Methods for Detection of Interactions with Multiple Components

Hristina Pashova

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2013

Reading Committee:

Charles Kooperberg, Chair

Michael LeBlanc, Chair

Daniela Witten

Program Authorized to Offer Degree:
School of Public Health

University of Washington

Abstract

Methods for Detection of Interactions with Multiple Components

Hristina Pashova

Co-Chairs of the Supervisory Committee:
PhD Charles Kooperberg
Biostatistics

PhD Michael LeBlanc
Biostatistics

In genetic association studies, it is typically thought that important insights will be obtained through joint modeling of genetic variants and environmental variables. However, weak effect of gene-environment interactions, and imprecise measurement of the environment make it hard to identify statistically significant interaction effects.

We propose two different modeling techniques. First, for regression problems in which the main effects are already established, as is the case with many diseases or their estimation is not a priority, we propose the use of dedicated boosting. Dedicated boosting is a variation to the usual ℓ_2 boosting procedure which focuses on the interaction search in contrast to most boosting methods which address overall model prediction or classification. We compare the performance of dedicated boosting to other competing methods in the WHI data and a simulation study.

Secondly, we use the idea of a structured interaction model form together with penalized regression to limit model complexity in regression problems where we believe interactions might behave in a similar way. We propose the directed LASSO, a regression modeling strategy using a pairwise fused LASSO penalty to encourage interaction model simplicity through fusion.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to thank all the people who have made this dissertation possible. I am particularly grateful to my advisors, Dr. Charles Kooperberg and Dr. Michael LeBlanc, for their continued patience, guidance and support. I would also like to thank my committee members, Dr. Jon Wakefield and Dr. Daniela Witten, for their invaluable advice and helpful suggestions. I greatly appreciate the love and encouragement of my parents, Ivan Pashov and Ivanka Pashova, my sisters, Maria Pashova and Victoria Kehl, my friends and classmates. Finally, a special thank you to the wonderful Cameron Patterson, Denka Markova and Margreta Anastasova, the kind of friends who help you believe in yourself.

Chapter 1

INTRODUCTION

1.1 The search for interactions in high-dimensional data

In genetic association studies, it is typically thought that important insights will be obtained through joint modeling of genetic variants and environmental variables. However, weak effect of gene-environment interactions, and imprecise measurement of the environment make it hard to identify "statistically significant" interaction effects. While these issues typically reduce the power to identify interactions, there are some situations in which we have some prior information (or suspicion) about the form of the interaction. For these situations we can try to design approaches that are more powerful than generic methods.

One such example is the situation in which a combination of the measured environmental variables interacts with a particular gene, either because these measured variables are all imprecise surrogates for the actual underlying factor that interacts with the gene, or because multiple environmental factors each trigger the same biological mechanism. Traditional methods to identify gene-environment interactions typically consider only one measured environmental variable at a time. As a consequence, the power to identify such interactions is then very limited. Another setting of interest is the identification of interactions between a treatment and genetic (or environmental) factors. When the main effects for both the drug and the other factors are already established, the main interest lies in estimating effect modifications.

This work is focused on developing methods for identification of effect modifiers with multiple components such as in the above examples. A method should be able to preserve heredity constraints by only allowing interactions to enter the model once

both of the relevant main effects have been included. This significantly improves the interpretability of the results. Such a method should also be able to treat the set of interactions separately from the main effects, and incorporate other information about what interactions may be expected to look like. Typically, we assume we are dealing with already established main effects and interactions that often will be so small that these interactions could easily be masked by main effects if all predictors are considered symmetrically in the model. Thus, separate treatment of the interactions should improve the power to detect them. We are also interested in methods which allow flexibility for the form of the environmental factors and the interactions.

In this dissertation, we develop new variations of boosting and the LASSO for these types of problems. Over the last 20 years many adaptive regression methods have been developed that are specifically designed to identify interactions [22, 9, 40, 26]. These methods, however, typically do not make use of the type of information about the form of the interaction as is available in the situations above. During the last few years regression penalization methods have been developed that are well suited to incorporate such types of information into the modeling [44, 10, 45]. These methods, however, have not been widely applied to the estimation of interactions. A common theme of the examples above is that we are interested in the interactions, but are less interested in the main effects. In this situation, it is attractive to use established main effect associations to aid in the search of interactions, thus significantly reducing the number of potential variables to be considered.

In Chapter 2, we propose a dedicated boosting algorithm, which conditions out the main effects and then works in a space "perpendicular" to these main effects, similar to the idea of added variable plots [15]. This simple adjustment is able to both enforce the heredity constraint and focus on the interaction terms, thereby improving the ability to detect them.

Because we are working with interactions with small effects compared to the main effects, and because we are using already identified genetic variables, such as SNPs,

we want to treat the group of interactions separately from the main effects. In the notation below, the vector G and the columns of the matrix E are the single genetic and multiple environmental effects, respectively. The length of the vector denotes the number of observations in the training data set. The form of the environmental factors is left unspecified. Since we are particularly interested in the interaction term, we would like to treat the group consisting of the interaction terms separately from the main effects,

$$Y = \alpha + \beta G + f_1(E) + f_2(E) \times G. \tag{1.1}$$

To this end we first consider conditioning out the main effects G and $f_1(E)$. This reduces the dimension of the problem and improves our ability to estimate the interaction term $f_2(E)$. In this setting the boosting algorithm can then be applied to select an appropriate basis for the interactions. We refer to this as dedicated boosting, discussed in further detail in Chapter 2.

Secondly, most off-the-shelf techniques deal with both main effects and interaction variables in the same way. We propose to instead enforce a particular structure on the model by fusing the main effects with the interactions. The idea is to use a new set of basis functions, which restrict the form of the interaction to be based on the form of the main effects. In particular, using the above notation, one such set of basis functions is $[1, G, (1 + hG) \times E]$. Using this basis keeps the coefficients the same for the main effects and the interaction. Using a similar notation as above for the dedicated boosting, this is similar to fitting the model

$$Y = \alpha + \beta G + f_1(E) + h f_1(E) \times G. \tag{1.2}$$

If a term is selected to be in the model then both the main effect and the interaction are simultaneously included. A second parameter, h, identifies the strength and direction of the interaction compared to the main effect. Enforcing this structure on the interaction reduces the variance of the model and potentially simplifies its interpretation. The simplest model involves a single h parameter, while the most

flexible one has a separate h for each interaction term. Grouping the h's increases the interpretability of the results.

The approach described above is similar to applying the fused LASSO [45] to problem (1.2), where the sum of the differences of coefficients between the main effects and the interaction terms are penalized. This would achieve sparseness in the differences between a given main effect and its corresponding interaction term, resulting in similar values for the interaction coefficient. We discuss this further in Chapter 3.

Before we describe the two methods introduced above in more detail, we give some background information. First, we discuss two data sets and the scientific questions that we would like to address. Next, we describe other approaches to the posed problems and competing methods. Finally, we summarize the LASSO and boosting techniques, the building blocks of this work.

1.2 Motivating data sets

We have available for analysis the WHI-PAGE (Women's Health Initiative Population Architecture, using Genomics and Epidemiology) data on obesity consisting of 11 SNPs previously identified, mostly in GWAS studies, to be associated with obesity. Genotype, demographic, and environmental data assumed to be associated with obesity and collected at recruitment are available on 17,049 women. See Section 2.3 for details. The response is measured BMI (weight in kilograms divided by height in meters squared). The study design is described in detail in [20].

We want to investigate the possibility of effect modification of the association between each of the SNPs and BMI by some of the environmental and demographic variables. Because the magnitude of the effect modification is likely to be small, the dedicated boosting algorithm that we propose in this dissertation is a good candidate method of analysis. The particular composition of the group of environmental and demographic variables that modify the SNP effect is only intended to provide an

illustration of our methodology: we consider this a group of predictors that *may* be associated with BMI and that could be interacting with the SNP effect on BMI.

A data set based on patients with Diffuse Large B-cell lymphoma, a non-Hodgkin lymphoma, was also available for analysis [39]. This lymphoma data set contains gene expression data and treatment information. Approximately half of the patients were treated with standard CHOP (a multi-drug chemotherapy regiment) chemotherapy and half were treated with CHOP plus antibody therapy (Rituxan, a monoclonal antibody that is now widely used in the treatment of non-Hodgkin lymphoma) for a total of 209 patients. Previous publications had identified 36 gene expression measurements some of which were prognostic. We are interested in determining whether therapy influences the prognostic performance of the profiles based on these genes. We will apply the directed LASSO method to this set of data in order to identify gene expression profiles associated with the therapeutic impact of Rituxan.

1.3 Gene-environment interactions

Traditional methods to identify gene-environment interactions typically consider only one measured environmental variable at a time. The power to identify such variables is then very limited. Chatterjee at al. use Tukey's one degree of freedom model to combine multiple levels of environmental factors but not multiple environmental factors [13]. Thomas mentions multiple relevant susceptibility factors (environmental factors) as one of the future challenges in identifying gene-environment interactions [43].

A popular approach in working with GWAS data is to do initial univariate analyses to identify individual SNPs associated with the disease or outcome of interest [32]. Once a first round of preselection has been performed, there is an array of methods which could be applied to look for interactions among the SNPs, or the SNPs and environmental factors. For instance, multivariate adaptive regression splines (MARS) have been used in such genetic association studies [22]. Model selection methods like

MARS, however, are restricted by the number of parameters the data can "support", and the types of interactions typically need to be specified in advance.

Different approaches are needed for dealing with GWAS studies. Approaches need to be suitable for the initial selection from large numbers of SNPs. For example, random forests have been used for detection of interactions between gene and environmental factors and gene-gene interactions when the main effects of these are supposed to be small [34, 31]. Other examples of methods investigated on large sets of SNPs are Monte Carlo logic regression, and generalized boosted regression which employs single node trees as the base procedure [35].

This dissertation focuses on methods which are applied to a more restricted set of predictors. These methods could be applied to confirmatory studies of already preselected groups of gene expression variables. While these methods can still be applied to large data sets, they cannot be applied to data on the scale of GWAS studies.

1.4 Techniques for high-dimensional regression

Penalized regression methods are regression methods that impose some type of complexity constraint on the model. An example of penalized regression are shrinkage methods like ridge regression and the LASSO, the coefficients in which minimize a penalized residual sum of squares. In this section we will introduce the LASSO and some of its decendents.

Using penalized regression methods for detection of complex interactions is expected to have better performance than least squares because this will help select important predictors and interaction terms to include in the model while estimating the coefficients. When the number of potential predictors is large, penalized regression methods can improve model performance by penalizing the estimated coefficients in some fashion. Thus they exchange a small amount of acquired bias by shrinking coefficients for a decrease in variance, which often leads to better prediction performance

on an independent data set.

1.4.1 Convex approaches

The LASSO (least absolute shrinkage and classification), initially proposed by Tibshirani [44], minimizes the residual sum of squares under the condition that the sum of the absolute values of the coefficients are less than a constant λ. This is referred to as an ℓ_1 penalty. The λ parameter is a non-negative tuning parameter chosen through cross-validation (though approximations based on criteria like the AIC have also been used). It controls the amount of shrinkage applied to the coefficients. The appeal of the LASSO penalty lies in its ability to perform shrinkage of the coefficients and variable selection at the same time. This sets it apart from Ridge regression [30] where the ℓ_2 penalty is used. Ridge regression gives as good prediction results as the LASSO, but the model contains far more predictors. In the LASSO, variable selection is an artifact of some of the coefficients being shrunken all the way to zero and excluded from the final model. The LASSO can be applied even when the number of predictors p far exceeds the number of observations N. In this case, at most N predictors would be nonzero in the resulting model [46].

The LASSO estimates is defined as

$$\hat{\beta}(LASSO) = \arg\min_{\beta} \|y - \sum_{j=1}^{p} x_j\beta_j\|^2 + \lambda \sum_{j=1}^{p} |\beta_j|.$$

The LASSO has been established as a successful model selection procedure when the number of potential predictors is large. Through the shrinkage parameter the LASSO has improved prediction accuracy by trading off an increase in bias with a decrease in variance.

Forward stage-wise linear regression, a version of boosting, when using small step sizes has been shown to produce solutions approximately equivalent to that of the LASSO [28].

Next we discuss another appeal of the LASSO. Least Angle Regression (LARS), a version of Forward Stagewise Linear Regression, is a computationally efficient algorithm that can be used to obtain the entire solution path of the LASSO. The procedure starts with all coefficients in the model equal to 0 and finds the predictor which is most highly correlated with the outcome variable. It takes the largest step possible in that direction until a second predictor is as correlated with the current residual vector. At this point LARS continues in a direction which is equiangular between the two predictors until a third one becomes equally as correlated and so forth. A small modification of the LARS algorithm implements the LASSO and calculates all possible estimates for the regression problem. This lends to making the LASSO a very fast and computationally attractive procedure.

Coordinate descent is another algorithm proposed by Friedman et al. for estimation of generalized linear models with convex penalties [25]. The algorithm uses cyclical coordinate descent which estimates even faster than competing methods the regularization paths for generalized linear models with ℓ_1 and ℓ_2 penalties. The R package "glmnet" which we make use of for solving LASSO problems in this work employs coordinate descent for optimization.

Since the LASSO's introduction, many modifications to improve performance for specific problems have been proposed. For example, the fused LASSO [45] penalizes both coefficients themselves and differences between consecutive coefficients, thus taking into account a natural ordering of predictors. The fused LASSO penalty can be written as

$$\lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=2}^{p} |\beta_j - \beta_{j-1}|.$$

It induces sparseness in the difference between coefficients, grouping predictors together. The fused LASSO method is appropriate when the number of predictors is much larger than the sample size and when a natural ordering of predictors exists. Tibshirani et al. [45] show that analogous asymptotic results to those for the LASSO

also hold for the fused LASSO when the number of predictors is fixed and sample size tends to infinity [27]. Analogously to the LASSO, the authors show that the fused LASSO, under some non-redundency conditions, will have a unique solution for which the number of sequences of identical non-zero coefficients is smaller than the sample size (assuming more predictors than samples are available).

The fused LASSO is also a quadratic programming problem and computation becomes harder for larger data sets. The authors propose the use of an algorithm for solving quadratic problems specifically with sparse linear constraints. An exhaustive search over a grid of values for λ_1 and λ_2 is possible for moderately sized problems.

The pairwise fused LASSO further extends the original fused LASSO problem [38]. It includes penalties for all pairwise differences between coefficients and can be used when no natural ordering of the predictors is available. The penalty term has the form

$$\lambda_1 \sum_j |\beta_j| + \lambda_2 \sum_j \sum_k |\beta_j - \beta_k|.$$

For a linear model, the authors propose a re-parametrization which transforms the problem into a LASSO problem and allows the use of the usual LARS or coordinate descent algorithm, an attractive feature from a computational standpoint.

Another recent procedure proposed by Zou [52] is the adaptive LASSO, a variation of the usual LASSO algorithm. The adaptive LASSO gives different weights to the λ penalty applied to each coefficient. The weights are data dependent in that they are the inverse of, for instance, the OLS coefficients raised to a power greater than zero. Thus parameters with bigger OLS coefficients receive smaller penalty. For $\hat{w} = 1/|\hat{\beta}|^\gamma$, where $\hat{\beta}$ is a root-n-consistent estimator of the true coefficient, with $\gamma > 0$, the adaptive LASSO estimates are defined as

$$\hat{\beta}^\star = \arg\min_\beta \|y - \sum_{j=1}^p x_j \beta_j\|^2 + \lambda \sum_{j=1}^p \hat{w}_j |\beta_j|.$$

The adaptive LASSO is a convex optimization problem with an ℓ_1 constraint, and thus can be solved by the same algorithm as the LASSO. This makes the adaptive

LASSO easy to implement and computationally competitive. And since it is a convex optimization problem it does not suffer from multiple minima issues and the global minimizer can be solved efficiently.

One of the main complaints about the LASSO is its tendency to bias downwards (shrink) larger coefficients, because it is set up to force all coefficients to be equally penalized. One way around this is the use of weights as in the adaptive LASSO. Larger coefficients receive smaller penalization through the use of weights which alleviates the bias and improves predictive performance. Zou [52] points out that the success of the procedure is in the data dependency of the weights. With increase of sample size the zero-coefficient weights increase, while the weights for the nonzero coefficients converge to a finite constant. Thus the procedure estimates asymptotically unbiasedly large coefficients as well as small threshold estimates which set other coefficients to zero.

Two-dimensional cross-validation is needed to tune the adaptive LASSO. An optimal pair (γ, λ_n) can be found by first searching for each γ for an optimal λ_n and cross-validating to find the optimal pair. An alternative is to search over a grid of values for both tuning parameters simultaneously or to set γ a priori. Zou [52] suggests the use of the least squares estimates for β when the number of parameters p is less than the number of observations n and collinearity is not a concern.

When choosing from groups of highly correlated predictors the LASSO tends to arbitrarily select only a single variable from the group. Interpretation of the resulting prediction model is hard since highly predictive variables are left out of the model. Bondell and Reich point this out and propose a new method which builds on the idea of the LASSO [3]. OSCAR is developed as a tool to discover groups of predictors which are correlated. The method encourages groups of predictors to all have the exact same coefficient. This is achieved through the use of combination of the ℓ_1 and the pairwise ℓ_∞ norm of the coefficients, as $\sum_j |\beta_j| + c \sum_{j \leq k} \max\{|\beta_j|, |\beta_k|\}$. As in the LASSO, the ℓ_1 norm is responsible for the sparceness of the resulting model, while the

pairwise ℓ_∞ norm encourages equality between coefficients, in that it penalizes the bigger one between each pair. Hence, OSCAR eliminates the unimportant predictors and at the same time clusters the important ones.

1.4.2 Non-convex approaches

Fan and Li [19] propose the smoothly clipped absolute deviation (SCAD) family of penalization functions, which are symmetric, nonconcave on $(0, \infty)$, and have singularities at the origin which produce sparse solutions. To reduce bias in estimating the larger coefficients, the penalty function is bounded by a constant. Such a penalty function combines the ability to simultaneously perform variable selection and estimation.

The SCAD penalty for $a > 2$ and $\theta > 0$ is

$$p'_\lambda(\theta) = \lambda \left\{ I(\theta \leq \lambda) + \frac{(a\lambda - \theta)_+}{(a-1)\lambda} I(\theta > \lambda) \right\}.$$

It can be viewed as a quadratic spline with knots at λ and $a\lambda$. This penalty function also reduces the bias for larger coefficients. This is done by not penalizing the larger coefficients excessively.

Apart from the reduction in bias, the authors also demonstrate superiority over the LASSO by investigating asymptotic properties of their method. Let $\boldsymbol{\beta}_0 = (\boldsymbol{\beta}_{10}^T, \boldsymbol{\beta}_{20}^T)^T$ where $\boldsymbol{\beta}_{20} = \mathbf{0}$, and let $I(\boldsymbol{\beta}_0)$ and $I_1(\boldsymbol{\beta}_{10}, \mathbf{0})$ be the Fisher information matrix in the general case and when the set of zero coefficients is known. The authors show that there exists a penalized likelihood estimator that converges at $O_P(n^{-1/2} + a_n)$ with $a_n = \max\{p'_{\lambda_n}(|\beta_{j0}|) : \beta_{j0} \neq 0\}$, thus for the SCAD family the estimator is $\sqrt{n}$-consistent provided that $\lambda_n \to 0$. Further, they show that the estimator is able to identify $\widehat{\boldsymbol{\beta}}_2 = \mathbf{0}$ and $\widehat{\boldsymbol{\beta}}_1$ is asymptotically normal with covariance I_1^{-1} when $n^{1/2}\lambda_n \to \infty$. The regularity conditions required for these results to hold are of the nature of the conditions that guarantee asymptotic normality of the MLE's.

When a model can consistently remove noise terms with probability which goes

to 1 and the non-zero estimates have the same distribution as they would have if the true model is known in advance, by the definition of Fan and Li, this is referred to as the oracle property. Importantly, the authors show that the for the ℓ_1 penalty, the regularity conditions require that $\lambda_n = O_P(n^{-1/2})$ while at the same time $\sqrt{n}\lambda_n \to \infty$. These two conditions cannot be satisfied simultaneously. Thus they show that only penalties ℓ_q with $q < 1$ can have the oracle property.

1.4.3 Boosting

Boosting was initially developed as a classification procedure [21] and has since been adapted to the regression and general prediction settings. In the original boosting algorithms, a weak classifier is applied iteratively to re-weighted versions of the data based on its performance on a training set. The estimated predictions from each of the classifiers are then averaged to obtain the final estimator. Friedman adapted boosting to the regression setting as an optimization problem with a squared error loss function [23]. ℓ_2 boosting has been shown to produce consistent estimates in very high dimensional settings where the number of predictors is exponentially increasing with sample size [10].

The ℓ_2 boosting procedure iteratively fits a learner, a simple fitting procedure, to the residuals from the previous model's fitted values [23]. The learner can be linear or non-parametric. For example, fitting componentwise smoothing splines as a base procedure yields an additive model. The number of boosting iterations, k, is a smoothing parameter generally chosen by cross-validation. Commonly, a second parameter, step length, is used to shrink the steps taken in the direction of the best fit at a given iteration. Friedman [23] showed that a small step size is generally a good idea, as it typically improves and rarely worsens the prediction performance of the final fitted model. A small step size will slow the computation time, but for most problems ℓ_2 boosting is sufficiently fast that this is not an issue. Because the boosting algorithm typically stops after a finite number of iterations, the boosting model automatically

performs variable selection as not all predictors will be in the model. This is a highly desirable property, especially in situations with large number of predictors (sometimes more predictors than observations). Extensions of gradient boosting to likelihood functions, for instance, appropriate for binary or survival outcomes, have also been developed.

A similar and broader problem to 1.1 and 1.2 is referred to as "mandatory covariates" and has been recently addressed by Boulesteix and Hothorn [4]. The mandatory covariates are necessarily included in the model and the aim is to determine the additional predictive value of other variables, such as high dimensional molecular data. In their paper, the authors suggest the utilization of a two stage boosting procedure, implemented in the R package globalboosttest. The mandatory variables are regressed out of the outcome and then boosting is performed to determine a model with the additional covariates. While the idea is similar to dedicated boosting, further considerations need to be taken into account when dealing with interactions.

Since the interactions and the main effects are expected to be correlated, taking the extra step of regressing out the main effects from the interactions, rather than just the outcome variable allows for better performance and detection of the interaction effects. We compare the performance of dedicated boosting to the algorithm globalboosttest in simulations and a real data example from the WHI study in Chapter 2.

1.4.4 Existing techniques for detecting interactions

In their paper [14], Choi et al. are interested in penalized methods for interactions. In particular, they develop the strong heredity interaction model (SHIM) algorithm, an iterative procedure which uses the LASSO at each step to fit a model of the form

$$g(x) = \beta_0 + \sum_i \beta_i x_i + \sum_i \sum_j \gamma_{ij} \beta_i \beta_j (x_i x_j).$$

The penalized regression model the authors consider is

$$\underset{\beta,\gamma}{\text{minimize}} ||y - g(x)||^2 + \lambda_\beta \sum |\beta_j| + \lambda_\gamma \sum |\gamma_{ij}|.$$

The interaction terms are based on the main effects forcing the interactions to be zero when either main effect is zero. Choi et al. showed that their model has the asymptotic oracle property when n goes to infinity. As the sample size increases and the number of predictors remains fixed, under some regularity conditions, the model performs as well as if the true model is known. They showed that the strong heredity interaction model (SHIM) algorithm has the oracle property. Under further conditions, the same can be shown for the case when both the sample size and the number of predictors tend to infinity.

