model averaging for linear regression models. *Journal of the American Statistical Association 92*(437), 179–191.

Raskutti, G., M. J. Wainwright, and B. Yu (2011). Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls. *IEEE Transactions on Information Theory 57*(10), 6976–6994.

Reid, S., R. Tibshirani, and J. Friedman (2016). A study of error variance estimation in lasso regression. *Statistica Sinica 26*, 35–67.

Ridout, M. S. (1988). Algorithm AS 233: An improved branch and bound algorithm for feature subset selection. *Journal of the Royal Statistical Society, Ser. C 37*(1), 139–147.

Roberts, G. O. and J. S. Rosenthal (2004). General state space Markov chains and MCMC algorithms. *Probability Surveys 1*, 20–71.

Roberts, G. O. and J. S. Rosenthal (2007). Coupling and ergodicity of adaptive Markov

chain Monte Carlo algorithms. *Journal of Applied Probability 44*(2), 458–475.

Roberts, S. J. (1984). Algorithm AS 199: A branch and bound algorithm for determining the optimal feature subset of given size. *Journal of the Royal Statistical Society, Ser. C 33*(2), 236–241.

Saeys, Y., I. Inza, and P. Larrañaga (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics 23*(19), 2507–2517.

Schäfer, C. and N. Chopin (2013). Sequential Monte Carlo on large binary sampling spaces. *Statistics and Computing 23*(2), 1–22.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics 6*(2), 461–464.

Scott, J. G. and J. O. Berger (2010). Bayes and empirical-Bayes multiplicity adjustment in the variable-selection problem. *The Annals of Statistics*, 2587–2619.

Shah, R. D. and R. J. Samworth (2013). Variable selection with error control: Another look at stability selection. *Journal of the Royal Statistical Society, Ser. B 75*(1), 55–80.

Shao, J. (1997). An asymptotic theory for linear model selection. *Statistica Sinica 7*, 221–242.

Somol, P., P. Pudil, and J. Kittler (2004). Fast branch & bound algorithms for optimal feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*(7), 900–912.

Song, Q. and F. Liang (2015a). A split-and-merge Bayesian variable selection approach for ultrahigh dimensional regression. *Journal of the Royal Statistical Society, Ser. B 77*(5), 947–972.

Song, Q. and F. Liang (2015b). High-dimensional variable selection with reciprocal L1-regularization. *Journal of the American Statistical Association 110*(512), 1607–1620.

Staerk, C. and M. Kateri (2017). Stable variable selection with AdaSub. In *Proc. 32nd International Workshop on Statistical Modelling*, pp. 91–96.

Staerk, C., M. Kateri, and I. Ntzoufras (2016). An adaptive subspace method for high-dimensional variable selection. In *Proc. 31st International Workshop on Statistical Modelling*, pp. 295–300.

Staerk, C., M. Kateri, and I. Ntzoufras (2018). High-dimensional variable selection via low-dimensional adaptive learning. *Submitted*.

Su, W., M. Bogdan, and E. Candes (2017). False discoveries occur early on the Lasso path.

The Annals of Statistics 45(5), 2133–2150.

Thompson, M. L. (1978). Selection of variables in multiple regression: Part II. Chosen procedures, computations and examples. *International Statistical Review/Revue Internationale de Statistique 46*(2), 129–146.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Ser. B 58*(1), 267–288.

Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: A retrospective. *Journal of the Royal Statistical Society, Ser. B 73*(3), 273–282.

Tibshirani, R., M. Saunders, S. Rosset, J. Zhu, and K. Knight (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society, Ser. B 67*(1), 91–108.

Tibshirani, R. J. (2015). Degrees of freedom and model search. *Statistica Sinica 25*(3), 1265–1296.

Tierney, L. (1994). Markov chains for exploring posterior distributions. *The Annals of Statistics 22*(4), 1701–1728.

Tropp, J. A. and A. C. Gilbert (2007). Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory 53*(12), 4655–4666.