Other recent methods have also been shown to achieve the oracle property of Fan and Li under certain conditions. For example, for a suitable choice of regulatory parameters the adaptive lasso results are consistent in variable selection and are asymptotically normal ([52]) .

SHIM, like SCAD, is nonconvex. There is no guarantee that the global minimum is reached rather than a local minimum. To investigate the severity of the problem, Choi et al. supply different starting values based on least squares estimates of bootstrap resamples of the data and estimate the difference in the resulting coefficients. They are evaluating whether the starting values result in different local minima. There is no guarantee however whether starting values which are far off the truth will result in good estimates. We have to assume that we are starting with reasonably good starting values for models like this. In the general case of $p < n$ we can estimate all of the parameters with a full least squares fit. However, if the number of predictors is larger then such a model cannot be fit and alternate fitting techniques need to employed.

On the other hand, it should be noted that the fused LASSO problem is strictly convex and thus has a unique solution.

Bien et al. propose a LASSO-like procedure [2] that produces sparse estimates for the main effects and all two-way interactions, while satisfying heredity constraints. Instead of employing group LASSO penalties, they add a set of convex constraints

to the LASSO model. A related idea is presented by Yuan et al., who propose non negative garrote methods that can naturally incorporate hierarchical structural relationships between variables [51]. They incorporate the structural relationships as linear constraints on the corresponding penalties. This approach allows them to incorporate a variety of such structural relationships between predictors.

For the search of gene-gene and gene-environment interactions, Park and Hastie suggest the use of penalized logistic regression [36]. In particular they employ an ℓ_2 penalty which regularizes the coefficients as in ridge regression. The authors argue that quadratic regularization is a good fit for large models with high-order interactions, as collinearity is not a problem.

Stepwise model building using Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC), two closely related procedures, can be adapted to the search of interactions [1, 41]. Both methods apply a penalty term to the number of parameters in the model to avoid overfitting. To chose between two models with BIC, we calculate $\text{BIC} = -2\ln(L) + k\ln(n)$ for the likelihood function L, sample size n and number of parameters k and chose the one with lower BIC value. For AIC we calculate $\text{AIC} = -2\ln(L) + 2k$. To apply AIC and BIC to the interaction model setting we can start with a model consisting of all main effects and build in a forward stepwise fashion models, considering possible interaction terms to be added to the model. We note that, when the number of parameters is fixed, BIC is consistent while AIC is not [6].

An alternative to the methods above is Bayesian model averaging [33]. It is a technique designed to account for uncertainty inherent in the process of model selection. It averages over many different models and thus incorporates model uncertainty in the parameter estimation. One approach to implementing BMA is to average over sets of parsimonious, data-supported models when an exhaustive search is impractical, one of several difficulties of this approach.

1.5 Organization of this Dissertation

The rest of this work is organized as follows. In Chapter 2 we propose the dedicated boosting algorithm and present results for the WHI data and an array of simulated scenarios. In Chapter 3 we propose the directed LASSO and investigate its behavior in simulations. We close in Chapter 4 with a discussion and plans for future work.

Chapter 2

DEDICATED BOOSTING

In regression problems in which the main effects are already established (as is the case with risk factors as well as SNPs for many diseases) or in which their estimation is not a priority, we propose the use of a dedicated boosting algorithm.[1]

Dedicated boosting is a variation of the usual ℓ_2 boosting procedure which focuses on the interaction search, in contrast to most boosting methods which address overall model prediction or classification. To be able to focus on the interaction space, the main predictors are regressed out of the response variable and the interactions. The usual ℓ_2 boosting procedure is then applied to the resulting residuals. Because the effect modifiers may have small signal compared to the main effects, focusing model selection in a space which is orthogonal to the main predictors allows improved performance of the algorithm as compared to applying the usual boosting algorithm which combines both main effects and interactions as learners.

Dedicated boosting is a method to build a model consisting of an ensemble of interactions with potentially small effects. The group of interactions is treated as a profile. The individual membership of factors in this profile is considered only suggestive as the method does not establish significance for the individual interactions but rather investigates the ensemble as a whole. We expect to have the greatest improvement in performance in higher dimensional settings with many interactions and main effects.

The already established marginal effects can be estimated on the data at hand as described above or prior estimates can be employed.

---

[1] A previous version of this chapter is published in Statistics in Medicine [37]

2.1 ℓ_2 *boosting*

We first describe the usual ℓ_2 boosting algorithm with component-wise linear least squares as the base procedure [12, 10, 11]. The algorithm iteratively refits the residuals at each step and performs a linear least squares regression against the single best predictor variable.

For a continuous outcome Y and a potentially large set of predictors X_j, the ℓ_2 boosting algorithm can be summarized as follows (following [11]):

1. Initialize $\widehat{f}^{(0)} = \bar{Y}$ and set $k = 0$, let ν be a small fixed number.

2. Iterate the following steps:

   (a) Increase k by 1. Compute the vector of residuals $R^{(k-1)} = Y - \widehat{f}^{(k-1)}(X)$ for all observations i.

   (b) Fit a simple linear regression for each X_j to the residual vector $R^{(k-1)}$ and set

   $$\widehat{g}^{(k)} = \widehat{\beta}_b X_b,$$

   where

   $$b = \underset{1 \le j \le J}{\arg\min} \sum_i (R_i^{(k-1)} - \widehat{\beta}_j X_{ij})^2.$$

   (c) Update $\widehat{f}^{(k)} = \widehat{f}^{(k-1)} + \nu \widehat{g}^{(k)}$.

3. Stop when $k = k_{stop}$.

The value k_{stop} is determined via a cross-validation estimate of the mean squared error of $(Y - \widehat{f}^{(k)})$.

The boosting estimator $\widehat{f}^{(k)} = \nu \sum_{k=0}^{k} \widehat{g}^{(k)}$ is the sum of the base procedures scaled by ν. The scalar ν is a shrinkage parameter, used to avoid over-fitting. In general, good results are achieved with small ν, but the procedure is relatively insensitive to

the size of ν. Of course, smaller ν will require the algorithm to use a larger number of iterations.

In Step 2(b) we select the predictor at iteration k in the simple linear model setting, which implies that we pick the predictor X_j which is most highly correlated with the residuals $R^{(k-1)}$ from iteration $k-1$. Note that the predictors X_j used at consecutive steps can be the same or different (thus formally we should add an additional superscript ^{k} to X_j, which we omit for simplicity). In the remainder we assume that the candidates X_j are the same at each step; in some applications the X_j are changing during the procedure, for example when splines or regression trees on the X_j are considered.

The fitted function is updated in a linear fashion; as the number of steps of the algorithm gets large the estimates converge to the least squares solution. The coefficient estimates are added at each iteration as well; the coefficient associated with the X_b at that step is the only one updated.

2.2 Dedicated boosting

For ease of notation we will assume that we are looking for an environmental interaction effect that may depend on multiple environmental variables $E_i = (E_1, \ldots, E_p)$ that modify the effect of a genetic single nucleotide polymorphism (SNP) on a regression outcome Y.

Let Y be an $n \times 1$ continuous response vector and G an $n \times 1$ vector be a SNP of interest. Let E be an $n \times p$ matrix of environmental variables. Let the matrix of potential interactions $I = G \times E$. We refer to $M = (G, E_1, \ldots, E_p)$ as the set of main effects and $I = (I_1, \ldots, I_p)$ as the set of interactions. We start by standardizing all continuous environmental variables to mean 0 and variance 1 prior to constructing the matrix of interactions with categorical variables transformed to 0/1. Results are transformed later back to the original scale. To be able to focus on the interaction space, the main effects are regressed out of both the response variable and the

interactions upfront. The ℓ_2 boosting procedure is then applied to the resulting residuals, using the residuals of I as the predictors. In particular, the dedicated boosting procedure is now:

1. Regress the main effects out of the outcome Y and the interaction terms I

$$Y = \sum_{j=1}^{p+1} \widehat{\alpha}_j M_j + res(Y), \tag{2.1}$$

$$I_1 = \sum_{j=1}^{p+1} \widehat{\gamma}_{j1} M_j + res(I_1), \tag{2.2}$$

$$\dots$$

$$I_p = \sum_{j=1}^{p+1} \widehat{\gamma}_{jp} M_j + res(I_p), \tag{2.3}$$

   where the notation $res(Z)$ is used to indicate the residuals of the regression model with Z as response and the main effects M as predictors. These models are fit using ordinary least squares.

2. Apply the ℓ_2 boosting procedure with outcome $res(Y)$ and predictor set $res(I_1)$, $\dots, res(I_p)$. In particular, let

$$res(Y) = \sum_{j=1}^{p} \widehat{\beta}_j^{(k)} res(I_j) + residuals,$$

   and let $\widehat{\beta}^{(k)}$ be the coefficients from the boosting procedure.

Then the fitted values of the whole boosting algorithm can be retrieved by adding $\sum_{j=1}^{p} \widehat{\beta}_j^{(k)} res(I_j)$ to $(Y - res(Y))$, so that the fit of the dedicated boosting solution can be expressed as

$$\sum_{j=1}^{p+1} \widehat{\alpha}_j M_j + \sum_{i=1}^{p} \widehat{\beta}_i^{(k)} \left(I_i - \sum_{j=1}^{p+1} \widehat{\gamma}_{ji} M_j \right).$$

We see that the interaction coefficients are identical to the boosting coefficients $\widehat{\beta}^{(k)}$. Because we applied boosting to the residuals, the main effect coefficient for M_j becomes $\widehat{\alpha}_j + \sum_{i=1}^{p} \widehat{\beta}_i^{(k)} \widehat{\gamma}_{ji}$.

Evaluating interactions

To assess the fit of an interaction model we propose focusing on the interaction part directly. We want to focus on the interaction part of the model, since the residual sums of squares will often be overwhelmed by the main effects and thus the quality of the interaction fit will not be fairly assessed. Existing methods like the F-test to evaluate interactions compare the log likelihoods of a model with and without interactions, to decide if interactions significantly improve the model with only main effects. We define the measure we use, MIaSE, in Section 2.4.

2.3 WHI data

The Women's Health Initiative (WHI) is a long-term national health study that focuses on strategies for preventing chronic diseases, such as heart disease, breast and colorectal cancer and fracture in postmenopausal women. The WHI consisted of an observational study of 93,773 postmenopausal women and four clinical trials studying various interventions in 68,035 postmenopausal women [42]. Participants were recruited between 1992 and 1998. The active intervention of the clinical trials was stopped between 2002 and 2005 (e.g. [50],[49]). Follow-up of subjects is ongoing.

At time of enrollment in the study, extensive environmental exposure data on WHI participants were collected. Blood collection also took place. Using the DNA extracted from this blood collection, a number of genetic studies among WHI participants were initiated.

Population Architecture using Genomics and Epidemiology (PAGE) is a National Human Genome Research Institute (NHGRI) funded consortium that includes WHI, the Multi Ethnic Cohort, Causal Variants Across the Life Course (CALiCo, a consortium of five cardiovascular cohorts), and Epidemiologic Architecture for Genes Linked to Environment (EAGLE, which studies the NHANES cohort). As part of PAGE tens of thousands of subjects are genotyped for SNPs that were identified as genome-wide

significant in other studies ("putative causal SNPs") to study the genetic architecture of the phenotypes for which the SNPs were identified. Each of the four PAGE groups genotyped a number of SNPs associated with obesity or body mass index (BMI).

We analyze the WHI-PAGE data on obesity. This consists of 11 SNPs previously identified, mostly in GWAS studies, to be associated with obesity. Genotype, demographic, and environmental data assumed to be associated with obesity and collected at recruitment are available on 17,049 women. These data include age, current exercise (expressed as METs/week, a continuous variable), whether the subject exercised at each of age 18, 35, and 50 years (binary), education (eleven levels, treated as continuous), ever smoking (binary), current smoking (binary) and alcohol consumption (five levels, treated as continuous), ethnicity (Caucasian, African American, Hispanic, Asian/Pacific Islander, American Indian), region (three levels corresponding to North-South, as a surrogate for sun (vitamin D) exposure), and estimated percent of calories from fat, protein, and carbohydrates based on food-frequency questionnaires. The response is measured BMI (weight in kilograms divided by height in meters squared).

We present results for linear regression, stepwise model building using AIC and BIC model selection (described below), the LASSO, globalboosttest, and dedicated boosting. The data are randomly divided into a training set with 13,049 subjects and a test set with 4,000 subjects. For each of the 11 SNPs, each method is applied to the training data set which contains a specific SNP, all the environmental and demographic variables, and the interactions between the SNP and the other variables. We reserve the test set for evaluating the performance of the models. With the exception of the three SNPs located at the *FTO* gene, the linkage disequilibrium as measured by the absolute value of the correlations between the SNPs is less than 0.12. The three *FTO* SNPs are in high linkage disequilibrium with correlations between 0.78 and 0.89.

To ensure comparability across methods, the main effects of all variables are included (unpenalized) in each method. The AIC and BIC model selection is done in

a forward fashion starting with the main effects model and adding the interaction effects one at a time. A penalization for the LASSO is applied only to the interaction terms, ensuring that all main effects are included in the final model. For dedicated boosting and LASSO we standardize the continuous predictors to mean zero and variance one. All results are back-transformed and presented on the original scale. For the simulations presented in Section 2.4, we also apply an AIC procedure which honors model heredity constraints. In other words, interactions are considered only once both main effects have been selected by the stepwise algorithm to be included in the model. Results for BIC with heredity constraint procedure are not presented since very rarely was an interaction term selected.

Based on our initial experiments we concluded that, like for the regular boosting algorithm, the value of ν is mostly irrelevant, as long as it is small enough. Therefore, we took $\nu = 0.1$ throughout.

We started our analysis by applying the dedicated boosting algorithm for each of the SNPs, as well as to versions of the data with the response permuted. We use the number of steps is a surrogate for the complexity of the model and the amount of signal in the data. When comparing the number of steps that the dedicated boosting algorithm took on the real data, as selected with cross-validation, with the number of steps it took on the permuted data, it appeared that for SNP rs10938397 in the GNPDA2 gene there was evidence of some possible interactions. For SNP rs17782313 in the MC4R gene there were maybe some interactions, but these interactions appeared to be weaker. In our analysis we focus on these two SNPs, providing some limited results for the other nine SNPs.

The interactions found by the dedicated boosting algorithm between rs10938397 and age, current exercise and exercise at 18, and Asian/Pacific Islander ethnicity (see Table 2.1) have a negative association with BMI, while the interactions with percent calories from protein in the diet, education, smoking, and Hispanic, African American and American Indian ethnicity have a positive association. For exercise at 18, educa-

tion level, Hispanic and American Indian ethnicities, the interactions are in the opposite direction of the main effects, while the rest of the selected interactions strengthen the corresponding main effects. We note that the magnitudes of the coefficients from the dedicated boosting algorithm are smaller than those from (unpenalized) linear regression and stepwise model selection using AIC. The LASSO coefficients are neither consistently smaller nor bigger than those of the boosting algorithm. The BIC method selects no interactions for this data set while the globalboosttest algorithm selects only one interaction term.

In Table 2.2 we present results for SNP rs17782313. We again note that for those variables where AIC and boosting selected the same terms, the boosting coefficients are smaller than the AIC coefficients. For this SNP, the group of variables selected by dedicated boosting include age, current exercise, exercise at 18 and 35 years of age, percent calories from carbohydrates in the diet, smoking, and Hispanic and African American ethnicity. Of these, smoking and African American ethnicity are in the opposite direction of the corresponding main effects.

Table 2.3 summarizes for each of the 11 SNPs the performance of each of the models. It also includes the minor allele frequencies of each of the SNPs included in the study. We compute the vector $U = \sum_{j=1}^{18} \widehat{\beta}_j \text{res}(I_j)$, where $\widehat{\beta}$ is the set of estimated interaction terms for the model and $\text{res}(I_j)$ are the residuals left from regressing the main effects out of interaction term I_j in the test data set (see (2.2)-(2.3)). We compute $\text{res}(Y)$ (2.1), the test set BMI residual vector after regressing out the main effects, and the residual sums of squares $RSS = \sum_{i=1}^{4000}(\text{res}(Y_i) - U_i)^2$. We report $RSS - RSS_{main}$, the residual sums of squares less the residual sums of squares of the main effects model. We compute this quantity for a random split of the data in a test set of 4,000 subjects and a training set of 13,049 subjects and nine random splits with the same division and average the resulting $RSS - RSS_{main}$ over all ten splits.

As far as test set RSS is concerned, **globalboosttest** and dedicated boosting have the best performance (Table 2.3), but dedicated boosting identifies slightly more

interactions that appear real. **globalboosttest** identifies some interactions but also misses some. In fact, we will see later in the simulation study that **globalboosttest** has fewer true positives and fewer false positives. For SNP rs17782313, the lowest error is achieved with the BIC model selection, which selected no interactions for any of the splits. This would signify that even though we have some evidence that dedicated boosting is selecting interaction terms that are associated with the outcome, these interactions are not strong enough to improve the predictive properties of the model.

2.4 Permutation Test

Next we discuss the results of a permutation test for SNPs rs10938397 and rs17782313. We permuted the response variable BMI 1000 times after the main effects were regressed out to generate data under the null hypothesis of no interaction effects. Each time we applied the dedicated boosting algorithm using the permutation of BMI as response variable. Note that this is not a typical global permutation test, as we are only removing the interactions, rather than removing both main effects and interactions.

Table 2.4 summarizes the results for SNP rs10938397. For each of the covariates that were selected by the dedicated boosting algorithm in the original analysis, we count how often the variable is selected during the 1000 permutations, and, if it is selected, whether the absolute value of the coefficient $\widehat{\beta}$ is at least as large in the permuted data as in the original version. We do the same for the variables that were not selected, except that here if a variable is selected during the permutations, its coefficient is larger in magnitude than the original analysis, since in that case the coefficient was zero.

With the exception of Hispanic ethnicity, the number of permutation models which included a larger coefficient than the original coefficient were less than or equal to 50. The Hispanic ethnicity interaction term had a larger coefficient in 121 of the permuted data samples. This suggests that if there were no true interactions for this

SNP, as is the case for the permuted data sets, results from the dedicated boosting model would be unlikely to be observed for all covariates that were selected except for Hispanic ethnicity. On the other hand, for all the covariates that were not selected in the original model, the analysis of the permuted data sets frequently selected a larger coefficient.

We also note that in none of the 1000 permutations the boosting algorithm took as many steps as the algorithm took on the original data. This suggests that the dedicated boosting algorithm indeed found a "signal" that is beyond noise.

Table 2.5 presents the permutation results for SNP rs17782313, organized the same way as Table 2.4. The interactions for exercise and exercise at age 35 resulted in coefficients more extreme than the original in more than 50 of the permutations, suggesting that these covariates may have ended up by chance in the original model. The rest of the interactions had coefficients large enough to make them unlikely if there were truly no effect modifications present for this SNP.

In 14 out of the 1000 permutations the dedicated boosting algorithm took as many steps or more as the algorithm took on the real data. This suggests that there likely is a true interaction effect for this data, but that the signal is not as strong as for rs10938397.

2.5 Simulation study

We conducted a simulation study to further examine the performance of dedicated boosting based on the WHI data. In particular, we simulate only a new response variable based on the main effects and interaction coefficients estimated with dedicated boosting for the outcome BMI. We use the original data set for the prediction variables.

Results are presented for the full least squares model without model selection, and AIC and BIC based forward stepwise model selection of interactions starting with the main effects model. For the LASSO, regularization is applied to the interaction terms

Table 2.1: rs10938397: Comparison of interaction terms chosen by the five methods. The dedicated boosting algorithm took 92 steps. Cells that are labeled "-" mean that a particular approach did not select that variable. Each approach first fits (the same) main effects; "Full" refers to fitting all interaction terms using a linear model; "GlobalB" is the globallboosttest algorithm; "Boosting" is the dedicated boosting algorithm.

	Main Effects			Coefficients for Interaction Effects					
	Estimate	Std. Error	p-value	Full	AIC	BIC	LASSO	GlobalB	Boosting
(Intercept)	40.597	1.928	< 0.001						
rs10938397	0.209	0.082	0.011						
Age	-0.195	0.008	< 0.001	-0.016	-0.018	-	-	-	-0.014
Amount of exercise	-0.066	0.005	< 0.001	-0.013	-0.013	-	-0.009	-	-0.010
Exercise at 18	1.387	0.138	< 0.001	-0.358	-0.318	-	-0.227	-	-0.215
Exercise at 35	0.345	0.147	0.019	0.074	-	-	-	-	-
Exercise at 50	-0.518	0.134	< 0.001	-0.067	-	-	-0.005	-	-
% Calories from carbo.	-0.007	0.017	0.665	-0.002	-	-	-	-	-
% Calories from protein	0.183	0.024	< 0.001	0.031	-	-	-	-	0.016
% Calories from fat	0.096	0.019	< 0.001	-0.005	-	-	-	-	-
Education level	-0.359	0.030	< 0.001	0.093	0.091	-	0.041	-	0.060
Ever smoking	0.401	0.121	0.001	0.278	0.261	-	0.191	-	0.164
Current smoking	-3.153	0.218	< 0.001	-0.093	-	-	-	-	-
Alcohol	-0.612	0.055	< 0.001	-0.007	-	-	-	-	-
Hispanic	-0.329	0.216	0.127	0.263	-	-	0.143	-	0.019
African American	2.532	0.160	< 0.001	0.525	0.469	-	0.467	0.030	0.362
Asian/Pacific Islander	-3.936	0.275	< 0.001	-0.389	-	-	-0.269	-	-0.229
American Indian	-0.603	0.565	0.286	1.336	1.308	-	0.991	-	0.816
Region middle	-0.315	0.144	0.029	-0.080	-	-	-	-	-
Region south	-0.361	0.137	0.008	-0.069	-	-	-	-	-

Table 2.2: rs17782313: Comparison of interaction terms chosen by the five methods. The dedicated boosting algorithm took 63 steps. Cells that are labeled "-" mean that a particular approach did not select that variable. Each approach first fits (the same) main effects; "Full" refers to fitting all interaction terms using a linear model; "GlobalB" is the globallboosttest algorithm; "Boosting" is the dedicated boosting algorithm.

	Main Effects			Coefficients for Interaction Effects					
	Estimate	Std. Error	p-value	Full	AIC	BIC	LASSO	GlobalB	Boosting
(Intercept)	40.730	1.927	< 0.001						
rs17782313	0.185	0.094	0.049						
Age	-0.195	0.008	< 0.001	-0.034	-0.035	-	-	-	-0.018
Amount of exercise	-0.066	0.005	< 0.001	-0.009	-	-	-	-	-0.004
Exercise at 18	1.382	0.138	< 0.001	0.218	0.327	-	-	-	0.123
Exercise at 35	0.352	0.147	0.017	0.166	-	-	-	-	0.075
Exercise at 50	-0.517	0.134	< 0.001	0.048	-	-	-	-	-
% Calories from carbo.	-0.008	0.017	0.656	-0.010	-0.019	-	-	-	-0.010
% Calories form protein	0.183	0.024	< 0.001	0.015	-	-	-	-	-
% Calories from fat	0.096	0.019	< 0.001	0.006	-	-	-	-	-
Education level	-0.360	0.030	< 0.001	-0.021	-	-	-	-	-
Ever smoking	0.398	0.121	0.001	-0.572	-0.558	-	-	-	-0.368
Current smoking	-3.157	0.218	< 0.001	0.234	-	-	-	-	-
Alcohol	-0.611	0.055	< 0.001	-0.010	-	-	-	-	-
Hispanic	-0.320	0.216	0.139	-0.861	-0.811	-	-	-	-0.352
African American	2.440	0.157	< 0.001	-0.473	-0.441	-	-	0.058	-0.152
Asian/Pacific Islander	-3.984	0.274	< 0.001	0.165	-	-	-	-	-
American Indian	-0.610	0.565	0.280	-0.066	-	-	-	-	-
Region middle	-0.316	0.144	0.028	-0.003	-	-	-	-	-
Region south	-0.361	0.137	0.008	0.026	-	-	-	-	-

Table 2.3: $(RSS - RSS_{main})$ for the 11 SNPs from the WHI-PAGE data based on the 5 examined approaches. Results are averages of ten random test sets with 4000 subjects that were not used in any aspect of the model building or selection; "Full" refers to fitting all interaction terms using a linear model; "GlobalB" is the globalboosttest algorithm; "Boosting" is the dedicated boosting algorithm. In bold is the best performing method for each SNP.

Nearest Gene	SNP	Minor allele freq.	Full	AIC	BIC	LASSO	GlobalB	Boosting
MTCH2	rs10838738	0.297	0.0272	0.0130	0.0000	−0.0006	0.0005	**−0.0017**
GNPDA2	rs10938397	0.387	0.0100	0.0019	0.0096	0.0182	**−0.0015**	0.0013
KCTD15	rs11084753	0.355	0.0058	0.0012	0.0091	−0.0029	0.0030	**−0.0108**
MC4R	rs17782313	0.236	0.0677	0.0534	**0.0000**	0.0010	0.0001	0.0124
NEGR1	rs2815752	0.367	0.0805	0.0551	0.0000	0.0017	**−0.0018**	0.0060
CTNNBL1	rs6013029	0.093	0.0762	0.0433	0.0000	0.0000	**−0.0020**	0.0049
TMEM18	rs6548238	0.155	0.0613	0.0533	**0.0000**	0.0072	0.0026	0.0095
SH2B1	rs7498665	0.355	0.0440	0.0062	0.0128	−0.0003	−0.0011	**−0.0095**
FTO	rs3751812	0.327	0.0360	0.0440	0.0110	0.0085	**0.0039**	0.0050
FTO	rs8050136	0.394	0.0054	−0.0048	0.0000	−0.0049	0.0050	**−0.0051**
FTO	rs9930506	0.378	0.0605	0.0328	0.0000	0.0000	0.0046	**−0.0023**

only while the main effects are left unpenalized. We also investigate AIC with heredity constraints, that is, we start with a null model considering only main effects, and only once both main effects are included in the model is the interaction term associated with them added to the set of active predictors to be considered. Finally we fit globalboosttest and dedicated boosting.

We consider the model

$$Y = \underbrace{\gamma_0 + \gamma_1 G + \sum_{i=2}^{19} \gamma_i E_i}_{\text{main effect}} + \underbrace{\sum_{i=1}^{18} \beta_i \left(E_i \times G\right)}_{\text{interaction}} + \varepsilon$$

[via dedicated boosting]

where

$$\varepsilon \sim N(0, 6.42^2);$$

note that 6.42 is the residual standard variation in the WHI data.

Table 2.4: rs10938397: Results for permutation study based on 1000 permutations of the null. While the dedicated boosting algorithm on the original data took 92 steps, only 95 out of the 1000 permutations had number of steps greater than or equal to 20 and none had number of steps larger than 85.

	Coef	Selected	Larger coef	Smaller coef
Age	−0.014	122	14	108
Amount of exercise	−0.010	126	4	122
Exercise at age 18	−0.215	119	8	111
% Calories from protein	0.016	126	43	83
Education level	0.060	115	5	110
Ever smoking	0.164	120	20	100
Hispanic	0.019	121	121	0
African American	0.362	127	4	123
Asian/Pacific Islander	−0.229	144	50	94
American Indian	0.816	123	20	103
Exercise at age 35		105	105	-
Exercise at age 50		131	131	-
% Calories from carbo.		67	67	-
% Calories from fat		100	100	-
Current smoking		129	129	-
Alcohol		107	107	-
Region middle		127	127	-
Region south		116	116	-

The β coefficients were taken from the dedicated boosting results applied to the outcome BMI and the γ coefficients are the main effects for BMI from a model which includes simultaneously all predictors from Table 2.6. For the interactions there are 10 non-zero coefficients and 8 zero coefficients. In particular, the non-zero coefficients were

$$\beta = (-0.014, -0.010, -0.215, 0.016, 0.060, 0.164, 0.019, 0.362, -0.229, 0.816),$$

for age, amount of exercise, exercise at 18, % of calories from protein, education level, ever smoking, Hispanic, African American, and American Indian ethnicity, and region middle, respectively. The random error is based on the residual variance of the same model.

Table 2.5: rs17782313: Results for permutation study based on 1000 permutations. On the original data, the dedicated boosting algorithm took 63 steps; 14 permutation runs had number of steps greater than or equal to 63.

	Coef	Selected	Larger coef	Smaller coef
Age	-0.018	122	6	116
Amount of exercise	-0.004	132	59	73
Exercise at age 18	0.123	139	47	92
Exercise at age 35	0.075	122	63	59
% Calories from carbo.	-0.010	93	18	75
Ever smoking	-0.368	134	2	132
Hispanic	-0.352	124	27	97
African American	-0.152	130	43	87
Exercise at age 50		130	130	-
% Calories from protein		148	148	-
% Calories from fat		100	100	-
Education level		149	149	-
Current smoking		153	153	-
Alcohol		126	126	-
Asian/Pacific Islander		152	152	-
American Indian		146	146	-
Region middle		135	135	-
Region south		137	137	-

To compare the five methods we compute

$$U = \sum_{j=1}^{18} \widehat{\beta}_j \text{res}(I_j)$$

and compare it to the true linear combination (TLC) of the interactions

$$TLC = \sum_{j=1}^{18} \beta_j \text{res}(I_j),$$

where res(I) represent the residuals from the linear regression models of the main effects on the interaction terms. We report the

$$\text{MIaSE} = n^{-1} \sum (TLC - U)^2, \tag{2.4}$$

an overall measure of the distance between the true and fitted coefficients for each model.

Table 2.6 presents the results from 1000 replications of the simulation model. We note that the dedicated boosting algorithm has the best performance out of all the methods with respect to RSS. For the 10 terms with non-zero β's we report on average how many times the model correctly assigned non-zero coefficients ("True positive"). The dedicated boosting algorithm has the highest proportion of true positives averaged over the 1000 runs. The procedure assigned a non-zero coefficient to the Hispanic variable only 21% of the time. The row "False positive" counts how often one of the eight covariates with zero coefficients was selected. Not surprisingly, the BIC model, which rarely picked any interactions, has the best false positive performance. Dedicated boosting has fewer false positives than the LASSO, but slightly more than AIC. Globalboosttest performs similarly to BIC, with very few false positives and very few true positives.

Further, we investigate the performance of the dedicated boosting algorithm in a range of scenarios, varying from very weak to very strong interaction effects. Figure 2.1 presents the MIaSE based on the same simulation setup as above. However, all of the interaction coefficients are multiplied by a factor between 0.1 and 5. Thus, the coefficients in these models are $a\beta_j$ where a is between 0.1 and 5, and the β_j are the same as above. For these models still a fixed number of the environmental factors (but not all) have interactions. The strength of these interactions varies between very weak and very strong. Results are based on 50 simulations. As expected the BIC model performs very well when the interaction terms are very small, as it in general rarely selects interactions for inclusion in the model. All methods perform very similarly once the interaction effects are large, as essentially every method finds the right model. Boosting outperforms the other methods for a range of values of the multiplier a between 0.75 and 3, which importantly contains $a = 1$ which corresponds to the interaction effects seen in the real data.

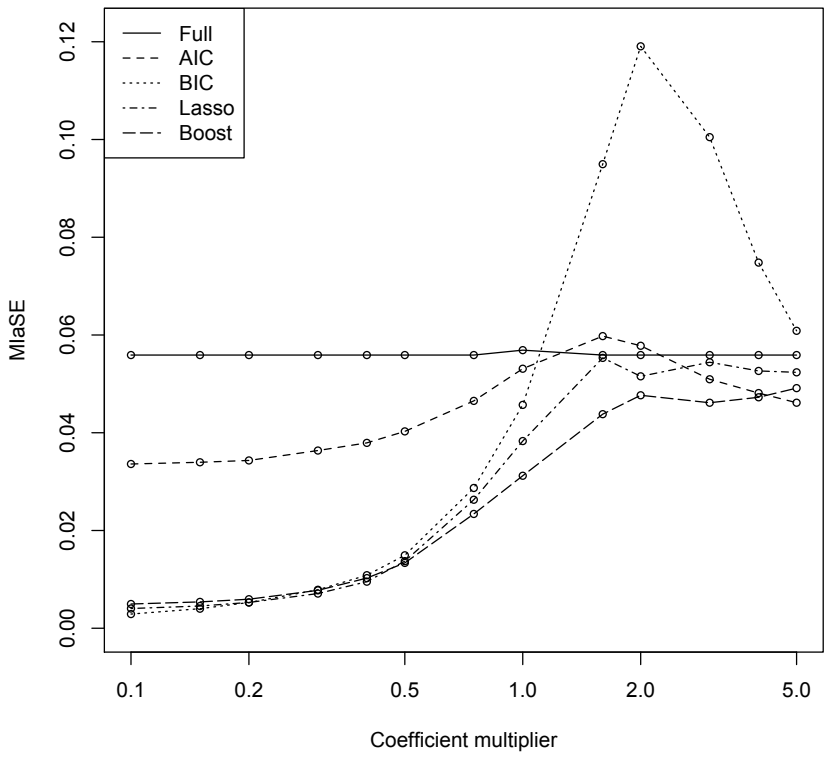

Figure 2.1: Simulation study results based on 50 replications for varying magnitude of interaction terms. "Full" refers to fitting all interaction terms using a linear model; "Boosting" is the dedicated boosting algorithm.

2.6 Extensions

2.6.1 Non-linear functions

A natural extension of the above is to consider non-linear functions in addition to linear functions of the environmental factors. This adds flexibility in modeling parameters and results in an additive model. A further extension would allow adaptive estimation of the basis functions added during the boosting algorithm, such as stepwise selection of regression spline functions. The form of the nonlinear function of a

Table 2.6: Simulation study results based on 1000 replications. Table presents fraction of coefficients from simulation study that are non-zero. "Full" refers to fitting all interaction terms using a linear model; "GlobalB" is the globallboosttest algorithm; "HAIC" is the heredity constraints AIC model; "Boosting" is the dedicated boosting algorithm.

	Full	AIC	HAIC	BIC	LASSO	GlobalB	Boosting
Non-zero coefficients							
Age	1.00	0.46	0.14	0.09	0.09	0.00	0.57
Amount of exercise	1.00	0.49	0.11	0.05	0.47	0.18	0.53
Exercise at age 18	1.00	0.42	0.11	0.02	0.40	0.12	0.44
% Calories from protein	1.00	0.25	0.06	0.01	0.11	0.00	0.28
Education level	1.00	0.50	0.13	0.03	0.22	0.00	0.50
Ever smoking	1.00	0.33	0.08	0.03	0.41	0.10	0.42
Hispanic	1.00	0.16	0.02	0.00	0.29	0.04	0.21
African American	1.00	0.60	0.16	0.11	0.66	0.53	0.66
Asian/Pacific Islander	1.00	0.23	0.06	0.01	0.39	0.13	0.30
American Indian	1.00	0.38	0.01	0.02	0.46	0.19	0.43
Zero coefficients							
Exercise at age 35	1.00	0.22	0.04	0.00	0.25	0.03	0.22
Exercise at age 50	1.00	0.17	0.04	0.00	0.28	0.05	0.24
% Calories from carbo.	1.00	0.26	0.03	0.00	0.06	0.00	0.19
% Calories from fat	1.00	0.26	0.03	0.00	0.10	0.00	0.17
Current smoking	1.00	0.17	0.04	0.01	0.32	0.06	0.23
Alcohol	1.00	0.20	0.05	0.00	0.21	0.01	0.24
Region middle	1.00	0.16	0.02	0.00	0.29	0.03	0.23
Region south	1.00	0.15	0.03	0.00	0.26	0.02	0.22
Overall summary							
MIaSE	0.0570	0.0532	0.0440	0.0456	0.0380	0.0395	**0.0312**
True Positive	1.0000	0.3819	0.0879	0.0369	0.3492	0.1293	**0.4337**
False Positive	1.0000	0.1979	0.0331	0.0033	0.2218	0.0249	**0.2172**

predictor is not pre-specified but is chosen by the boosting algorithm.

2.6.2 Bootstrap confidence intervals for the interaction terms

In this section we propose bootstrap confidence intervals for each of the interaction coefficients in a profile as well as the number of steps. The merit of assessing the

significance of the number of steps comes from our supposition that interactions are likely small and thus on their own might not reach significance. We also investigate the coverage of the bootstrap confidence intervals in select simulation scenarios.

We construct bootstrap confidence intervals for the coefficients of the dedicated boosting model. The idea of bootstrapping a coefficient was introduced by Efron in a 1979 paper ([16]). To construct them we draw a sample of size n with replacement from the original data B times. We apply the dedicated boosting algorithm to each new data set and collect the interaction terms $T^\star$ selected for each bootstrap sample. This yields us a sample of B coefficients. We construct 95% confidence intervals using the $\alpha/2$ and $1-\alpha/2$ quantiles of the the bootstrap samples. The confidence interval (CI) has the form $C_n = (T^\star_{(B\alpha/2)}, T^\star_{(B(1-\alpha)/2)})$. Note that this interval is not symmetrical. There are bias-correction methods for percentile bootstrap confidence intervals ([17]), but for our purposes we will employ the simple percentile bootstrap intervals.

Note that we are constructing bootstrap samples by resampling from the data in its original form and we are regressing the main effects for each sample separately rather than resampling from the original data with the main effects regressed out. This is done so that the independence of main effects and interactions is preserved in each bootstrap sample.

Tables 2.7 and 2.8 present the bootstrap 95% CI for SNP rs10938397 and SNP rs17782313. For all of the interactions, the confidence intervals include 0. This reinforces our original assertion that dedicated boosting identifies a group of environmental factors that is jointly associated with the outcome, but that none of the individual components are associated with the outcome.

Next, we consider bootstrap confidence intervals for the number of steps taken. We develop a bootstrap based 95% percentile CI for the number of steps h and check whether they contain 0. If the CI does not contain 0 we have evidence that dedicated boosting is picking up on signal in the data. We notice that for all the originally

selected interactions, the intervals are bound by zero while for the interactions the method did not select the intervals straddle zero. As we expect, none of the effects are big enough to be significant on their own. For SNPs rs10938397, the number of steps the dedicated boosting algorithm took on the original data set h was 92 and the 95% bootstrap CI for h is $(11, 148)$. For SNPs rs17782313 the original number of steps h is 63 and the 95% bootstrap CI for h is $(9, 148)$. This provides further evidence that there is an interaction effect for these SNPs.

Table 2.7: SNPs rs10938397 dedicated boosting interaction effects and 95% bootstrap confidence intervals.

	Int. Coef	[95% CI]
Age	−0.01	[−0.03, 0.00]
Amount of exercise	−0.01	[−0.02, 0.00]
Exercise at 18	−0.21	[−0.52, 0.00]
Exercise at 35	0.00	[−0.23, 0.24]
Exercise at 50	0.00	[−0.34, 0.13]
% Calories from carbo.	0.00	[−0.01, 0.01]
% Calories fro protein	0.02	[0.00, 0.06]
% Calories from fat	0.00	[−0.02, 0.01]
Education level	0.06	[0.00, 0.14]
Ever smoking	0.16	[0.00, 0.51]
Current smoking	0.00	[−0.44, 0.28]
Alcohol	0.00	[−0.07, 0.08]
Hispanic	0.02	[−0.18, 0.67]
African American	0.36	[0.00, 0.89]
Asian/Pacific Islander	−0.22	[−0.74, 0.00]
American Indian	0.80	[0.00, 2.17]
Region middle	0.00	[−0.29, 0.15]
Region south	0.00	[−0.29, 0.17]

Table 2.8: SNPs rs17782313 dedicated boosting interaction effects and 95% bootstrap confidence intervals.

	Int. Coef	[95% CI]
Age	−0.02	[−0.05, 0.00]
Amount of exercise	0.00	[−0.02, 0.00]
Exercise at 18	0.12	[0.00, 0.52]
Exercise at 35	0.06	[0.00, 0.47]
Exercise at 50	0.00	[−0.21, 0.28]
% Calories from carbo.	−0.01	[−0.03, 0.00]
% Calories fro protein	0.00	[−0.01, 0.06]
% Calories from fat	0.00	[0.00, 0.03]
Education level	0.00	[−0.08, 0.04]
Ever smoking	−0.37	[−0.80, 0.00]
Current smoking	0.00	[−0.16, 0.59]
Alcohol	0.00	[−0.09, 0.05]
Hispanic	−0.32	[−1.28, 0.00]
African American	−0.13	[−0.67, 0.00]
Asian/Pacific Islander	0.00	[−0.06, 0.60]
American Indian	0.00	[−1.13, 1.19]
Region middle	0.00	[−0.31, 0.22]
Region south	0.00	[−0.24, 0.30]

2.6.3 Paired t-tests for stopping criteria

When investigating the behavior of our algorithm based on simulating the null hypothesis we notice that the RSS curves are rather flat and the number of steps which minimize the average of the cross-validated curves would be too away from zero. The particular location of the minimum would be based on random noise in the data. To be able to distinguish better between actual minimums and random bumps we propose a different summary of the curves which takes into account the difference between all preceding steps and the minimum. We calibrate it under a few different null scenarios where we vary the number of effects, the strength of main effects and the random noise in the simulated data sets.

The usual approach is to look at the minimum of each 10-fold cross-validation set

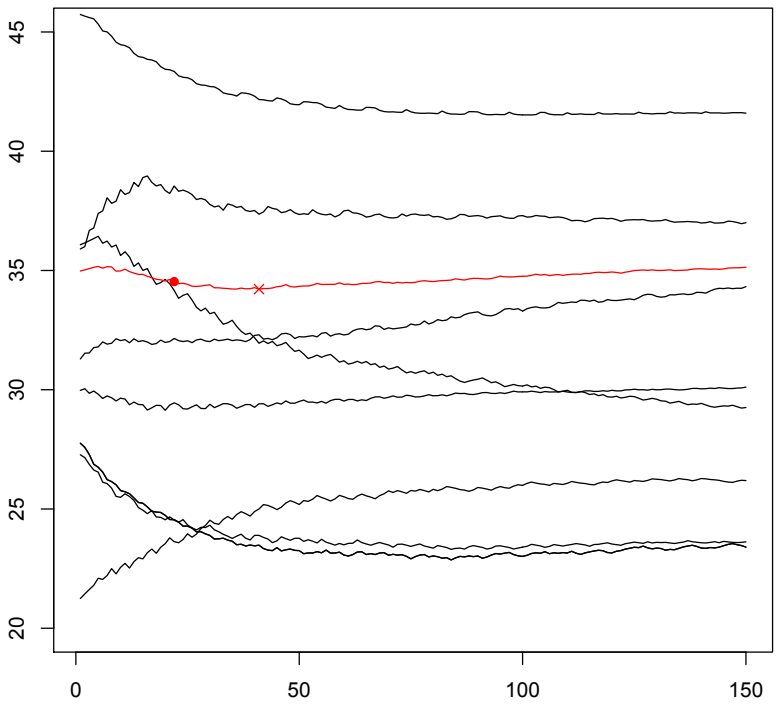

Figure 2.2: Cross-validated curves. The x-axis is the number of steps and the y-axis is the MIaSE for the test set. The cross shows the minimum of the average curves. The solid point shows the t-test selected stopping criteria.

and average it over the 10 runs. For $i = 1, \ldots, 10$ repetitions and $j = 1, \ldots, 10$ cross-validation runs, let $R_{ij}(k)$ be the residual sums of squares for a particular number of steps k in the j^{th} cross-validation set in the i^{th} repetition. Thus, we would choose step k as follows:

$$\frac{1}{10}\sum_{i=1}^{10}(\arg\min_{k}\frac{1}{10}\sum_{j=1}^{10}R_{ij}(k)). \tag{2.5}$$

The 10 repetitions of 10-fold cross validation are used to help reduce variation due to the random nature of the cross-validation splits.

We describe a different way to select the stopping criteria, motivated by the one

standard error rule in CART ([8]). Instead of the algorithm described above, we propose to take the value of each of the cross-validated curves at the minimum of the average of the curves, which we call the $k^\star$. We compute the paired t-test statistic between the values of the 10 curves at $k^\star$ and the values at each of the previous steps. If the test statistic exceeds a pre-specified value $T^\star$ we select the largest step which is significant at the level we have chosen. We choose $T^\star = 1.83$ as the usual t-test cutoff for a test at $\alpha = 0.05$.

The algorithm for selecting stopping criterion takes the following form:

1. $k_i^\star = \arg\min_k \sum_{j=1}^{10} R_{ij}(k)$
2. $T_i(k) = \frac{\sum(R_{ij}(k) - R_{ij}(k^\star))}{\sqrt{var(R_{ij}(k) - R_{ij}(k^\star))}/\sqrt{10}}$
3. $K_i = \min_k\{k : T_i(k) < 1.83\}$
4. $K = \frac{1}{10}\sum_{i=1}^{10} K_i$.

When applying the t-test stopping criterion to the original data set, as expected, the same interactions are selected by the boosting method as with the usual approach (equation 2.5). However, the size of the coefficients is a bit smaller because usually fewer steps are taken by the algorithm. Table 2.9 displays the coefficients for SNPs rs10938397 and rs17782313. These can be compared the the last column of Tables 2.1 and 2.2.

2.7 Extended simulations

In this section, we further investigate the performance of dedicated boosting under a wider range of simulated scenarios. We vary the sample size, the amount of random error, the strength, and number of interactions.

For s = 10, let the environmental factors be distributed as

$$E_1 \dots E_s \sim MVN((0 \dots 0), \Sigma)),$$

Table 2.9: SNPs rs10938397 and rs17782313 dedicated boosting interaction coefficients using the t-test stopping criterion.

	rs10938397	rs17782313
Age	−0.011	−0.012
Amount of exercise	−0.007	−0.003
Exercise at 18	−0.135	0.104
Exercise at 35	-	0.042
Exercise at 50	-	-
% Calories from carbo.	-	−0.009
% Calories fro protein	0.006	-
% Calories from fat	-	-
Education level	0.036	-
Ever smoking	0.094	−0.315
Current smoking	-	-
Alcohol	-	-
Hispanic	-	−0.183
African American	0.267	−0.049
Asian/Pacific Islander	−0.050	-
American Indian	0.419	-
Region middle	-	-
Region south	-	-

with Σ specified below, and let the gene effect G have minor allele frequency 0.3 and be in Hardy-Weinberg equilibrium: $P(y = 0) = 0.49$, $P(y = 1) = 0.42$, $P(y = 2) = 0.09$. We generate data from the model

$$Y = \alpha G + \beta^T \mathbf{E} + \gamma^T \mathbf{E} G + \sigma\epsilon,$$

with $\epsilon = N(0, 1)$ and $\sigma = 6$ or $\sigma = 15$.