Unler, A. and A. Murat (2010). A discrete particle swarm optimization method for feature selection in binary classification problems. *European Journal of Operational Research 206*(3), 528–539.

van de Geer, S. (2008). High-dimensional generalized linear models and the lasso. *The Annals of Statistics 36*(2), 614–645.

van de Geer, S. and P. Bühlmann (2009). On the conditions used to prove oracle results for the Lasso. *Electronic Journal of Statistics 3*, 1360–1392.

van de Geer, S., P. Bühlmann, Y. Ritov, and R. Dezeure (2014). On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics 42*(3), 1166–1202.

van de Geer, S., P. Bühlmann, and S. Zhou (2011). The adaptive and the thresholded Lasso for potentially misspecified models (and a lower bound for the Lasso). *Electronic Journal of Statistics 5*, 688–749.

van de Geer, S. and P. Müller (2012). Quasi-likelihood and/or robust estimation in high

dimensions. *Statistical Science 27*(4), 469–480.

Wang, H. (2009). Forward regression for ultra-high dimensional variable screening. *Journal of the American Statistical Association 104*(488), 1512–1524.

Wang, H., B. Li, and C. Leng (2009). Shrinkage tuning parameter selection with a diverging number of parameters. *Journal of the Royal Statistical Society, Ser. B 71*(3), 671–683.

Wang, H., R. Li, and C.-L. Tsai (2007). Tuning parameter selectors for the smoothly clipped absolute deviation method. *Biometrika 94*(3), 553–568.

Wang, X., D. B. Dunson, and C. Leng (2016). DECOrrelated feature space partitioning for distributed sparse regression. In *Advances in Neural Information Processing Systems*, pp. 802–810.

Wasserman, L. (2000). Bayesian model selection and model averaging. *Journal of Mathematical Psychology 44*(1), 92–107.

Wasserman, L. (2014). Discussion: "A significance test for the Lasso". *The Annals of Statistics 42*(2), 501–508.

Wasserman, L. and K. Roeder (2009). High dimensional variable selection. *The Annals of Statistics 37*(5A), 2178–2201.

White, H. (1982). Maximum likelihood estimation of misspecified models. *Econometrica: Journal of the Econometric Society 50*(1), 1–25.

Witten, D. M. and R. Tibshirani (2009). Covariance-regularized regression and classification for high dimensional problems. *Journal of the Royal Statistical Society, Ser. B 71*(3), 615–636.

Wolpert, D. H. and W. G. Macready (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation 1*(1), 67–82.

Xue, B., M. Zhang, W. N. Browne, and X. Yao (2016). A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation 20*(4), 606–626.

Yang, J. and V. Honavar (1998). Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems and their Applications 13*(2), 44–49.

Yang, Y. (2005). Can the strengths of AIC and BIC be shared? A conflict between model indentification and regression estimation. *Biometrika 92*(4), 937–950.

Yang, Y., M. J. Wainwright, and M. I. Jordan (2016). On the computational complexity of

high-dimensional Bayesian variable selection. *The Annals of Statistics 44*(6), 2497–2532.

Yuan, M. and Y. Lin (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Ser. B 68*(1), 49–67.

Żak-Szatkowska, M. and M. Bogdan (2011). Modified versions of the Bayesian information criterion for sparse generalized linear models. *Computational Statistics & Data Analysis 55*(11), 2908–2924.

Zellner, A. (1986). On assessing prior distributions and Bayesian regression analysis with g-prior distributions. *Bayesian inference and decision techniques: Essays in Honor of Bruno De Finetti 6*, 233–243.

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics 38*(2), 894–942.

Zhang, C.-H. and J. Huang (2008). The sparsity and bias of the lasso selection in high-dimensional linear regression. *The Annals of Statistics 36*(4), 1567–1594.

Zhang, T. (2009). On the consistency of feature selection using greedy least squares regression. *Journal of Machine Learning Research 10*, 555–568.

Zhang, T. (2011). Adaptive forward-backward greedy algorithm for learning sparse representations. *IEEE Transactions on Information Theory 57*(7), 4689–4708.

Zhang, Y., R. Li, and C.-L. Tsai (2010). Regularization parameter selections via generalized information criterion. *Journal of the American Statistical Association 105*(489), 312–323.

Zhang, Y., M. J. Wainwright, and M. I. Jordan (2017). Optimal prediction for sparse linear models? Lower bounds for coordinate-separable M-estimators. *Electronic Journal of Statistics 11*(1), 752–799.

Zhao, P. and B. Yu (2006). On model selection consistency of Lasso. *Journal of Machine Learning Research 7*, 2541–2563.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association 101*(476), 1418–1429.