The sample size is either $N = 100$ or 1000. The coefficients and correlations are as follows: $\alpha = 2$;

$\beta_{1\ldots k} = (7, 2, 1, 1, 1, 0, \ldots, 0)^T$;

Model A: $\gamma = (2, 2, 2, 1, 1, 0, \ldots, 0)^T$; Model B: $\gamma = (2, -2, 2, -1, 1, 0, \ldots, 0)^T$; Model C: $\gamma = (7, 2, 2, 1, 1, 0, \ldots, 0)^T$; Model D: $\gamma = (14, 4, 4, 2, 2, 0, \ldots, 0)^T$;

Model E: $\gamma = (14, -4, 4, -2, 2, 0, \ldots, 0)^T$. The correlation between environmental predictors is defined as

$$\Sigma_{s\times s} = \begin{pmatrix} 1 & 0.4 \ldots 0.4 & 0 \ldots 0 \\ 0.4 \ldots 0.4 & 1 & 0 \ldots 0 \\ 0 \ldots 0 & 0 \ldots 0 & 1 \end{pmatrix}$$

or $I_{s\times s}$.

Boxplots 4.1 through 4.12 present the results over 1000 replications of the simulated scenarios as a visual representation of the summaries in Tables 2.10 through 2.15. On the y-axis we have plotted the MIaSE measure, as defined in Section 2.4. Through these simulations we use the t-test stopping criterion to determine the number of steps for our model. We note that dedicated boosting performs better with smaller sample size and smaller interaction effects, which are harder to capture for the other methods. However, as expected, when the sample size is big or the effects are big the performance is not better, since all methods identify the interactions. There is not much difference between the results of Models A and B and the results of Models D and E. In the latter models where a large effect is combined with a large sample size, the best performance is obtained by the AIC and BIC model selection procedures.

When the interaction effects are large we expect that both boosting and the LASSO will not perform well. This is due to the fact that both of these methods shrink the larger coefficients. Thus some bias is introduced when large coefficients are involved in the true underlying model. In such cases (Model D and E) all methods perform similarly. A better approach, however, would be a method which allows different weights for the different coefficients, for example a method based on the adaptive LASSO.

Tables 2.16 through 2.21 show the average true positive and false positive for each of the algorithms in each of the simulated scenarios. Tables 2.10 through 2.15 show the performance of the methods in all scenarios as captured by MIaSE. In all these

Table 2.10: MIaSE (SE) for Null Model over 1000 replications. "Full" refers to fitting all interaction terms using a linear model; "Boost" is the dedicated boosting algorithm.

	Full	AIC	BIC	LASSO	Boost
$n = 100$					
$I, \sigma = 6$	4.82	2.89	1.13	0.47	**0.28**
	(0.091)	(0.081)	(0.063)	(0.041)	(0.023)
$I, \sigma = 15$	29.77	18.70	7.27	3.43	**1.74**
	(0.564)	(0.518)	(0.369)	(0.303)	(0.143)
$\Sigma, \sigma = 6$	4.64	2.91	1.19	0.42	**0.28**
	(0.085)	(0.080)	(0.060)	(0.035)	(0.021)
$\Sigma, \sigma = 15$	29.79	18.71	8.06	3.34	**1.80**
	(0.565)	(0.511)	(0.423)	(0.301)	(0.150)
$n = 1000$					
$I, \sigma = 6$	0.37	0.21	0.03	0.04	**0.01**
	(0.005)	(0.006)	(0.003)	(0.003)	(0.001)
$I, \sigma = 15$	2.29	1.29	0.18	0.23	**0.07**
	(0.033)	(0.034)	(0.019)	(0.018)	(0.007)
$\Sigma, \sigma = 6$	0.38	0.23	0.03	0.04	**0.02**
	(0.006)	(0.006)	(0.003)	(0.003)	(0.002)
$\Sigma, \sigma = 15$	2.29	1.35	0.17	0.26	**0.09**
	(0.034)	(0.035)	(0.019)	(0.019)	(0.009)

tables the bold value is the best performing method. In the null model (Table 2.16) the dedicated boosting algorithm does not have the lowest number of false positives however it has the best predictive error (Table 2.10). As expected, the BIC models consistently have the lowest false positive rates and when the number of observations

Table 2.11: MIaSE (SE) for Model A over 1000 replications. "Full" refers to fitting all interaction terms using a linear model; "Boost" is the dedicated boosting algorithm.

	Full	AIC	BIC	LASSO	Boost
$n = 100$					
$I, \sigma = 6$	4.68	4.56	4.76	3.43	**3.12**
	(0.090)	(0.093)	(0.083)	(0.058)	(0.048)
$I, \sigma = 15$	30.52	22.50	12.84	7.59	**6.18**
	(0.600)	(0.554)	(0.397)	(0.250)	(0.140)
$\Sigma, \sigma = 6$	4.74	4.57	4.69	**2.94**	3.69
	(0.089)	(0.090)	(0.083)	(0.065)	(0.080)
$\Sigma, \sigma = 15$	30.26	23.14	16.64	10.82	**9.57**
	(0.621)	(0.551)	(0.443)	(0.290)	(0.168)
$n = 1000$					
$I, \sigma = 6$	0.37	**0.31**	0.35	0.34	0.40
	(0.005)	(0.006)	(0.008)	(0.005)	(0.007)
$I, \sigma = 15$	2.28	2.26	3.67	**2.04**	2.30
	(0.032)	(0.037)	(0.045)	(0.035)	(0.038)
$\Sigma, \sigma = 6$	0.36	0.31	0.38	**0.29**	0.40
	(0.006)	(0.006)	(0.007)	(0.005)	(0.008)
$\Sigma, \sigma = 15$	2.32	2.35	3.26	**1.71**	2.37
	(0.033)	(0.038)	(0.041)	(0.030)	(0.048)

is large (n=1000) they have extremely few false positives.

We show the solution path for the dedicated boosting algorithm, the LASSO and AIC and BIC model selecting procedures for one data set simulated under Model B. Figures 2.3 - 2.5 present the results. We would like to point out that there are two as-

Table 2.12: MIaSE (SE) for Model B over 1000 replications. "Full" refers to fitting all interaction terms using a linear model; "Boost" is the dedicated boosting algorithm.

	Full	AIC	BIC	LASSO	Boost
$n = 100$					
$I, \sigma = 6$	4.80	4.66	4.80	3.44	**3.16**
	(0.087)	(0.090)	(0.080)	(0.060)	(0.048)
$I, \sigma = 15$	29.34	21.53	12.53	7.80	**6.17**
	(0.538)	(0.518)	(0.402)	(0.262)	(0.140)
$\Sigma, \sigma = 6$	4.67	4.31	4.00	3.10	**2.77**
	(0.086)	(0.083)	(0.072)	(0.054)	(0.040)
$\Sigma, \sigma = 15$	30.34	21.88	12.07	6.78	**5.49**
	(0.602)	(0.551)	(0.425)	(0.277)	(0.170)
$n = 1000$					
$I, \sigma = 6$	0.37	**0.30**	0.35	0.33	0.38
	(0.006)	(0.006)	(0.009)	(0.006)	(0.007)
$I, \sigma = 15$	2.32	2.34	3.62	**2.04**	2.24
	(0.033)	(0.040)	(0.047)	(0.036)	(0.037)
$\Sigma, \sigma = 6$	0.36	**0.31**	0.42	0.34	0.41
	(0.005)	(0.006)	(0.008)	(0.005)	(0.007)
$\Sigma, \sigma = 15$	2.33	2.34	3.05	2.13	**2.11**
	(0.032)	(0.036)	(0.031)	(0.031)	(0.030)

pects to performance of the methods in simulations. Firstly, there is the performance of the method in fitting the simulated data set. Apart from that, the performance on a test set is also dependent on the choice of tuning parameter. We note that difference between performance seen in our simulations could be due to poor choice of tuning

Table 2.13: MIaSE for Model C (SE) over 1000 replications. "Full" refers to fitting all interaction terms using a linear model; "Boost" is the dedicated boosting algorithm.

	Full	AIC	BIC	LASSO	Boost
$n = 100$					
$I, \sigma = 6$	4.73	4.36	4.07	**3.57**	3.94
	(0.091)	(0.089)	(0.076)	(0.074)	(0.089)
$I, \sigma = 15$	30.22	23.58	17.73	14.67	**13.58**
	(0.594)	(0.592)	(0.510)	(0.371)	(0.264)
$\Sigma, \sigma = 6$	4.92	4.47	4.27	**3.08**	4.51
	(0.090)	(0.089)	(0.084)	(0.067)	(0.121)
$\Sigma, \sigma = 15$	30.21	23.70	18.38	**13.51**	17.03
	(0.566)	(0.542)	(0.498)	(0.337)	(0.346)
$n = 1000$					
$I, \sigma = 6$	0.37	**0.30**	0.35	0.33	0.43
	(0.006)	(0.006)	(0.008)	(0.005)	(0.008)
$I, \sigma = 15$	2.34	2.27	2.98	**2.03**	2.37
	(0.033)	(0.038)	(0.039)	(0.033)	(0.042)
$\Sigma, \sigma = 6$	0.37	0.32	0.39	**0.30**	0.43
	(0.006)	(0.006)	(0.007)	(0.006)	(0.009)
$\Sigma, \sigma = 15$	2.31	2.21	2.69	**1.69**	2.38
	(0.033)	(0.037)	(0.038)	(0.030)	(0.047)

parameters.

Table 2.14: MIaSE (SE) for Model D over 1000 replications. "Full" refers to fitting all interaction terms using a linear model; "Boost" is the dedicated boosting algorithm.

	Full	AIC	BIC	LASSO	Boost
$n = 100$					
$I, \sigma = 6$	4.65	4.36	4.53	**4.01**	5.24
	(0.087)	(0.093)	(0.108)	(0.082)	(0.129)
$I, \sigma = 15$	29.15	25.64	22.06	**19.42**	21.57
	(0.534)	(0.507)	(0.422)	(0.427)	(0.485)
$\Sigma, \sigma = 6$	4.58	4.31	4.41	**3.42**	5.69
	(0.086)	(0.093)	(0.100)	(0.072)	(0.167)
$\Sigma, \sigma = 15$	29.61	25.64	23.05	**17.53**	25.19
	(0.568)	(0.549)	(0.510)	(0.403)	(0.630)
$n = 1000$					
$I, \sigma = 6$	0.37	0.29	**0.20**	0.32	0.52
	(0.005)	(0.005)	(0.004)	(0.005)	(0.010)
$I, \sigma = 15$	2.38	**2.10**	2.68	2.16	2.58
	(0.036)	(0.041)	(0.050)	(0.036)	(0.045)
$\Sigma, \sigma = 6$	0.37	0.29	**0.19**	0.29	0.51
	(0.005)	(0.005)	(0.004)	(0.005)	(0.013)
$\Sigma, \sigma = 15$	2.28	2.04	2.54	**1.83**	2.55
	(0.034)	(0.039)	(0.041)	(0.033)	(0.053)

Table 2.15: MIaSE (SE) for Model E over 1000 replications. "Full" refers to fitting all interaction terms using a linear model; "Boost" is the dedicated boosting algorithm.

	Full	AIC	BIC	LASSO	Boost
$n = 100$					
$I, \sigma = 6$	4.74	4.40	4.61	**4.20**	5.31
	(0.092)	(0.096)	(0.106)	(0.090)	(0.125)
$I, \sigma = 15$	29.51	25.98	22.67	**19.34**	21.57
	(0.551)	(0.525)	(0.460)	(0.410)	(0.496)
$\Sigma, \sigma = 6$	4.86	4.48	4.52	**4.35**	5.34
	(0.092)	(0.094)	(0.096)	(0.091)	(0.118)
$\Sigma, \sigma = 15$	29.97	24.92	19.50	**18.99**	21.15
	(0.544)	(0.527)	(0.471)	(0.436)	(0.497)
$n = 1000$					
$I, \sigma = 6$	0.37	0.29	**0.19**	0.33	0.54
	(0.005)	(0.005)	(0.004)	(0.005)	(0.011)
$I, \sigma = 15$	2.33	**2.00**	2.57	2.06	2.45
	(0.033)	(0.039)	(0.044)	(0.032)	(0.041)
$\Sigma, \sigma = 6$	0.37	0.29	**0.20**	0.34	0.66
	(0.005)	(0.005)	(0.005)	(0.005)	(0.011)
$\Sigma, \sigma = 15$	2.34	**2.10**	2.63	2.17	2.59
	(0.033)	(0.038)	(0.043)	(0.034)	(0.041)

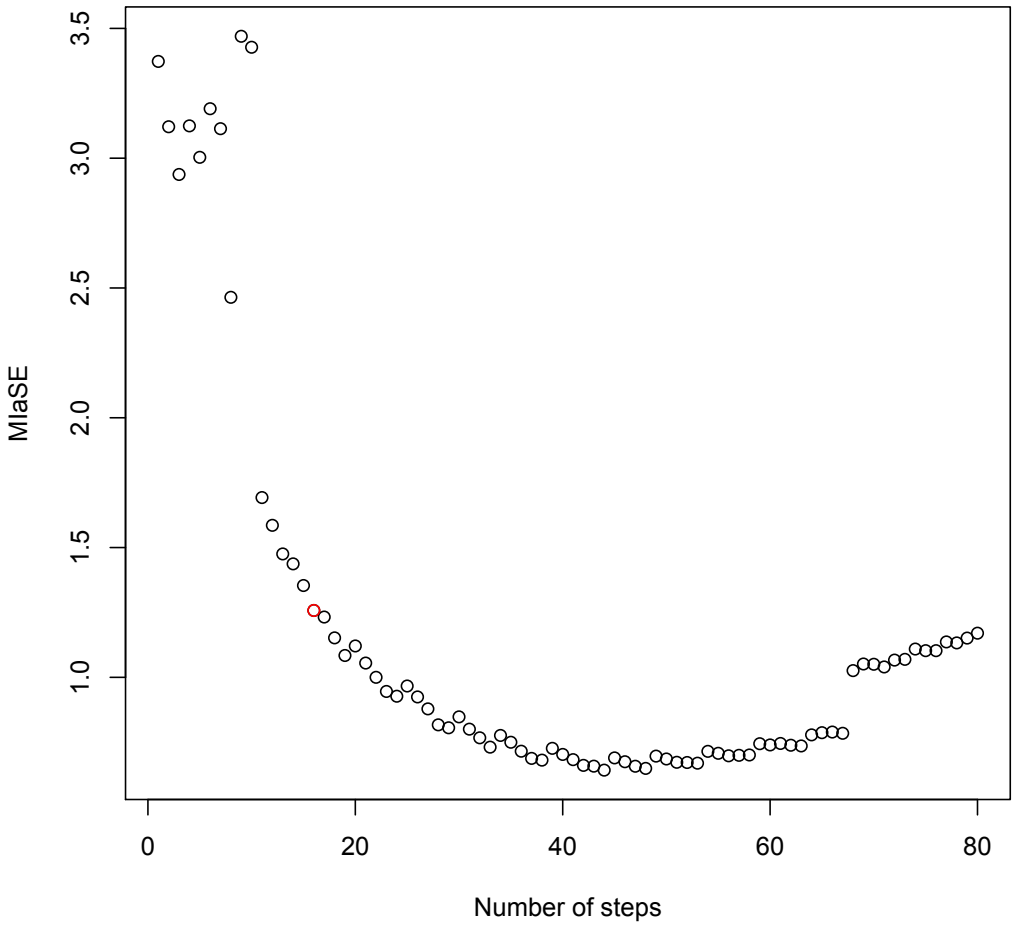

Figure 2.3: Solution path for the dedicated boosting algorithm. The red point shows the number of steps as selected through cross-validation.

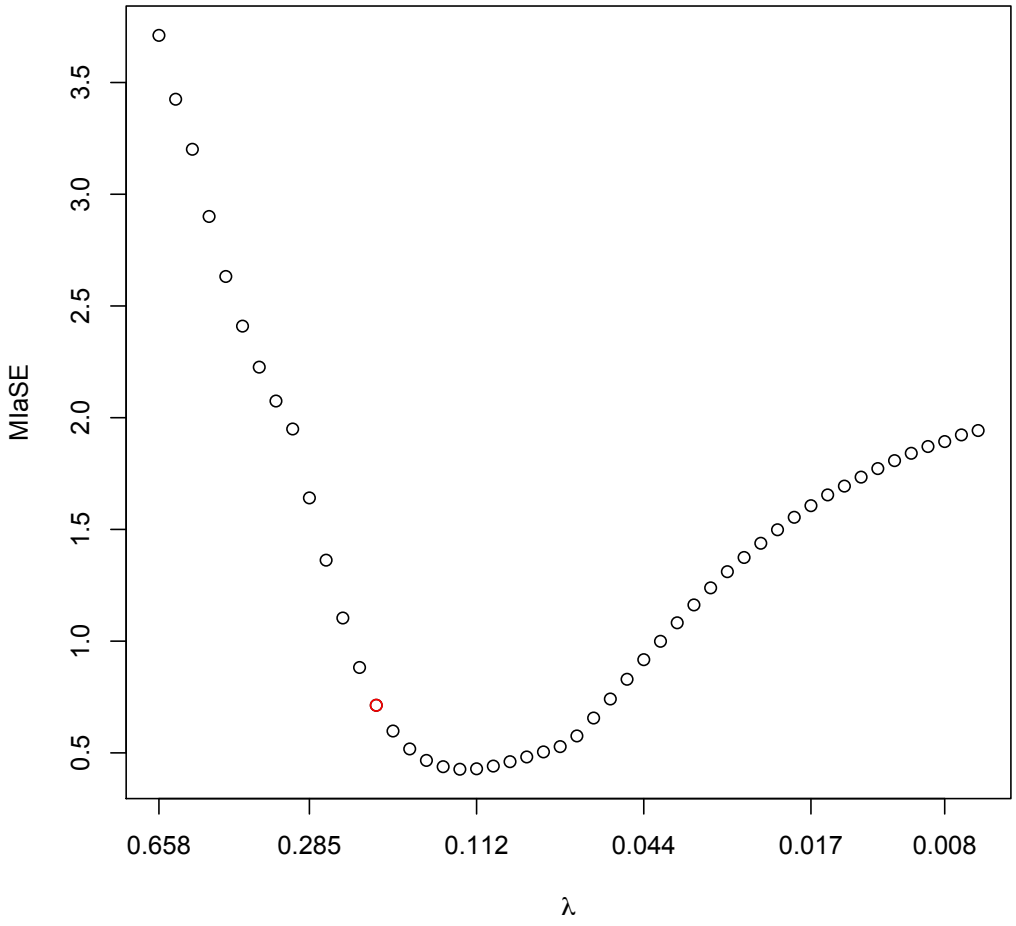

Figure 2.4: Solution path for the LASSO. The red point shows the λ selected through cross-validation.

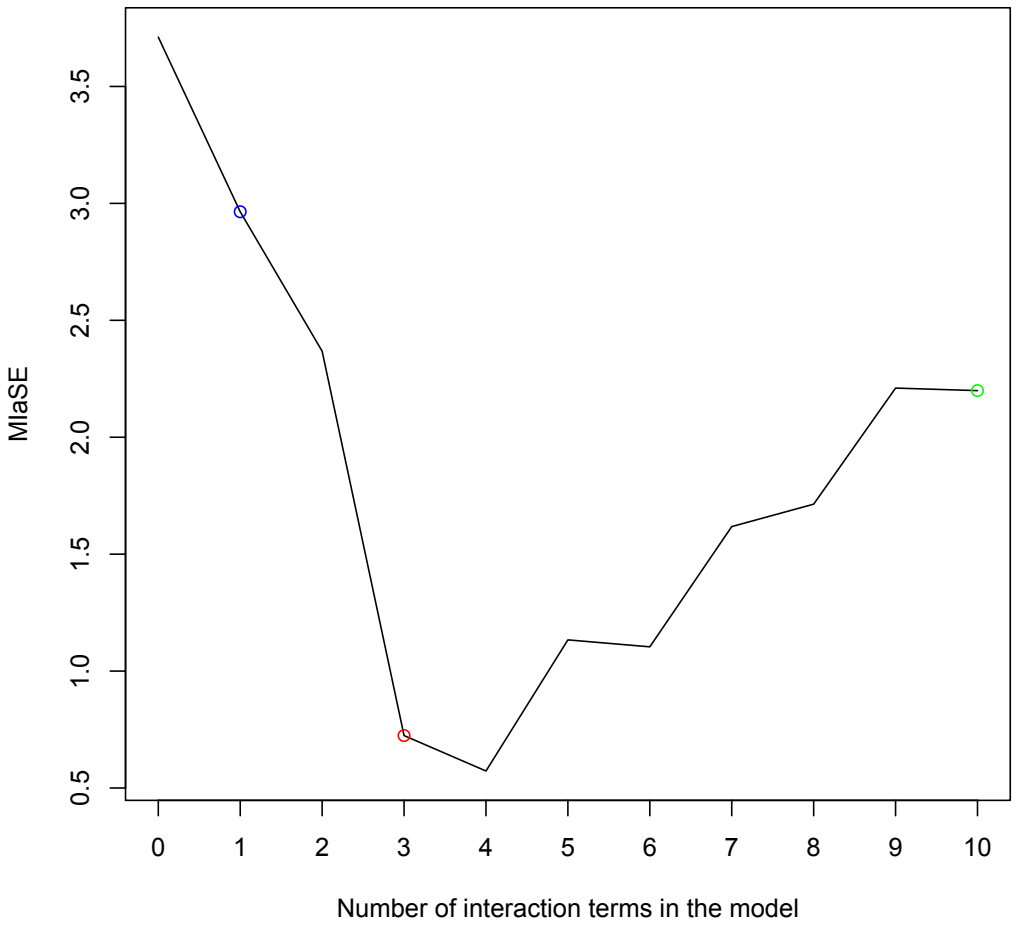

Figure 2.5: Solution path for AIC and BIC. The red point shows the number of interactions selected with AIC; the blue point shows the number of interactions selected with BIC and the green point shows the model with all interactions ("Full").

Table 2.16: Average number of times a coefficient is selected for the Null Model. "TP" is average true positive over the true non-zero coefficients over 1000 simulations. "FP" is average false positive over the true zero coefficients over 1000 simulations.

		Full	AIC	BIC	LASSO	Boost
$n = 100$						
$I, \sigma = 6$	FP	1000	217.0	53.6	112.5	134.6
$I, \sigma = 15$	FP	1000	218.4	54.9	127.2	135.8
$\Sigma, \sigma = 6$	FP	1000	218.1	57.7	104.5	132.3
$\Sigma, \sigma = 15$	FP	1000	222.9	58.8	112.3	125.8
$n = 1000$						
$I, \sigma = 6$	FP	1000	160.7	8.6	102.2	78.7
$I, \sigma = 15$	FP	1000	157.0	9.3	96.3	67.1
$\Sigma, \sigma = 6$	FP	1000	173.2	10.2	105.8	83.7
$\Sigma, \sigma = 15$	FP	1000	170.8	9.1	109.6	87.9

Table 2.17: Average number of times a coefficient is selected for Model A. "TP" is average true positive over the true non-zero coefficients over 1000 simulations. "FP" is average false positive over the true zero coefficients over 1000 simulations.

		Full	AIC	BIC	LASSO	Boost
$n = 100$						
$I, \sigma = 6$	TP	1000	585.2	350.6	577.6	579.8
	FP	1000	216.4	62.6	320.0	265.8
$I, \sigma = 15$	TP	1000	308.6	102.4	192.0	218.0
	FP	1000	223.2	53.8	141.0	163.8
$\Sigma, \sigma = 6$	TP	1000	554.0	406.8	786.0	678.6
	FP	1000	225.2	70.6	336.4	215.0
$\Sigma, \sigma = 15$	TP	1000	331.0	164.8	331.0	282.8
	FP	1000	227.2	64.0	197.6	166.4
$n = 1000$						
$I, \sigma = 6$	TP	1000	988.8	914.6	999.2	995.2
	FP	1000	162.4	8.4	590.4	355.0
$I, \sigma = 15$	TP	1000	726.8	359.0	831.2	757.2
	FP	1000	154.4	9.6	464.8	283.6
$\Sigma, \sigma = 6$	TP	1000	969.8	854.2	996.8	992.8
	FP	1000	157.8	9.4	389.0	208.0
$\Sigma, \sigma = 15$	TP	1000	688.2	446.6	908.0	829.6
	FP	1000	173.4	11.4	365.8	185.8

Table 2.18: Average number of times a coefficient is selected for Model B. "TP" is average true positive over the true non-zero coefficients over 1000 simulations. "FP" is average false positive over the true zero coefficients over 1000 simulations.

		Full	AIC	BIC	LASSO	Boost
$n = 100$						
$I, \sigma = 6$	TP	1000	578.8	349.4	598.0	592.2
	FP	1000	227.2	67.0	350.6	281.6
$I, \sigma = 15$	TP	1000	300.0	94.8	193.0	215.0
	FP	1000	218.4	52.6	152.0	154.8
$\Sigma, \sigma = 6$	TP	1000	503.8	283.4	440.8	445.4
	FP	1000	212.2	66.0	289.0	263.4
$\Sigma, \sigma = 15$	TP	1000	291.2	100.6	162.8	190.8
	FP	1000	224.6	57.8	151.2	176.2
$n = 1000$						
$I, \sigma = 6$	TP	1000	989.8	912.6	998.2	996.0
	FP	1000	164.0	9.2	590.4	347.8
$I, \sigma = 15$	TP	1000	729.4	362.0	846.6	773.0
	FP	1000	167.2	9.8	476.4	297.4
$\Sigma, \sigma = 6$	TP	1000	967.8	805.2	991.0	981.4
	FP	1000	156.6	9.8	678.0	397.6
$\Sigma, \sigma = 15$	TP	1000	635.6	252.4	692.4	647.0
	FP	1000	169.6	7.4	462.8	327.6

Table 2.19: Average number of times a coefficient is selected for Model C. "TP" is average true positive over the true non-zero coefficients over 1000 simulations. "FP" is average false positive over the true zero coefficients over 1000 simulations.

		Full	AIC	BIC	LASSO	Boost
$n = 100$						
$I, \sigma = 6$	TP	1000	638.4	458.4	715.4	653.6
	FP	1000	215.2	54.0	396.4	274.4
$I, \sigma = 15$	TP	1000	416.8	231.4	398.0	370.4
	FP	1000	215.8	58.4	237.0	198.8
$\Sigma, \sigma = 6$	TP	1000	617.8	474.0	819.2	718.6
	FP	1000	228.4	66.2	353.6	201.6
$\Sigma, \sigma = 15$	TP	1000	415.4	271.0	530.4	428.2
	FP	1000	218.4	54.6	264.2	176.0
$n = 1000$						
$I, \sigma = 6$	TP	1000	990.0	909.8	999.0	995.2
	FP	1000	167.2	10.0	606.8	346.0
$I, \sigma = 15$	TP	1000	749.6	455.6	872.2	795.4
	FP	1000	164.2	10.4	481.6	291.6
$\Sigma, \sigma = 6$	TP	1000	968.4	850.8	997.2	992.4
	FP	1000	160.8	8.2	375.8	203.2
$\Sigma, \sigma = 15$	TP	1000	709.2	506.6	906.4	833.4
	FP	1000	166.6	9.6	356.6	177.8

Table 2.20: Average number of times a coefficient is selected for Model D. "TP" is average true positive over the true non-zero coefficients over 1000 simulations. "FP" is average false positive over the true zero coefficients over 1000 simulations.

		Full	AIC	BIC	LASSO	Boost
$n = 100$						
$I, \sigma = 6$	TP	1000	875.8	763.0	937.8	905.6
	FP	1000	214.8	60.6	547.4	355.4
$I, \sigma = 15$	TP	1000	571.4	396.4	650.8	569.2
	FP	1000	213.8	61.4	358.8	254.2
$\Sigma, \sigma = 6$	TP	1000	840.8	731.2	954.8	928.2
	FP	1000	222.6	57.6	394.6	239.0
$\Sigma, \sigma = 15$	TP	1000	547.8	410.2	754.4	639.8
	FP	1000	218.2	67.8	335.6	191.2
$n = 1000$						
$I, \sigma = 6$	TP	1000	1000.0	1000.0	1000.0	1000.0
	FP	1000	158.8	9.0	606.4	284.0
$I, \sigma = 15$	TP	1000	956.4	810.0	989.8	978.8
	FP	1000	171.8	11.0	588.4	354.2
$\Sigma, \sigma = 6$	TP	1000	1000.0	999.6	1000.0	1000.0
	FP	1000	161.0	8.0	398.2	203.6
$\Sigma, \sigma = 15$	TP	1000	927.4	775.0	990.0	980.6
	FP	1000	158.4	9.6	373.6	196.6

Table 2.21: Average number of times a coefficient is selected for Model E. "TP" is average true positive over the true non-zero coefficients over 1000 simulations. "FP" is average false positive over the true zero coefficients over 1000 simulations.

		Full	AIC	BIC	LASSO	Boost
$n = 100$						
$I, \sigma = 6$	TP	1000	872.6	760.6	937.4	906.2
	FP	1000	218.4	64.2	557.6	375.2
$I, \sigma = 15$	TP	1000	569.4	392.6	629.2	563.8
	FP	1000	223.2	62.8	345.2	259.8
$\Sigma, \sigma = 6$	TP	1000	821.4	685.8	881.6	842.6
	FP	1000	215.8	59.2	580.2	409.2
$\Sigma, \sigma = 15$	TP	1000	531.0	350.4	550.4	507.4
	FP	1000	215.6	58.0	354.2	273.0
$n = 1000$						
$I, \sigma = 6$	TP	1000	1000.0	1000.0	1000.0	1000.0
	FP	1000	157.4	9.0	611.0	282.4
$I, \sigma = 15$	TP	1000	961.4	816.4	990.6	979.6
	FP	1000	163.2	9.8	603.2	361.2
$\Sigma, \sigma = 6$	TP	1000	1000.0	999.6	1000.0	1000.0
	FP	1000	165.4	12.6	690.8	253.4
$\Sigma, \sigma = 15$	TP	1000	918.8	718.6	967.0	939.6
	FP	1000	164.6	9.4	667.2	403.0

Chapter 3

DIRECTED LASSO: LASSO WITH STRUCTURED INTERACTIONS

A common aspect of studying complex genetic associations, and specifically interactions, is that the power to detect them is usually limited. We believe that using a controlled specification of the interaction models, e.g. forcing a particular functional form, increases the power to test such associations. The idea is to use a structured interaction model, together with penalized regression, in order to limit the model complexity.

Suppose that there is a linear combination of genetic or environmental variables which puts an individual at high or low risk of disease. It would be beneficial to model all of the interactions using a pre-specified model which accounts for the fact that this group of variables modifies the risk in a similar fashion compared to the main effects.

We propose the directed LASSO, a regression modeling strategy using a fused set of basis functions. Let G be a single genetic effect and let E be a matrix in which each column is an environmental factor. We fuse each main effect E and the interaction term GE of this effect with a specific effect modifier into a single basis function. The most restrictive case allows no deviations from the product interaction model, this amounts to the set basis functions $[1, G, (1 + hG)E]$. Using such fused basis functions then decreases the dimensionality of the model. Under the assumption of multiplicative interactions, the parameter h estimates the strength and direction of the interactions in the model relative to the main effects. Note that in this initial formulation, h is global for all the interactions estimated in the model. This is a rather restrictive specification of the model, which we will relax later.

3.1 Motivating Example

We have available data on 208 Diffuse Large B-cell lymphoma patients. Complete data, however, is available only for 161 subjects. The data contains 32 gene expression measurements and treatment information for each patient. We are interested in assessing the possibility of effect modification of the prognostic performance of the genes by the treatment assignment. We would like to find groupings of genetic factors which modify the treatment effect in a similar way.

The outcome of interest is relapse-free survival. As a first step we will consider a yes/no outcome of whether the patient had a relapse for the time followed. We use logistic regression for this outcome.

3.2 Directed LASSO Algorithm

Let Y be a $n \times 1$ continuous response vector, and let the $n \times 1$ binary vector G be a SNP or treatment of interest. Let E be an $n \times p$ matrix of continuous environmental variables which might modify the effect of G on Y. We want to model the interaction terms $G \times E$ relative to the estimated main effects of E. The most restrictive form of the directed LASSO assumes that all the elements of E interact with G in the same way. Only a single parameter h estimates the relationship between the main effects and the interaction effects. This leads to the following algorithm.

First, for a range of values of h we construct a set of basis functions $[1, G, (1 + hG) \times E]$. For each h, the LASSO algorithm is applied to the set of basis functions. The directed LASSO estimates can be defined as

$$\hat{\beta}(\text{directed LASSO}) = \arg\min_{\beta} ||Y - \beta_1 G - \sum_{j=2}^{p} \beta_j (1 + h^B G) E_j||^2 + \lambda \sum_{j=2}^{p} |\beta_j|,$$

where h^B is the "optimal" value of h. This method has two tuning parameters (h, λ). We select both through cross-validation. We use the fact that for each h any algorithm the fits a LASSO will compute estimates for every single λ in a single run. We choose

an optimal λ for a given h by 10-fold cross-validation and then choose an optimal h through another level of cross-validation, effectively performing a two-dimensional grid search over h and λ. (A computationally cheaper approach would select one or both tuning parameters using AIC.) We discuss fitting directed LASSO models in Section 3.7.5.

For a given j, the main effect E_j is estimated by β_j and the interaction $G \times E_j$ is estimated by $h\beta_j$. To ensure that the model satisfies heredity constraints, the coefficient for G is not penalized, and as a result is always included in the final model. The interaction effect is defined in a single basis with the corresponding main effect. Therefore, if the β_j for the basis function is set to 0, both the main effect and the interaction effect are excluded from the model, guaranteeing the heredity constraints with respect to E_j are satisfied. Through the ℓ_1 penalty, the LASSO performs simultaneous regularization and variable selection. Some of the β_j's are set to zero, which excludes the entire basis function, resulting in exclusion of both main effect E_j, and interaction term $G \times E_j$.

3.3 Directed Adaptive LASSO

As an alternative to the directed LASSO we also study the properties of the directed adaptive LASSO. To achieve better performance, we apply the idea of the directed LASSO to the adaptive LASSO algorithm. The adaptive LASSO alleviates the bias in larger coefficients and we expect it to have better performance in the scenarios we investigate. The model for the directed adaptive LASSO then becomes

$$\hat{\beta}(\text{directed adaptive LASSO}) = \arg\min_{\beta} ||Y - \beta_1 G - \sum_{j=2}^{p} \beta_j (1 + h^B G) E_j||^2 + \lambda \sum_{j=2}^{p} \hat{w}_j |\beta_j|,$$

where $\hat{w}_j = 1/|\hat{\beta}_j|^\gamma$ and $\hat{\beta}$'s are root-n-consistent estimates of the true β's. The set of parameters to be estimated is now (h, γ, λ). One could perform three levels of cross-validation to choose the set of parameters, however, this would be very time consuming. A common practice is to set the γ parameter equal to 1, since the size of

γ has not been found to be of major importance. We adopt this practice as well. We choose the h and λ parameters using the procedure described above for the directed LASSO.

3.4 Simulation

We investigate the performance of the directed LASSO and the directed adaptive LASSO as compared to the regular LASSO algorithm in a variety of scenarios.

We simulate training and test data sets of 50 predictors and 1000 observations each with varying number and strength of interaction effects. The 50 predictors we denote with E are independent random $\mathrm{N}(0,1)$. The effect G is Binomial$(n, 0.6)$, independent of E.

$$Y = \underbrace{\beta_0 + \beta_1 G + \sum_{j=2}^{51} \beta_j E_j}_{\text{main effect}} + \underbrace{\sum_{j=52}^{101} \beta_j \left(E_j \times G\right)}_{\text{interaction}} + \varepsilon$$

where

$$\varepsilon \sim N(0,1).$$

The particular interaction models are described in Table 3.1. We investigate sparse models where between one and three interaction effects out of the 50 are non-zero.

The training sets are used to tune the model parameters and build the directed LASSO model, and the test sets are used to test the performance of the model. Table 3.2 presents preliminary results based on 50 replications of the simulation models. The performance of the directed LASSO is compared to the LASSO, the adaptive LASSO and the directed adaptive LASSO. The mean squared error (MSE) on the test set is reported for each method.

For Models 1, 3, and 4, the performance of the adaptive LASSO and the directed adaptive LASSO is very similar, and the best from all the methods compared. Model

Table 3.1: Simulation setup: Model coefficients

	Intercept	G	Main Effects			Int. Effects		
	β_0	β_1	β_2	β_3	β_4	β_{52}	β_{53}	β_{54}
Model 1	1.5	0.2	0.2	0.2	0.2	0.05	0.05	0.05
Model 2	1.5	0.2	0.2	0.2	0.2	0.2	0.2	0.2
Model 3	1.5	0.2	0.2	0.2	0.2	0.1	0	0
Model 4	1.5	0.2	0.2	0.2	0.2	0.05	0	0

Table 3.2: Simulation results: MSE based on 50 replication. "A. LASSO" is the adaptive LASSO; "D. LASSO" is the directed LASSO; "D.A. LASSO" is the directed adaptive LASSO.

	LASSO	A. LASSO	D. LASSO	D.A. LASSO
Interaction model satisfied				
Model 1	0.020	**0.014**	0.018	0.015
Model 2	0.024	0.019	0.018	**0.013**
Interaction model not satisfied				
Model 3	0.021	0.015	0.019	**0.014**
Model 4	0.020	**0.014**	0.019	0.015

1 fits a scenario where our model should perform well, however it has small interaction effects and all methods perform comparably. In Model 2, when the interaction effects are much bigger, the directed adaptive LASSO outperforms the other methods.

In models 3 and 4, the underlying interaction structure does not fit the assumed form in the directed adaptive LASSO model. However, the directed adaptive LASSO still performs on par with the adaptive LASSO and outperforms the regular LASSO and the directed LASSO.

3.5 Extensions

The amount of flexibility in the interaction model can be increased with the expansion of the basis set to include additional interaction terms. The basis function then becomes $[1, G, (1 + hG) \times E, G \times E]$. The interaction terms in the unfused basis receive higher penalization to encourage simpler models. This higher penalization can be achieved by incorporating a weighted penalty function such that weights corresponding to the unfused basis functions are higher than the fused ones. This results in some of the coefficients being the same for the main effects and the interactions, which aids both interpretation and variance reduction, while others are allowed to differ.

A different approach to allow for deviations from the product interaction model would be to apply a separate penalty such as the ℓ_0 penalty to the unfused basis functions. The ℓ_0 penalty directly penalizes the number of nonzero coefficients and thus will help reduce the number of interactions which deviate from the controlled model.

Thus far we have only considered a single h parameter. It is plausible that there are several groups of variables which modify the risk in similar ways. A second way to allow more flexibility is to assign each of these groups a different h parameter which describes the direction and strength of the effect modification for the particular set of variables. We can start with a single h for each variable and then cluster them in groups, thus discovering groupings of variables with similar behavior. The rest of the variables for which effect modification is not present will be clustered in a group with $h = 0$. These groupings may be being useful in the interpretation of the model.

In the following sections we propose a different way to specify the model and add flexibility to the form of the interactions.

3.6 More flexible specification

In this and the following sections we explore more flexible formulations of the model. In particular, we allow multiple h parameters, or in other words multiple groups of factors which modify the treatment effect in a similar manner. We let each interaction differ from its corresponding main effect by a factor of h where there are as many h parameters as there are interactions. We apply a penalty term that groups some of the h's together by penalizing differences between h parameters. This extra penalty encourages natural groupings between interactions which modify the genetic effect G in a similar way.

We use the method described by Choi et al. to estimate the multiple h's together with the rest of the unknown parameters in the model ([14]). In their model SHIM, the authors do not distinguish between types of predictors and consider all two-way interactions between all predictors X. The SHIM model with all pairwise interactions can be expressed in the following form

$$\text{minimize}||Y - g(X)||^2 + \lambda_\beta \sum |\beta_i| + \lambda_h \sum |h_{ij}| \tag{3.1}$$

where

$$g(X) = \beta_0 + \sum_i \beta_i X_i + \sum_i \sum_j h_{ij}\beta_i\beta_j(X_i \times X_j).$$

The interaction coefficient is broken down in three parts, so that the coefficient for $(X_i \times X_j)$ is $h_i\beta_i\beta_j$ and is the product of the main effect coefficients for X_i and X_j and a factor h_i. Since the interaction terms are based on the main effects, interactions can be nonzero only when the main effect is nonzero.

To apply this idea to our scenario, consider again the specific genetic effect G and the set of environmental variables E. We are not going to consider all pairwise interactions but only interactions between G and each component of E. Thus the interaction term between G and E_i will look like $h_i\beta_i$ and not on the coefficient for the main effect of G. Specifics of the model are presented in Section 3.7.

We split the model in two parts, each of which is a LASSO type problem, and alternate between minimizing each of them to reach an optimal solution. Each of the problems is convex and the two together are bi-convex, thus any solution we reach is not guaranteed to be a global minimum. We estimate the main effects in one step and the interaction effects, or rather the h's in a second step. We propose an additional step to speed up convergence which is added after each time both sets of parameters have been optimized.

The model we present in this work differs from SHIM in that we add an additional penalty term to encourage a structure to the interactions estimated in the model (see Section 3.71 for details).

3.7 Algorithm for fitting a single group of interactions

We begin by applying the idea to a model with a single group of interaction effects estimated with a single h parameter. For simplicity, assuming the response is continuous, with a single group for all interactions, the linear regression model we propose is

$$\hat{\phi} = \underset{\beta_1 \ldots \beta_k, h}{\arg\min} ||Y - \beta_1 G - \sum_{j=2}^{K} \beta_j E_j - \sum_{j=2}^{K} h\beta_j G E_j||^2 + \lambda_\beta \sum_{j=2}^{K} |\beta_j| + \lambda_h |h| \qquad (3.2)$$

for $\phi = (\beta_1, \ldots, \beta_k, h)$. This is the strictest constraint we can impose on the form of the interaction effects and it is reasonable in very few models. For example, if there are a lot of potential interactions to be evaluated but only a few are non-zero, then this method estimates an h very close to 0 and will not perform well compared to regular LASSO. However, in the unlikely scenario that all interactions act in the same way, then this approach outperforms the LASSO. We again use this simple and very restrictive model as a building block to more flexible models.

To avoid the use of cross-validation or other computationally intensive methods for the estimation of the h parameter, we use an alternative model specification. We estimate it together with the other slope parameters β. Also note, that unlike the λ_h

tuning parameter, h does not control model complexity, so it is natural to estimate it in the same way as other model parameters rather than as a tuning parameter.

To solve problem 3.2, we iterate through the following steps until a convergence criterion is satisfied.

1. Hold $\beta_1 \dots \beta_k$ fixed and solve for h using the LASSO.

2. Hold h fixed and solve for $\beta_1 \dots \beta_k$ using the LASSO.

In (3.2), the h parameter is simultaneously estimated with the variable coefficients $\beta_1 \dots \beta_k$. Since the objective function is reduced at each step, convergence to a local minimum is guaranteed. As LASSO algorithms are fast and typically convergence is achieved in just a few iterations between the two steps, this is a very fast algorithm.

3.8 Algorithm for fitting multiple groups of interactions

Next, we would like to allow multiple groups of interactions in our model. We can express that by allowing each interaction effect to have its own h which relates it to the main effect, in that it is the ratio of the interaction effect to the main effect.

A less restrictive model for a single gene effect G can be formulated as

$$\hat{\phi} = \underset{\gamma,\beta_1 \dots \beta_k, h_1 \dots h_k}{\arg\min} ||Y - \gamma G - \sum_{k=1}^{K} \beta_k E_k - \sum_{k=1}^{K} h_k \beta_k G E_k||^2 + \lambda_h \sum_{k=1}^{K} |h_k| + \lambda_\beta \sum_{k=1}^{K} |\beta_k| \quad (3.3)$$

where $\phi = (\gamma, \beta_1, \dots, \beta_k, h_1, \dots, h_k)$ is the set of all parameters.

As for the previous example, we formulate this model for linear regression. Logistic regression is discussed in Section 3.7.3.

Note that if all h_k's are different this would be the traditional saturated model in which all interaction terms are included, but the h penalty will shrink some interactions away.

1. Initialize $\hat{\beta}^{(0)}$ and $\hat{h}^{(0)}$.

2. Iterate between the following two steps:

   (a) Fix the β's and γ and estimate the h_k's, by solving

   $$\hat{h} = \underset{h_1 \ldots h_k}{\arg\min} ||(Y - \gamma G - \sum_{k=1}^{K} \beta_k E_k) - \sum_{k=1}^{K} h_k(\beta_k G E_k)||^2 + \lambda_h \sum_{k=1}^{K} |h_k|.$$

   This is a standard LASSO problem with response $Y - \gamma G - \sum_{k=1}^{K} \beta_k E_k$ and predictors $\beta_k G E_k$.

   (b) Estimate the β's and the γ for fixed h_k's by solving

   $$\hat{\beta} = \underset{\gamma, \beta_1 \ldots \beta_k}{\arg\min} ||Y - \gamma G - \sum_{k=1}^{K} \beta_k (E_k + h_k G E_k)||^2 + \lambda_\beta \sum_{k=1}^{K} |\beta_k|.$$

   This is again a standard LASSO problem.

3. Stop when

   $$\text{diff} = \frac{M(\phi^{(j-1)}) - M(\phi^{(j)})}{M(\phi^{(j-1)})}$$

   is less then a set small number, where

   $$M(\phi) = ||Y - \gamma G - \sum_{k=1}^{K} \beta_k E_k - \sum_{k=1}^{K} h_k \beta_k G E_k||^2 + \lambda_h \sum_{k=1}^{K} |h_k| + \lambda_\beta \sum_{k=1}^{K} |\beta_k|$$

   is the fitted model for $\phi = (\gamma, \beta_1, \ldots, \beta_k, h_1, \ldots, h_k)$.

For a fixed set of parameters $(\lambda_\beta, \lambda_h)$ in Step 1 we start with the ordinary least squares (generalized linear model) estimates as the initial values for all the coefficients.

Note, that in the above equations, the intercept has been omitted from the penalty term. We standardize all predictors to mean zero and variance one so that the intercept can be estimated by $\sum_i Y_i/n$.

3.8.1 Directed LASSO with pairwise fused LASSO penalty

To apply the above model to our structured interactions scheme, we add in the pairwise fused LASSO penalty to the difference between h's [38]. This penalty term will

control the number of groups of interactions. When we use the same idea as Equation 3.3 and estimate the h's together with the main effects we are able to avoid having to pre-specify the number of groups or group membership in the model. Instead, the penalty we add controls the differences between h's and thus naturally encourages the formation of groups of interactions.

Following is the same model as Equation 3.3, but with additional penalization for the difference between h parameters. Let $\phi = (\gamma, \beta_1, \ldots, \beta_k, h_1, \ldots, h_k)$ and $\hat{\phi}$ be the minimizer of the following equation

$$\begin{aligned}\hat{\phi} = \underset{\gamma,\beta_1,\ldots,\beta_k,h_1,\ldots,h_k}{\arg\min} & ||Y - \gamma G - \sum_{k=1}^{K} \beta_k E_k - \sum_{k=1}^{K} h_k \beta_k G E_k||^2 \\ & + \lambda_h \left(\alpha \sum_{k=1}^{K} \sum_{j=1}^{K} |h_k - h_j| + \sum_{k=1}^{K} |h_k| \right) + \lambda_\beta \sum_{k=1}^{K} |\beta_k|. \end{aligned} \tag{3.4}$$

Here α is a pre-specified constant. To solve it we can again split the model in two parts and iterate between minimizing each until a solution is reached.

1. Initialize $\hat{\beta}^{(0)}$ and $\hat{h}^{(0)}$.

2. Iterate between the following two steps:

   (a) Fix the β's and γ and estimate the h_k's by solving

$$\begin{aligned}\hat{h} = \underset{h_1,\ldots,h_k}{\arg\min} & ||(Y - \gamma G - \sum_{k=1}^{K} \beta_k E_k) - \sum_{k=1}^{K} h_k (\beta_k G E_k)||^2 \\ & + \lambda_h \left(\alpha \sum_{k=1}^{K} \sum_{j=1}^{K} |h_k - h_j| + \sum_{k=1}^{K} |h_k|. \right) \end{aligned} \tag{3.5}$$

   (b) Estimate the β's and the γ for fixed h_k's by solving

$$\hat{\beta} = \underset{\gamma,\beta_1,\ldots,\beta_k}{\arg\min} ||Y - \gamma G - \sum_{k=1}^{K} \beta_k (E_k + h_k G E_k)||^2 + \lambda_\beta \sum_{k=1}^{K} |\beta_k|.$$

3. Stop when the relative difference between two consecutive steps is small.

In step 2(a) the response is $Y - \gamma G - \sum_{k=1}^{K} \beta_k E_k$ and the predictors are $\beta_k G E_k$. Because of the extra penalty on differences between parameters this is not a standard LASSO problem. Step 2(b) however is a LASSO problem with predictors $E_k + h_k G E_k$.

We minimize the objective function with respect to either the set of β's or h's and hence the objective function decreases at each step. The value of the objective function is then guaranteed to converge since it is bounded below. However, convergence to the global optimum is not guaranteed. The difficult part of solving 3.4 is the minimization in step 2(a) (Equation 3.5).

A simple way to speed up a two step alternating algorithm is to consider the complete parameter vector $\Theta = (\gamma, \beta_1, \ldots, \beta_p, h_1, \ldots, h_p)$. Let Θ_0 be the vector before the first step, and let Θ_1 be the vector after the second step. We add in a third step to the algorithm to perform a linesearch along the direction of $\Theta_1 - \Theta_0$, and we find the value τ so that $\Theta_0 \tau \Theta_1 - \Theta_0$ optimizes our objective. As our objective functions are all fast to evaluate, such a simple one-dimensional line-search is quick to carry out, and could reduce the number of times we need to carry out Steps 2(a) and 2(b) of our algorithm considerably in every situation.

3.8.2 Estimating the pairwise fused LASSO problem

Next we discuss two algorithms for implementing Step 2(a), i.e. minimizing Equation 3.5. Due to the presence of a pairwise fused LASSO penalty term, we cannot simply apply a LARS algorithm to the problem. However, a re-parameterization of the problem turns the optimization of step 2(a) (equation 3.5) into a LASSO-type problem. The idea mimics the approach taken for the elastic net [53]. Here we follow a technical report on the pairwise fused lasso [38].

Consider the following setup. Let $\theta_{jk} = |h_j - h_k|$ and let $\theta_{k0} = |h_k|$, with the additional constraint on the θ's that $\theta_{jk} = \theta_{j0} - \theta_{k0}, 1 \leq k \leq j \leq p$. Then the parameter vector becomes $\theta = (\theta_{10}, \ldots, \theta_{p0}, \theta_{21}, \ldots, \theta_{p(p-1)})^T$ of length $p + \binom{p}{2}$ and the expanded design matrix is $(\mathbf{X}|\mathbf{0}_{p \times \binom{p}{2}})$ where $\mathbf{0}_{p \times \binom{p}{2}}$ is a matrix of zeros. Let $\mathbf{Y}$ be

the continuous centered response. The constraint is included as a penalty term with a high penalty parameter γ:

$$\begin{aligned}\hat{\theta} &= \underset{\gamma,\beta_1\ldots\beta_k,h_1\ldots h_k}{\arg\min} ||\mathbf{Y} - (\mathbf{X}|\mathbf{0}_{\mathbf{p}\times(\mathbf{p/2})})\theta||^{\mathbf{2}} \\ &+ \lambda_h \left(\alpha \sum_{j=1}^{K-1} \sum_{k=j+1}^{K} |\theta_{jk}| + \sum_{k=1}^{K} |\theta_{k0}| \right) \\ &+ \gamma \sum_{j=1}^{p-1} \sum_{k=j+1}^{p} (\theta_{jk} - \theta_{j0} - \theta_{k0})^2\end{aligned}$$

Now the problem can be written as an ℓ_1-penalized regression. A similar strategy is employed by the authors of the elastic net ([53]). We re-write the minimization problem and incorporate the extra penalty terms in the predictor matrix $\mathbf{X}$.

Let $\mathbf{Y}_0 = (\mathbf{Y}, \mathbf{0}_{\binom{p}{2}})^T$ be the outcome augmented by a vector of zeros. The design matrix can be rewritten as

$$\tilde{\mathbf{D}} = \begin{pmatrix} \mathbf{X}|\mathbf{0}_{\mathbf{p}\times(\mathbf{p/2})} \\ \sqrt{\gamma}\mathbf{C} \end{pmatrix}$$

with the $p \times (\binom{p}{2} + p)$ matrix $\mathbf{C}$ accounting for the restriction $\theta_{jk} = \theta_{j0} - \theta_{k0}, 1 \leq k \leq j \leq p$, as follows:

$$\mathbf{C} = \begin{pmatrix} \boldsymbol{\delta}_{21} & \boldsymbol{\tau}_1 \\ \vdots & \vdots \\ \boldsymbol{\delta}_{32} & \boldsymbol{\tau}_p \\ \boldsymbol{\delta}_{42} & \boldsymbol{\tau}_{p+1} \\ \vdots & \vdots \\ \boldsymbol{\delta}_{p(p-1)} & \boldsymbol{\tau}_{\binom{p}{2}} \end{pmatrix}$$

with $\boldsymbol{\delta}_{jk}$ a p-dimentional row vector of zeros with -1 at the kth position and 1 at the jth position. And $\boldsymbol{\tau}_l$ is a $\binom{p}{2}$ row vector of zeros with -1 at the lth position.

Putting the above components together we have

$$\hat{\beta} = \underset{\gamma,\beta_1 \ldots \beta_k, h_1 \ldots h_k}{\arg\min} ||\mathbf{Y} - \tilde{\mathbf{D}}\theta||^2 + \lambda_h \left(\alpha \sum_{j=1}^{K-1} \sum_{k=j+1}^{K} |\theta_{jk}| + \sum_{k=1}^{K} |\theta_k| \right) \tag{3.6}$$

Equation 3.6 is a LASSO problem with an extended design matrix $\tilde{\mathbf{D}}$. Thus we have reconstructed the problem to fit into the LASSO set-up. We can use this to perform Step 2(a).

Note that this algorithm only works for a linear response. For a binary (logistic) response we discuss an alternate algorithm in Section 3.7.3.

3.8.3 Binary outcome

For a binary outcome, the binomial log likelihood can be written as

$$\ell(\beta) = \sum_{i=1}^{N} \left(y_i \beta^T x_i - \log(1 + e^{\beta^T x_i}) \right) \tag{3.7}$$

for a vector $\beta = (\beta_0, \beta_1, \ldots, \beta_k)^T$, for a model without an intercept. For the binomial log likelihood, our penalized model can be written as

$$\begin{aligned} \hat{\phi} = \underset{\gamma,\beta_k,h_k}{\arg\min} \frac{1}{n} \sum_{i}^{n} [& -Y_i(\beta_0 + \gamma G + \sum_{k=1}^{K} \beta_k E_k + \sum_{k=1}^{K} h_k \beta_k G E_k) \\ & + \log(1 + e^{\beta_0 + \gamma G + \sum_{k=1}^{K} \beta_k E_k + \sum_{k=1}^{K} h_k \beta_k G E_k})] \\ & + \lambda_h \left(\alpha \sum_{k=2}^{K} \sum_{j=1}^{k-1} |h_k - h_j| + \sum_{k=1}^{K} |h_k| \right) + \lambda_\beta \sum_{k=1}^{K} |\beta_k|. \end{aligned}$$

As in the linear model, we begin by standardizing the predictors, and the intercept term is not penalized. We also exclude the γ coefficient from the penalization.

To solve this we again split the problem into two estimation steps between which we iterate until a solution is reached.

1. Initialize $\hat{\beta}^{(0)}$ and $\hat{h}^{(0)}$.

2. Iterate between the following two steps:

(a) Fix the β's and γ and estimate the h_k's by solving

$$\begin{aligned}\hat{h} &= \arg\min_{\gamma,\beta_k,h_k} \frac{1}{n}\sum_{i}^{n}(-Y_i(\beta_0+\gamma G+\sum_{k=1}^{K}\beta_k E_k+\sum_{k=1}^{K}h_k\beta_k GE_k)\\ &+ \log(1+e^{\beta_0+\gamma G+\sum_{k=1}^{K}\beta_k E_k+\sum_{k=1}^{K}h_k\beta_k GE_k}))\\ &+ \lambda_h\left(\alpha\sum_{k=2}^{K}\sum_{j=1}^{k-1}|h_k-h_j|+\sum_{k=1}^{K}|h_k|\right).\end{aligned}$$

(b) Estimate the β's and the γ for fixed h_k's by solving

$$\begin{aligned}\hat{\beta} &= \arg\min_{\gamma,\beta_k,h_k} \frac{1}{n}\sum_{i}^{n}[-Y_i(\beta_0+\gamma G+\sum_{k=1}^{K}\beta_k E_k+\sum_{k=1}^{K}h_k\beta_k GE_k)\\ &+ \log(1+e^{\beta_0+\gamma G+\sum_{k=1}^{K}\beta_k E_k+\sum_{k=1}^{K}h_k\beta_k GE_k})]\\ &+ \lambda_\beta\sum_{k=1}^{K}|\beta_k|.\end{aligned}$$

3. Stop when the relative difference between two consecutive steps is small.

As before, we have a non-convex problem and though the objective function is guaranteed to converge no guarantee can be made about reaching the global minimum. In Step 2(a) of the algorithm, the model is fit with an offset value for the $\gamma G + \sum_{k=1}^{K}\beta_k E_k$ term for which no coefficient is estimated.

Again, we have a more complicated Step 2(a) with multiple penalty functions while Step 2(b) is a straightforward LASSO problem. However, the fitting method we outlined Section 3.7.2 for Step 2(a) does not work for generalized linear functions. We instead employ a strategy first developed by Ulbricht [47] and then adapted to the pairwise fused lasso penalty function by Petry et al [38].

3.8.4 Local quadratic approximation

The procedure is referred to as Local Quadratic Approximation (LQA). The algorithm fits a penalized generalized linear model by minimizing the log likelihood

$$\min_{b} -\ell(b)+P(\lambda,\beta). \tag{3.8}$$

The penalty term is in the form $P(\lambda, \beta) = \sum p_{\lambda,j}(|a_j^T \beta|)$, where a_j is a known vector of constants. There is a lot of flexibility in the types of penalty functions that can be expressed in this form. The sum of all the penalty functions represents the penalty region and the number of total penalties does not need to depend on the number of regressors. In other words, a single parameter can be involved in multiple penalty terms, as is the case for the fused LASSO penalty. The penalty functions in the sum and their corresponding λ's need not be the same.

The pairwise fused LASSO penalty, which has the form

$$P_{PFL}(\lambda_1, \lambda_2, \boldsymbol{\beta}) = \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \sum_{k=1}^{j-1} |\beta_j - \beta_k|,$$

fits the above equation form using

$$P_{PFL}(\lambda_1, \lambda_2, \boldsymbol{\beta}) = \sum_{j=1}^{\tilde{p}+p} p_{\lambda,j}(|a_j^T \beta|).$$

Here $\tilde{p} = \binom{p}{2}$, and

$$p_{\lambda,j} = \lambda_1 |\mathbf{a}_j^T \boldsymbol{\beta}|$$

for $j = 1, \ldots, p$ with $a_j = (0, \ldots, 0, 1, 0, \ldots, 0)^T$ with the one at the jth position, and

$$p_{\lambda,j} = \lambda_2 |\mathbf{a}_j^T \boldsymbol{\beta}|$$

for $j = p+1, \ldots, \tilde{p}+p$ with $a_j = (0, \ldots, 0, -1, 0, \ldots, 0, 1, 0, \ldots, 0)^T$ with one at the kth position and negative one at the lth.

For the penalty term part of (3.8) Ulbricht [47] developed a quadratic approximation based on which Newton type algorithms can be applied. The first part of equation 3.8 is the negative log-likelihood and the second part contains the ℓ_1-norm terms. Commonly convex optimization problems are approximated with a quadratic function and iterative Newton type algorithms are applied if the objective function is twice continuously differentiable. To get around this problem, Ulbricht proposes a quadratic approximation to the penalty term.

We follow Ulbricht [47] and Petry's [38] layout of the algorithm. For $u_j = |\mathbf{a}_j^T\boldsymbol{\beta}|$ and

$$p'_{\lambda,j} = \frac{dp_{\lambda,j}}{du_j},$$

Ulbricht shows that the gradient of the jth penalty function is

$$\nabla p_{\lambda,j} = \frac{\partial p_{\lambda,j}}{\partial \boldsymbol{\beta}} = p'_{\lambda,j}(u_j)\text{sgn}(\mathbf{a}_j^T\boldsymbol{\beta})\mathbf{a}_j. \tag{3.9}$$

And when $\beta_{(k)}$ is close to β we can approximate

$$\text{sgn}(\mathbf{a}_j^T\boldsymbol{\beta}_{(k)}) \approx \frac{\mathbf{a}_j^T\boldsymbol{\beta}}{|\mathbf{a}_j^T\boldsymbol{\beta}_{(k)}|}. \tag{3.10}$$

To avoid the restriction of $|\mathbf{a}_j^T\boldsymbol{\beta}_{(k)}| \neq 0$ when the term appears in the denominator it is replaced by the approximation

$$|\mathbf{a}_j^T\boldsymbol{\beta}_{(k)}| \approx \sqrt{(\mathbf{a}_j^T\boldsymbol{\beta}_{(k)})^2 + c}$$

with c being a small positive number.

Using simple arithmetic one can show that

$$\mathbf{a}_j^T\boldsymbol{\beta}\mathbf{a}_j^T(\boldsymbol{\beta} - \boldsymbol{\beta}_{(k)}) = \frac{1}{2}[\mathbf{a}_j^T(\boldsymbol{\beta} - \boldsymbol{\beta}_{(k)})]^2 + \frac{1}{2}(\boldsymbol{\beta}^T\mathbf{a}_j\mathbf{a}_j^T\boldsymbol{\beta} - \boldsymbol{\beta}_{(k)}^T\mathbf{a}_j\mathbf{a}_j^T\boldsymbol{\beta}_{(k)}). \tag{3.11}$$

When the $\boldsymbol{\beta}_{(k)}$ Newton-type approximation is close to β the first term is close to zero and so we can use the approximation

$$\mathbf{a}_j^T\boldsymbol{\beta}\mathbf{a}_j^T(\boldsymbol{\beta} - \boldsymbol{\beta}_{(k)}) \approx \frac{1}{2}(\boldsymbol{\beta}^T\mathbf{a}_j\mathbf{a}_j^T\boldsymbol{\beta} - \boldsymbol{\beta}_{(k)}^T\mathbf{a}_j\mathbf{a}_j^T\boldsymbol{\beta}_{(k)}). \tag{3.12}$$

The first order Taylor expansion of the jth penalty function using the two approximations above can be expressed as

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

which is a quadratic function in β. Summing over all J penalties yields

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

where

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

To accommodate the intercept in the model, the penalty matrix is extended with zero vectors to

$$\mathbf{A}_\lambda^\star = \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{A}_\lambda \end{bmatrix} \tag{3.13}$$

The algorithm starts with an initial step $\mathbf{b}_{(\mathbf{0})}$. Using the second order Taylor expansion of the negative log-likelihood at $\mathbf{b}_{(\mathbf{k})} = (\beta_{0,(k)}, \boldsymbol{\beta}_{(k)})^T$ and the approximation of the penalty term the next steps to update the Newton-type algorithm look like

$$\mathbf{b}_{(\mathbf{k+1})} = \mathbf{b}_{(\mathbf{k})} - (\mathbf{F}(\mathbf{b}_{(\mathbf{k})}) + \mathbf{A}^\star{}_\lambda)^{-1}(-\mathbf{s}(\mathbf{b}_{(\mathbf{k})}) + \mathbf{A}^\star{}_\lambda\mathbf{b}_{(\mathbf{k})}), \tag{3.14}$$

where $\mathbf{s}$ is the score function and $\mathbf{F}(\mathbf{b}_{(\mathbf{k})})$ is the Fisher information. The algorithm iterates until the relative distance taken between steps is less than some small $\epsilon > 0$.

Ulbricht published an R package LQA ([48]) which allows the user to define their own penalty function as long as it can be written in the above form. We define the pairwise fused penalty as described above and use the LQA approximation to estimate the h's in step 1.

In our experience, the LQA algorithm converges fairly well for problems that are not too large, although it is somewhat sensistive to good starting values. A disadvantage of the LQA algorithm is that the h parameters are not put exactly equal (or exactly equal to 0) by the LQA algorithm, so that some mild post-processing of the results is needed. We also note that the LQA package ([48]), as available from CRAN, did not include a working `offset` option, which we ended up adding to the package.

3.8.5 Tuning parameter selection

We have three tuning parameters $(\lambda_h, \lambda_\beta, \alpha)$ in problem 3.5. An exhaustive grid search is an onerous task. We instead propose to treat α as a user-chosen constant, as it is only a scaling parameter of how much more penalization is applied to the differences between parameters than to the parameters themselves. In our experience, using $\alpha = 2$ gives satisfactory results.

3.8.6 Alternate estimation: ADMM

An alternate approach to minimizing the problem in Equation 3.5 is to use the Alternating Directions Method of Multipliers algorithm (ADMM) ([5]). The methods were developed in the 1970s and is closely related to other methods, such as dual decomposition ([18]) and method of multipliers ([29]).

The algorithm solves problems of the form

$$\underset{x}{\text{minimize}}(f(x) + g(z))$$

subject to $Ax + Bz = c$. Where $x \in \boldsymbol{R}^n$ and $z \in \boldsymbol{R}^m$ with $A \in \boldsymbol{R}^{p \times n}$ and $B \in \boldsymbol{R}^{p \times m}$. We assume that both f and g are convex. The augmented Lagrangian is formed as

$$L_p(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + (\rho/2)||Ax + BZ - c||_2^2.$$

Then the algorithm consists of iterating between the following three steps; until convergence,

$$\begin{aligned} x^{k+1} &:= \arg\min_x L_\rho(x, z^k, y^k) \\ z^{k+1} &:= \arg\min_z L_\rho(x^{k+1}, z, y^k) \\ u^{k+1} &:= y^k + \rho(Ax^{k+1} + Bz^{k+1} - 1) \end{aligned}$$

with $\rho > 0$ chosen a priori.

Applying the ADMM algorithm to the LASSO problem takes the form

$$(1/2)||Ax - b||_2^2 + \lambda||x||_1$$

where $\lambda > 0$. This can be translated in the ADMM problem as

$$\begin{aligned} &\text{minimize} && f(x) + g(z) \\ &\text{subject to} && x - z = 0 \end{aligned}$$

where $f(x) = (1/2)||Ax - b||_2^2$ and $g(z) = \lambda||z||_1$. The steps of the algorithm are as follows:

$$\begin{aligned} x^{k+1} &:= (A^T A + \rho I)^{-1}(A^T b + \rho(z^k - u^k)) \\ z^{k+1} &:= S_{\lambda/\rho}(x^{k+1} + u^k) \\ u^{k+1} &:= u^k + x^{k+1} - z^{k+1}. \end{aligned}$$

We are interested in applying the ADMM algorithm to the pairwise fused LASSO problem. We want to solve the following

$$\underset{h}{\text{minimize}} \frac{1}{2}||Y - Xh||_2^2 + \lambda_h(\sum_k |h_k| + \alpha \sum_{1 \le k < j \le K} |h_k - h_j|). \tag{3.15}$$

Instead of doing a grid search, we can simplify the search of optimal parameters by re-parameterizing the penalty as $\lambda_1 \sum |h_k| + \lambda_2 \sum |h_k - h_j|$ so that the penalty for the differences which was expressed as $\alpha\lambda_h$ in (3.16) is not tied to the penalty for the absolute value of the h's. Friedman [24] has shown that instead of solving for a grid of λ_1 and λ_2 values you can set $\lambda_2 = 0$ and search over a grid of λ_1 values. Then soft-thresholding solve for all possible λ_2's. So we focus on the first penalty term only. Let F be a $\binom{p}{2} \times p$ matrix representation of the all the pairwise differences between the elements of h. Thus here

$$F = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \dots & 0 \\ -1 & 0 & 1 & 0 & \dots & 0 \\ \vdots & & & & & \vdots \\ 0 & \dots & 0 & -1 & 0 & 1 \\ 0 & \dots & 0 & 0 & -1 & 1 \end{bmatrix}.$$

Then we are minimizing the following

$$\underset{h}{\text{minimize}} \frac{1}{2}||Y - Xh||_2^2 + \lambda_1||Fh||,$$

which in ADMM form looks like

$$\underset{h,z}{\text{minimize}} \frac{1}{2}||Y - Xh||_2^2 + \lambda_1||z||,$$

subject to $Fh - z = 0$.

The three steps in the algorithm which we iterate between are then

$$\begin{aligned} h^{k+1} &:= (X^T X + \rho F^T F)^{-1}(X^T Y + \rho F^T(z^k - u^k)) \\ z^{k+1} &:= S_{\lambda/\rho}(Fh^{k+1} + u^k) \\ u^{k+1} &:= u^k + Fh^{k+1} - z^{k+1} \end{aligned}$$

where the soft thresholding operator $S_{\lambda/\rho}$ is interpreted elementwise.

3.9 Further simulations and data analysis

3.9.1 Note on comparisons

We choose to estimate overall prediction error for the directed LASSO, unlike the dedicated boosting algorithm, in which case performance was based on ability of the model to pick out the correct interaction terms. Even though with this model we are still focusing on the search of interactions we believe that since the estimation of the main effects and the interaction is tied together, the best estimate of performance will take into account the whole fitted model.

For the linear model we report residual sums of squares (RSS) based on a test set and the number of true positive (TP) and false positive (FP) interaction terms selected by the model. The model is tuned on a training set, and optimal parameters are chosen based on performance on a validation set. The test set is finally used to measure performance.

We compare the performance of directed LASSO to the SHIM model, the LASSO, and a full unpenalized model. The LASSO model was fit in two different ways. First we fit the LASSO without any restrictions allowing all main effects and interactions to be included. This often results in fitted models that do not satisfy the heredity constraints. We also fit the LASSO model, with the restriction that no penalty is applied to the main effects, forcing all of them in the model and automatically satisfying the heredity constraint.

Computations for the LQA algorithm are on the order of magnitude of $O(s[np^2 + up^3])$, where p is the number of predictors, n is the sample size, s is the number of steps the algorithm takes, and u is the number of updates of the Newton-type algorithm needs before converging. The computation for ADMM is on the order of $O(s[pn+p^3])$ and thus using ADMM would improve computation time. LQA is readily available for use in R which ADMM is not. However, at this stage we will use the LQA algorithm in our simulations.

3.9.2 WHI data example

We apply the linear directed LASSO to the WHI example for SNP rs10938397 for which we found evidence for interaction effects with the dedicated boosting algorithm (See Section 2.3). We applied the directed LASSO using the LQA algorithm to fit the model and used 10-fold cross-validation on a random training sample of 7000 observations. The remaining 10049 samples were left as a test sample.

Note that we used a different, smaller training set than in Section 2.3 because of limitations with the LQA algorithm not being able to allocate a data set larger than 10000 observations. So to be able to compare it to the boosting algorithm, we re-ran the boosting algorithm on this smaller sample.

Table 3.3 is a comparison of the coefficients as estimated by dedicated boosting and the directed LASSO with $\alpha = 2$ and the corresponding h parameters.

The directed LASSO rarely sets any coefficient exactly to zero. We believe this is

due to two factors. On one side, when there are a lot of larger coefficients and less null coefficients, when we apply the fused LASSO and penalize differences, some of the zero coefficients will instead be estimated with non-zero estimates. A second reason is that the LQA algorithm does not set estimates exactly equal to zero. Thus we find when comparing the interaction coefficients for SNP rs10938397 that the larger interaction effects are similar, while the interaction effects not selected by dedicated boosting are estimated with small coefficients by the directed LASSO.

3.9.3 Lymphoma data results

In approaching the Lymphoma data set, we select a random sample of 10 SNPs to consider in the model. We form the interaction between the 10 SNPs and the treatment variable. To build the directed LASSO model we perform 10-fold cross-validation to select optimal penalty factors. We record the interactions and main effects in the model. We repeat this 100 times and count how many times a certain interaction effect makes it in the model.

The idea for our approach to the Lymphoma data comes from the well know method of random forests [7], an ensemble learning method, where a subset of the predictors are used to estimate the regression coefficients over bootstrap samples of the observations. These coefficients are aggregated over many runs where a different sample of predictors is used.

Note that we take this approach because the LQA algorithm did not converge when presented with all 32 interaction terms. We believe this is a limitation of the algorithm and plan to execute step 1 of our model with an alternate approach in the future. The alternative we are considering is ADMM and was described in Section 3.8.

Unfortunately, there is no signal for interactions in this data set. Not a single interaction term is selected through the 100 runs of the model. In more detail, Table 3.4 summarizes the results. For two different thresholds, we count the average number

Table 3.3: Interaction coefficients selected by dedicated boosting and directed LASSO

	Boost	Dir. LASSO ($\alpha = 2$)	h
Age	−0.01	−0.20	0.8
Amount of exercise	0.00	−0.02	0.3
Exercise at 18	−0.29	−0.52	−0.4
Exercise at 35	0.00	0.20	2.8
Exercise at 50	0.00	0.03	0.01
% Calories from carbo.	0.00	−0.08	−5.1
% Calories from protein	0.00	0.00	0.1
% Calories from fat	0.00	−0.10	−0.8
Education level	0.05	0.24	−0.7
Ever smoking	0.00	0.09	0.6
Current smoking	0.00	−0.47	0.2
Alcohol	0.00	−0.05	0.1
Hispanic	0.00	−0.17	0.4
African American	0.00	0.24	0.1
Asian/Pacific Islander	−2.10	−1.03	0.3
American Indian	1.08	1.01	−0.7
Region middle	−0.36	−0.49	−4.8
Region south	0.00	0.11	−0.4

of times a particular interaction (between the treatment and a SNP) is estimated with a coefficient larger than the threshold. None of the interaction coefficients reach the threshold of 0.001. Very few reach the threshold of 0.00001. We present results only for the SNP that reach that threshold. The omitted interactions were never selected with a coefficient larger than 0.00001.

Table 3.4: Results for the Lymphoma Data: Average number of times a coefficient for a specific SNP is larger than the given threshold.

SNP	Coef > 0.001	Coef > 0.00001
7	0.00	0.08
11	0.00	0.69
16	0.00	0.11
19	0.00	0.42
21	0.00	0.94
23	0.00	0.07
27	0.00	0.30
28	0.00	0.50
31	0.00	0.21

3.9.4 Simulations: Linear regression

In the linear regression simulation scenarios, we use the LQA algorithm in step 1 of our algorithm. We found that LQA is faster than the matrix augmentations proposed in Section 3.71 for the simulation sizes below, while giving very similar results.

$$y = \rho g + \beta^T \mathbf{x} + \gamma^T \mathbf{x} g + \epsilon \tag{3.16}$$

We simulate a 100 observations from model 3.16 where $\mathbf{x}$ are standard normal uncorrelated continuous predictors, $g \sim \text{Bin}(0.6)$ and $\epsilon \sim N(0, 1)$ with $\rho = 1$. The model coefficients as presented in Table 3.5.

Table 3.6 presents the MSE from the five simulated scenarios. When there are more predictors in the model (Model B vs. Model C) the directed LASSO performs the best. It also performs well in the null interactions scenario (Model D). When the model does not follow the heredity constraint the directed LASSO outperforms

Table 3.5: Simulation set up: Model coefficients for linear regression.

	β_1 $\vdots$ β_5	β_6 $\vdots$ β_{10}	β_{11} $\vdots$ β_{15}	γ_1 $\vdots$ γ_5	γ_6 $\vdots$ γ_{10}	γ_{11} $\vdots$ γ_{15}
Model A	2	2	0	1	0	0
Model A2	2	2	0	0.25	0	0
Model B	2	1	1	1	0	0
Model B2	2	1	1	0.25	0	0
Model C	2	1	-	1	0	-
Model C2	2	1	-	0.25	0	-
Model D	2	0	0	0	0	0
Model E	2	0	0	0	0	1

SHIM, but the LASSO models do the best. We believe this is the case because both of the models assume the true underlying model does follow the heredity constraint and do not allow for deviations from that.

To estimate the true positive (TP) coefficients each model selects, we average the number of true non-zero interaction coefficients that are estimated to be larger than 0.001 and average this over all simulations. Similarly, false positives (FP) are the average of the zero interaction coefficients which are estimated to be larger than 0.001 by the model, averaged over all simulation runs for each simulated scenario.

Table 3.8 presents the average true positive and false positive coefficients selected by all the methods. We note that SHIM does exceptionally well at not selecting any interactions when the true underlying model is null, however that does not significantly improve its performance and directed boosting has smaller MSE.

Interestingly, when the interaction coefficients are much smaller than the main

effects, as is the case with Model A2, directed LASSO outperforms the other methods in terms of MSE, but also has perfect TP selection and better FP than the original LASSO. The restricted LASSO has lower FP but also lower TP and worse overall MSE performance. The SHIM model is not selecting many interactions. It has both very low TP and FP rates, and worse MSE performance than directed LASSO.

Table 3.6: Simulation Results: MSE (SE). Uncorrelated predictors. "Full" is the full regression model that includes all predictors and interactions. "LASSO" is the LASSO model without any constraints. "Res. LASSO" is the LASSO model where the main effects are not penalized. "Dir. LASSO" is the directed LASSO. The **boldfaced** results are the best for a particular model.

	Dir. LASSO	SHIM	LASSO	Res. LASSO	Full
Model A	0.36	0.50	**0.35**	0.48	0.55
	(0.012)	(0.017)	(0.012)	(0.016)	(0.019)
Model A2	**0.26**	0.30	0.32	0.38	0.51
	(0.010)	(0.010)	(0.011)	(0.011)	(0.016)
Model B	**0.37**	0.50	0.45	0.52	0.53
	(0.013)	(0.019)	(0.018)	(0.022)	(0.022)
Model B2	**0.25**	0.30	0.40	0.48	0.51
	(0.011)	(0.011)	(0.016)	(0.018)	(0.019)
Model C	**1.06**	1.68	1.21	1.12	2.89
	(0.226)	(0.359)	(0.066)	(0.057)	(0.549)
Model C2	0.75	0.67	1.03	**0.58**	4.26
	(0.116)	(0.064)	(0.068)	(0.028)	(1.494)
Model D	**0.18**	0.22	0.20	0.32	0.53
	(0.008)	(0.009)	(0.009)	(0.013)	(0.019)
Model E	0.44	0.63	**0.29**	0.32	0.51
	(0.018)	(0.023)	(0.013)	(0.014)	(0.018)

Table 3.7: Simulation Results: MSE (SE). Correlated predictors. "Full" is the full regression model that includes all predictors and interactions. "LASSO" is the LASSO model without any constraints. "Res. LASSO" is the LASSO model where the main effects are not penalized. "Dir. LASSO" is the directed LASSO. The **boldfaced** results are the best for a particular model.

	Dir. LASSO	SHIM	LASSO	Res. LASSO	Full
Model A	0.33	0.36	**0.30**	0.46	0.52
	(0.013)	(0.014)	(0.013)	(0.015)	(0.019)
Model B	**0.36**	0.42	0.43	0.54	0.53
	(0.012)	(0.016)	(0.016)	(0.02)	(0.02)
Model E	0.38	0.56	**0.26**	0.32	0.50
	(0.013)	(0.018)	(0.011)	(0.013)	(0.017)

Next we look at performance when the non-zero predictors are correlated. We simulated random normal predictors with mean zero and variance 1, with correlation 0.4 between predictors following models A, B and E.

Table 3.7 presents the MSE from the simulation study over 100 runs of each model. As we would expect, the performance of both SHIM and directed LASSO suffers when the true underlying model does not satisfy the heredity constraint (Model E). In this situation, while directed LASSO is outperformed by the regular LASSO, it still does better than SHIM. We believe that this is the case because, even though the model is not satisfied, there are still groups of interactions which directed LASSO would be able to pick out and group together more easily.

When there are fewer main effects and fewer interactions (Model A) we note that the performance of the directed LASSO, SHIM and LASSO is very similar. In the non-correlated scenario (Table 3.6) the directed LASSO does the best, while in the correlated scenario the LASSO does slightly better. When there are fewer interactions than main effects, the directed LASSO model performs best in both the correlated

Table 3.8: Simulation Results: Average True Positive and False Positive coefficients for uncorrelated models. "Dir. LASSO" is the directed LASSO.

		Dir. LASSO	SHIM	LASSO	Res. LASSO	Full
Model A	TP	1.00	1.00	1.00	1.00	1.00
	FP	0.91	0.52	0.99	0.59	0.99
Model A2	TP	1.00	0.33	1.00	0.89	0.55
	FP	0.85	0.14	0.99	0.60	0.99
Model B	TP	1.00	1.00	1.00	1.00	1.00
	FP	0.97	0.69	1.00	0.71	0.99
Model C	TP	1.00	0.90	1.00	0.98	0.87
	FP	0.99	0.40	1.00	0.73	0.57
Model D	FP	0.61	0.05	0.99	0.41	0.75
Model E	TP	1.00	1.00	1.00	1.00	1.00
	PT	0.90	0.71	0.99	0.54	0.66

Table 3.9: Simulation Results: Average True Positive and False Positive coefficients for correlated models. "Dir. LASSO" is the directed LASSO.

		Dir. LASSO	SHIM	LASSO	Res. LASSO	Full
Model A	TP	1.00	1.00	1.00	1.00	1.00
	FP	0.89	0.38	1.00	0.45	0.99
Model B	TP	1.00	1.00	1.00	1.00	1.00
	FP	0.97	0.35	1.00	0.69	0.99
Model E	TP	1.00	0.99	1.00	1.00	1.00
	FP	0.90	0.65	1.00	0.44	0.60

and uncorrelated cases (Model B).

3.9.5 Simulations: Binary outcome

We simulate 1000 observations from 20 independent standard normal predictors. The genetic effect G is again simulates as Bin(0.6), independent of the continuous predictors. Table 3.10 summarizes the logistic regression coefficients used in the different simulated scenarios. In all cases, the intercept is 0.

Results are based on 100 replications of the simulation scenarios. We report the deviance as a summary measure in Table 3.11.

When the true underlying model satisfies the heredity constraints (Model 1) the directed LASSO performs the best out of the models compared. In the second scenario (Model 2) where we simulate a null interaction situation our model is still able to outperform the other methods.

Table 3.10: Simulation setup: Model coefficients

	G	Main Effects		Int. Effects			
	γ	$(\beta_2 \dots \beta_{12})$	$(\beta_{13} \dots \beta_{22})$	$(\beta_{23} \dots \beta_{27})$	$(\beta_{28} \dots \beta_{32})$	$(\beta_{33} \dots \beta_{37})$	$(\beta_{38} \dots \beta_{42})$
1)	1	2	0	1	1	0	0
2)	1	2	0	0	0	0	0

Table 3.11: Simulation Results: Deviance. Uncorrelated predictors. "Full" is the LS model including all predictors. "LASSO" is the LASSO model without any constraints. "Res. LASSO" is the LASSO model where the main effects are not penalized. "Dir. LASSO" is the directed LASSO.

	Dir. LASSO	SHIM	LASSO	Res. LASSO	Full
Model 1	**343.16**	358.09	356.78	354.29	391.22
	(2.97)	(3.29)	(2.82)	(3.40)	(4.64)
Model 2	**409.54**	423.69	420.57	415.04	448.48
	(2.96)	(3.17)	(2.81)	(3.32)	(4.00)

Chapter 4

DISCUSSION

In this dissertation, we propose two different approaches for detection of interaction with multiple components. They are designed to pick out interactions with small effects when other off-the-shelf methods may not identify any interaction effects.

In many genetic epidemiological studies, it is not just of interest to identify SNPs that are associated with particular phenotypes, but it is also of interest to identify environmental and demographical factors that modify these genetic effects. The search for such effect modifiers has often had limited success, both because the effect modifications are small, and because various of the variables are measured with error.

The directed LASSO is designed for instances where we want to link the main effects and the interactions effects. We can impose constraints on how the interaction effects are associated with the main effects and control that relation via one or more penalty parameters. The big advantages of this model are when there are groups of interactions that modify the main effects in a similar fashion. This is a plausible scenario when, for example, we have a treatment effect and we are investigating the interactions between the treatment and a group of SNPs. SNPs that are located on the same gene or genes that are associated with a similar process are likely to modify the treatment effect in a similar way. In Chapter 3 we explore the benefits of imposing such a structure on the interactions. We found that the biggest gains for our model are found when there is a group of factors with medium to large interaction effects. When there are no interactions our approach typically does not estimate any.

Dedicated boosting is a variation of ℓ_2 boosting which focuses on the search for effect modifiers. We were interested in developing a method that is able to pick

out ensembles of weaker effects of covariates that interact with another risk factor, such as a SNP. Well-known methods such as AIC and BIC model selection with stepwise model building can be modified to be used for finding interactions. However, when using these methods, the effect of the interactions needs to be fairly strong for them to be included in the final model. Penalized regression methods, such as the lasso and boosting, are well suited for finding solutions which consist of combinations of weaker effects. Our interest was in adapting such a method for low signal in a search for interactions. In a simulation study our method outperforms the LASSO, globalboosttest, AIC, and BIC model selection procedures as having the lowest test error. In the WHI-PAGE data example the dedicated boosting method was able to pick out two SNPs for which effect modification appears present. The performance was evaluated on an independent test set and the results are promising. For most SNPs no effect modification was detected by any of the methods. In these cases the performance of dedicated boosting is not markedly different than the rest of the methods. However, when some effect modification is present dedicated boosting gives lower error rates on the independent test set, as was the case with SNP rs10938397.

We also compare dedicated boosting to the directed LASSO on the WHI data looking for interactions with SNP rs10938397. The directed LASSO ends up with many more small non-zero interaction effects while the dedicated boosting sets those directly to zero.

We plan to continue working with the directed LASSO model and extend it to survival responses. We also plan on utilizing the ADMM algorithm to speed up fitting of the model, as described in Section 3.8.

BIBLIOGRAPHY

[1] H. Akaike. Model selection adn akaike's information criterion (aic): The general theory and its analytical extensions. *IEEE Trans. Auto. Control*, 19:716–723, 1974.

[2] J. Bien, J. Taylor, and R. Tibshirani. A lasso for hierarchical interactions. *Annals of Statistics*, in press, 2013.

[3] H. Bondell and B. Reich. Simultaneous regression shrinkge, variable selection, and supervised clustering of predictors with oscar. *Biometrics*, 64:115–123, 2008.

[4] A.L. Boulesteix and T. Hothorn. Testing the additional predictive value of high-dimensional molecular data. *BMC Bioinformatics*, 11, 2010.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3:1–122, 2011.

[6] H. Bozdogan. Model selection adn akaike's information criterion (aic): The general theory and its analytical extensions. *Psychometrika*, 52:345–370, 1987.

[7] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[8] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.

[9] L. Breiman, J. Friedman, R.A. Olshen, and C. Stone. *Classification and regression trees*. Wadsworth, 1984.

[10] P. Bühlmann. Boosting for high-dimensional linear models. *Annals of Statistics*, 34(2):559–583, 2006.

[11] P. Bühlmann and T. Hothorn. Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science : a Review Journal of the Institute of Mathematical Statistics*, 22(4):477–505, 2007.

[12] P. Bühlmann and B. Yu. Boosting with the l_2 loss: Regression and classification. *Journal of the American Statistical Association*, 98(462):324–339, 2003.

[13] N. Chatterjee, Z. Kalaylioglu, R. Moslehi, U. Peters, and S. Wacholder. Powerful multilocus tests of genetic association in the presence of gene-gene and gene-environment interactions. *American Journal of Human Genetics*, 79(6):1002–1016, 2006.

[14] N. Choi, L. William, and J. Zhu. Variable selection with the strong heredity constraint and its oracle property. *Journal of the American Statistical Association*, 105(489):354–364, 2010.

[15] R.D. Cook and S. Weisberg. *Residuals and influence in regression.* London: Chapman and Hall, 1982.

[16] B. Efron. Bootstrap methods: Another look at the jackknife. *Annals of statistics*, 7(1):1–26, 1979.

[17] B. Efron. Better bootstrap confidence intervals. *Journal of American Statistical Association*, 82(397):171–185, 1987.

[18] H. Everett. Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research*, 11(3):399–417, 1963.

[19] J. Fan and R. Li. Variable selection via non concave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

[20] M. Fesinmeyer, K.E. North, M.D. Ritchie, U. Lim, N. Franceschini, et al. Genetic risk factors for body mass index and obesity in an ethnically diverse population: results from the Population Architecture using Genomics and Epidemiology (PAGE) Study. *Obesity*, 2012.

[21] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):199–139, 1997.

[22] J. Friedman. Multivariate adaptive regression splines. *Annuals of Statistics*, 19(1):1–67, 1991.

[23] J. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001.

[24] J. Friedman, T. Hastie, H. Hoefling, and R. Tibshirni. Pathwise coordinate optimization. *Annals of applied statistics*, 1:302–332, 2007.

[25] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2 2010.

[26] J. Friedman and W. Stuetzle. Projection pursuit regression. *Journal of the American Statistical Association*, 76(376):817–823, 1981.

[27] W. Fu and K. Knight. Asymptotics for lasso-type estimators. *Annals of statistics*, 28(5):1356–1378, 2000.

[28] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: Data mining, inference, and prediction*. Springer series in statistics, New York: Springer, 2001.

[29] M. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4:302–320, 1969.

[30] A. Hoerl and R. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

[31] Y. Kim, R. Wojciechowski, S. Heejong, R. A. Mathias, W. Li, A. P. Klein, R. K. Lenroot, J. E. Bailey-Wilson, and J. Malley. Evaluation of random forests performance for genome-wide association studies in the presence of interaction effects. *BMC Proceedings*, 3, 2009.

[32] C. Kooperberg and M. LeBlanc. Increasing the power of identifying gene x gene interactions in genome-wide association studies. *Genetic Epidemiology*, 32:255–263, 2008.

[33] E. Leamer. *Specification searches*. Wiley, New York, 1978.

[34] K. Lunetta, L. Hayward, J. Segal, and E. Van. Screening large-scale association study data: Exploiting interactions using random forests. *BMC Genetics*, 5, 2004.

[35] K.K. Nicodemus, W. Wang, and Y.Y. Shugart. Stability of variable importance scores and rankings using statistical learning tools on single-nucleotide polymorphisms and risk factors involved in gene x gene and gene x environment interactions. *BMC Proceedings*, 1, 2007.

[36] M. Park and T. Hastie. Penalized logistic regression for detecting gene interactions. *Biostatistics*, 9(1):30–50, 2008.

[37] H. Pashova, M. Leblanc, and C. Kooperberg. Boosting for detection of gene-environment interactions. *Statistics in Medicne*, 3(2):255–266, 2013.

[38] S. Petry, C. Flexeder, and G. Tutz. Pairwise fused lasso. *Technical report, University of Munich*, 2011.

[39] L.M. Rimsza, M.L. LeBlanc, J.M. Unger, T.P. Miller, T.M. Grogan, and et al. Gene expression predicts overall survival in paraffin-embedded tissues of diffuse large b-cell lymphoma treated with r-chop. *Blood*, 112:3425–3433, 2008.

[40] I. Ruczinski, C. Kooperberg, and M. LeBlanc. Logic regression. *Journal of Computational and Graphical Statistics*, 12(3):475–511, 2003.

[41] G. Schwartz. Estimating the dimension of a model. *Annals of Statistics*, 19:716–723, 1978.

[42] The Women's Health Initiative Study Group. Design of the women's health initiative clinical trial and observational study. *Controlled Clinical Trials*, 19:61–109, 1998.

[43] D. Thomas. Methods for investigating gene-environment interactions in candidate pathway and genome-wide association studies. *Annual Review of Public Health*, 31:21–36, 2010.

[44] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58:267–288, 1996.

[45] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (statistical Methodology)*, 67(1):91–108, 2005.

[46] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society Series B*, pages 91–108, 2005.

[47] J. Ulbricht. Variable selection in generalized linear models. *PhD Thesis*, 2010.

[48] Jan Ulbricht. *lqa: Penalized Likelihood Inference for GLMs*, 2010. R package version 1.0-3.

[49] Women's Health Initiative Steering Committee. Effects of conjugated equine estrogen in postmenopausal women with hysterectomy. *Journal of the American Medical Association*, 291:1701–1712, 2004.

[50] Writing Group for the Women's Health Initiative. Risk and benefit of estrogen plus progestin in healthy postmenopausal women: Principal results from the women's health initiative randomized controlled trial. *Journal of the American Medical Association*, 288:321–333, 2002.

[51] M. Yuan, R. Joseph, and H. Zou. Structured variable selection and estimation. *Annals of Applied Statistics*, 3(4):1738–1757, 2009.

[52] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

[53] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B*, 67(2):301–320, 2005.

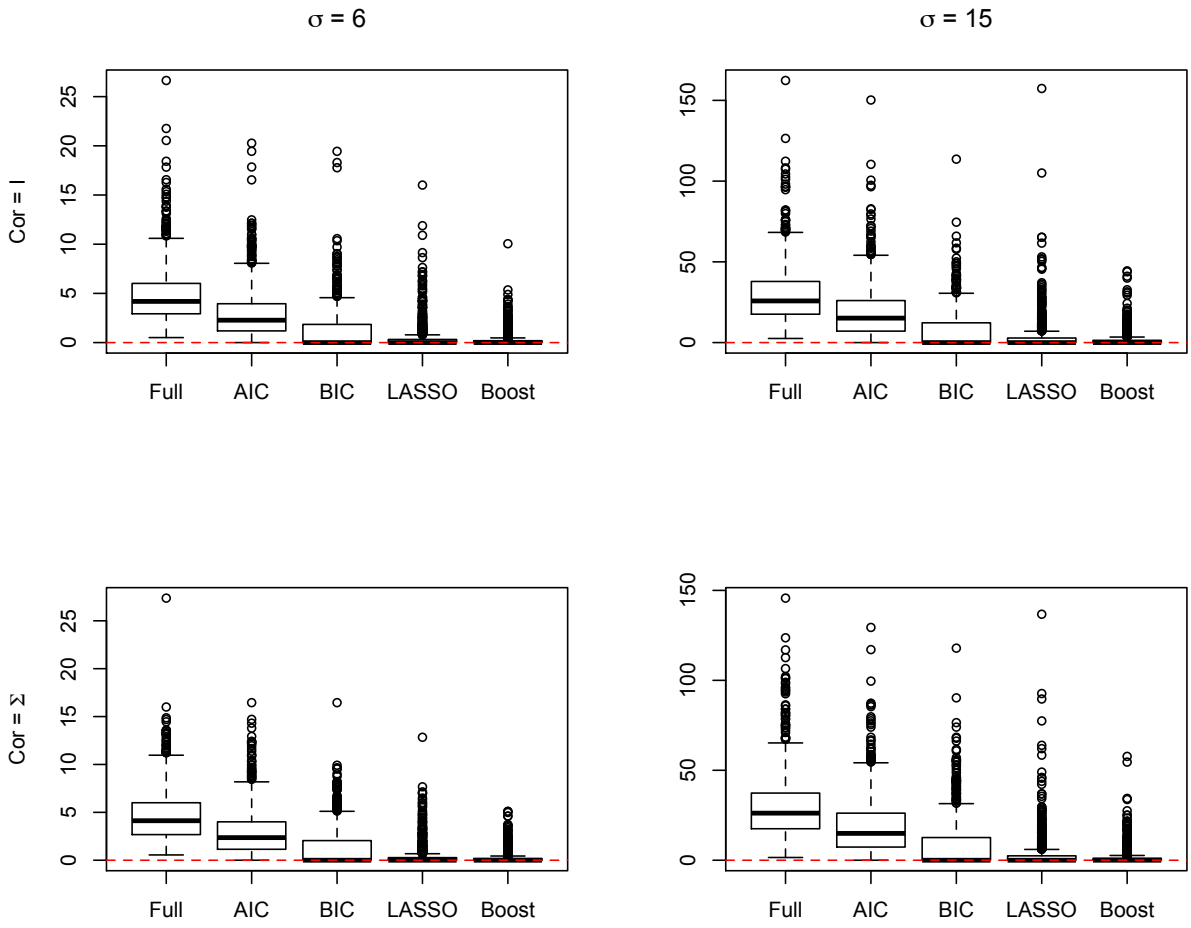

Figure 4.1: Simulation study results based on 1000 replications no interactions models (Null Models, N = 100). "Full" refers to fitting all interaction terms using a linear model; "Boosting" is the dedicated boosting algorithm.

Tables and Figures

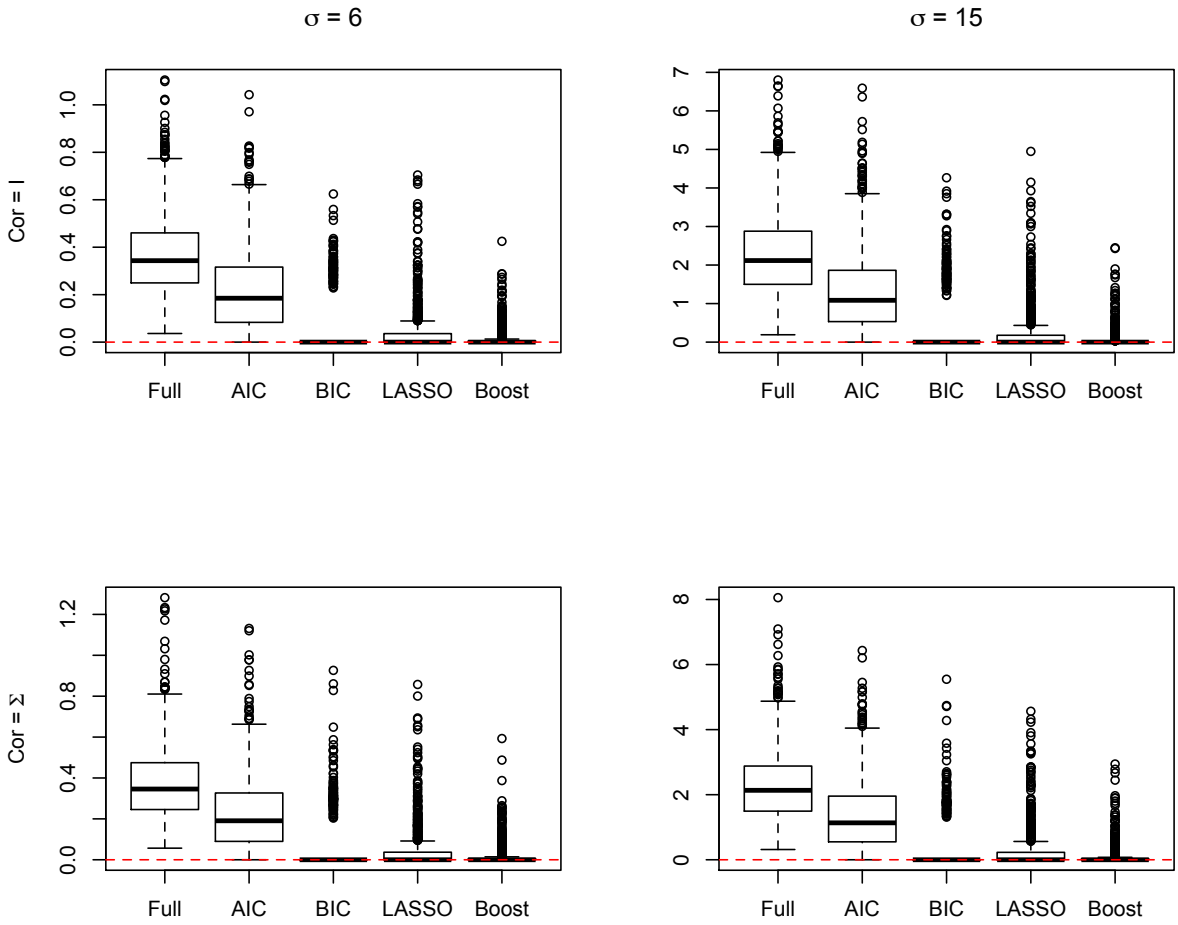

Figure 4.2: Simulation study results based on 1000 replications with no interactions (Null Models, N = 1000). "Full" refers to fitting all interaction terms using a linear model; "Boosting" is the dedicated boosting algorithm.

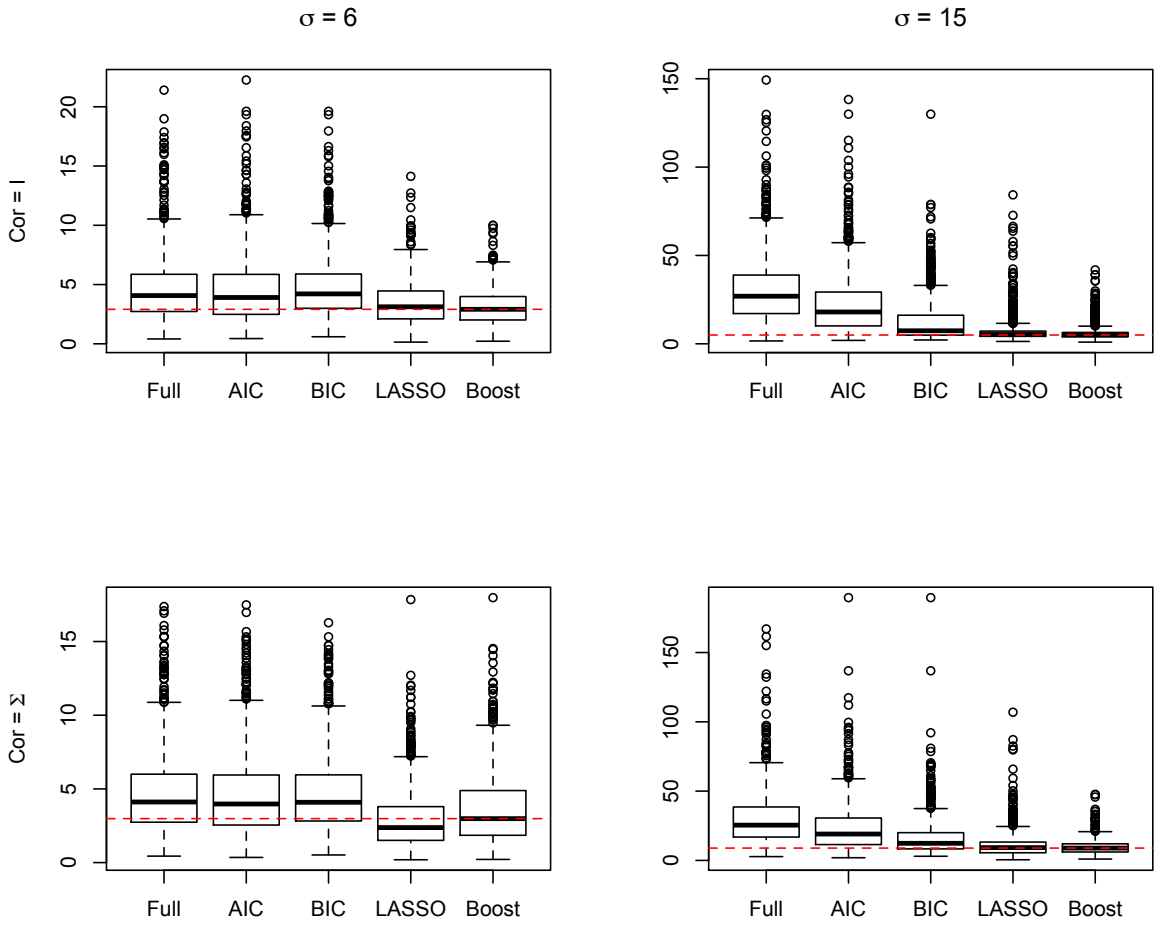

Figure 4.3: Simulation study results based on 1000 replications (Model A, N = 100). "Full" refers to fitting all interaction terms using a linear model; "Boosting" is the dedicated boosting algorithm.

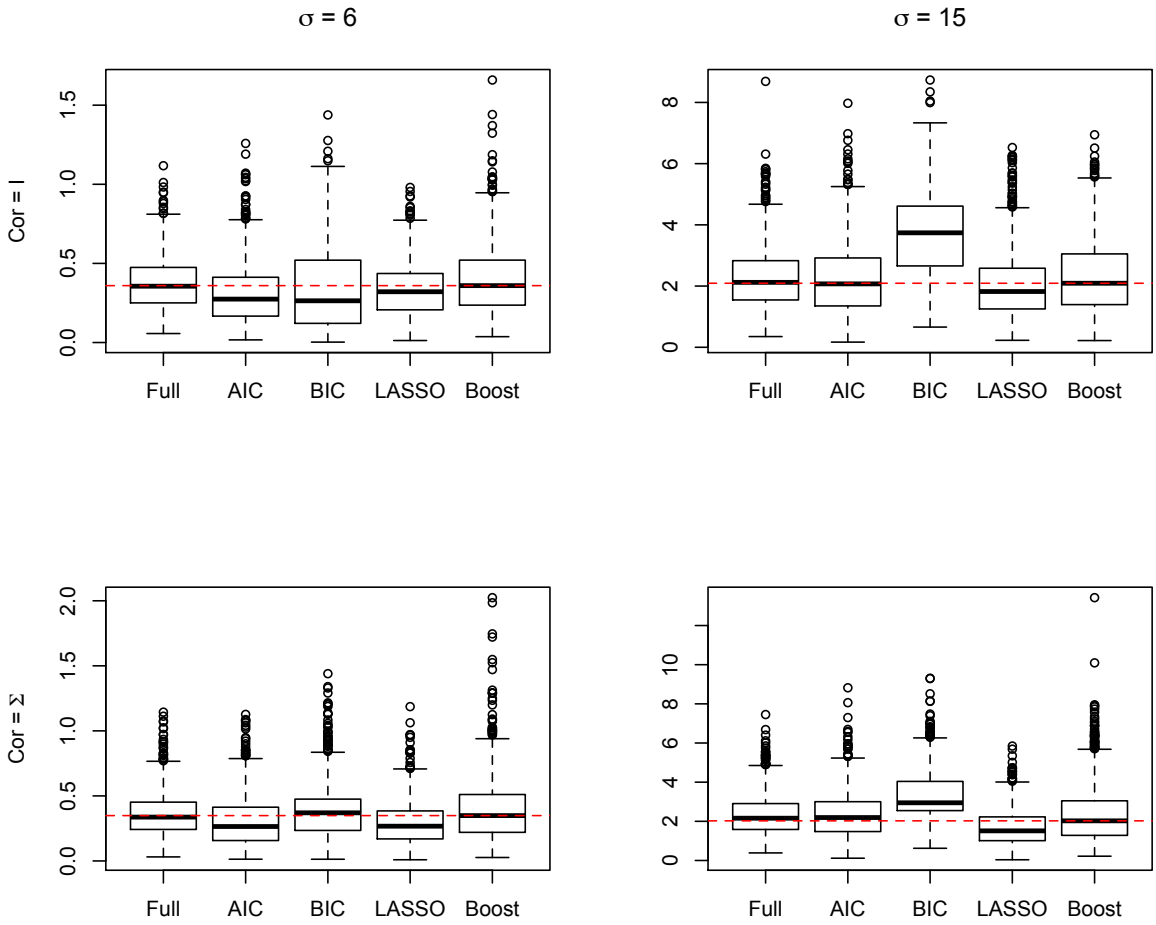

Figure 4.4: Simulation study results based on 1000 replications (Model A, N = 1000). "Full" refers to fitting all interaction terms using a linear model; "Boosting" is the dedicated boosting algorithm.

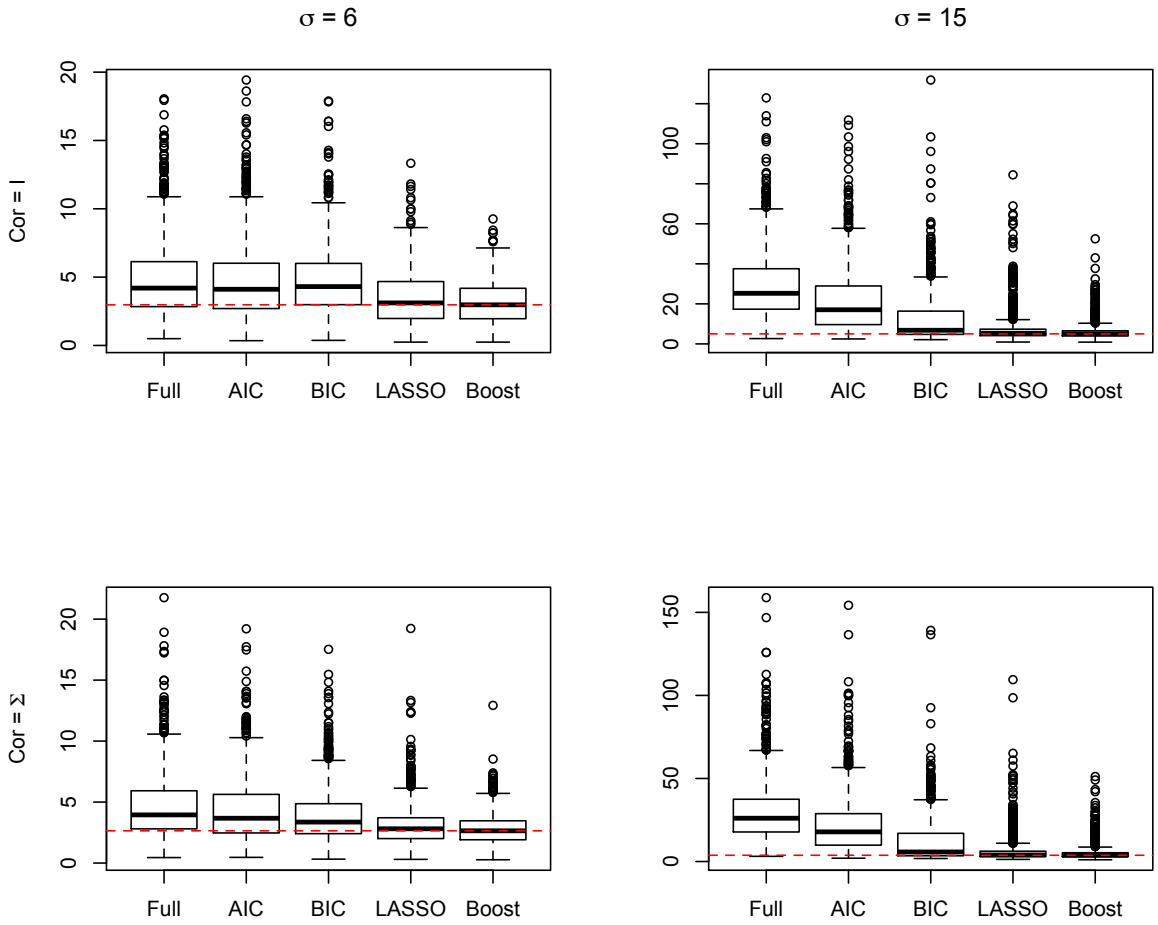

Figure 4.5: Simulation study results based on 1000 replications (Model B, N = 100). "Full" refers to fitting all interaction terms using a linear model; "Boosting" is the dedicated boosting algorithm.

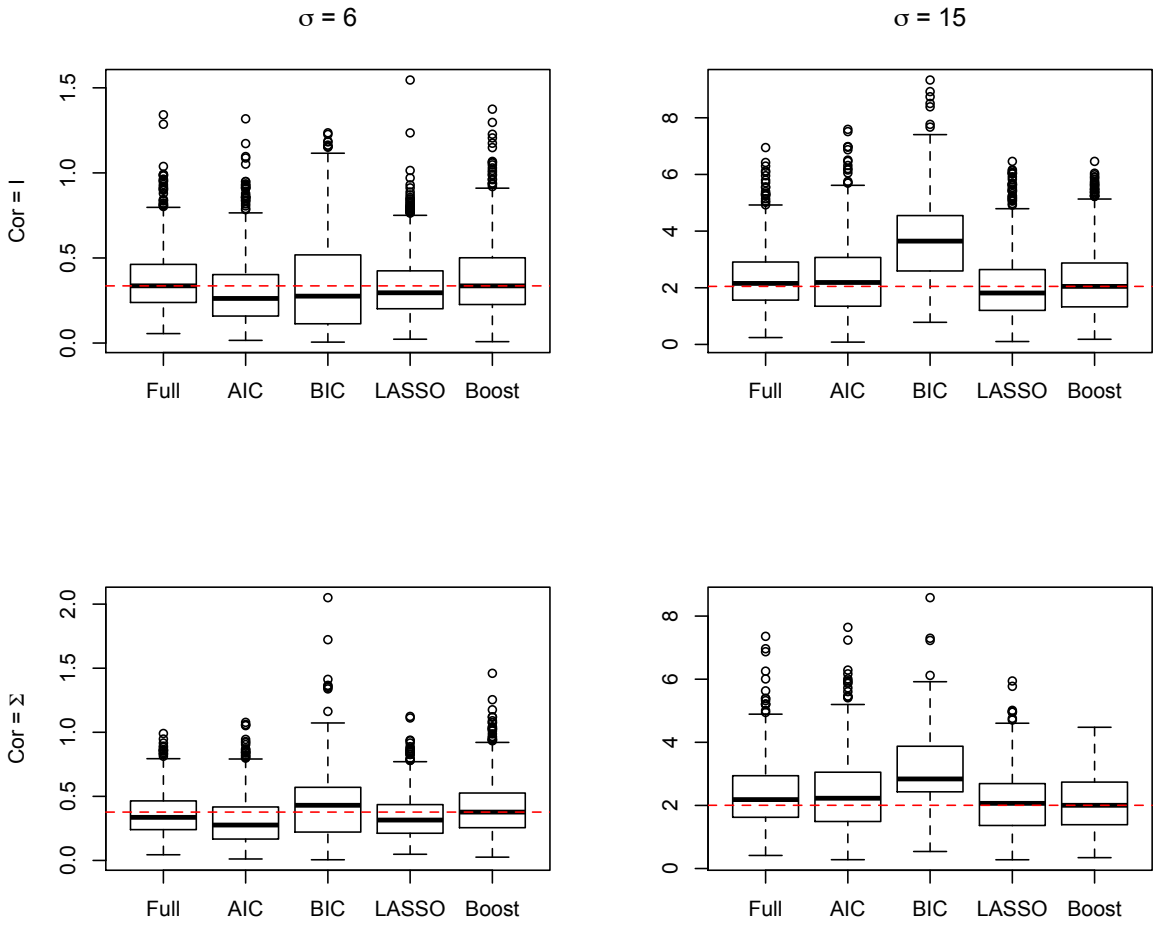

Figure 4.6: Simulation study results based on 1000 replications (Model B, N = 1000). "Full" refers to fitting all interaction terms using a linear model; "Boosting" is the dedicated boosting algorithm.

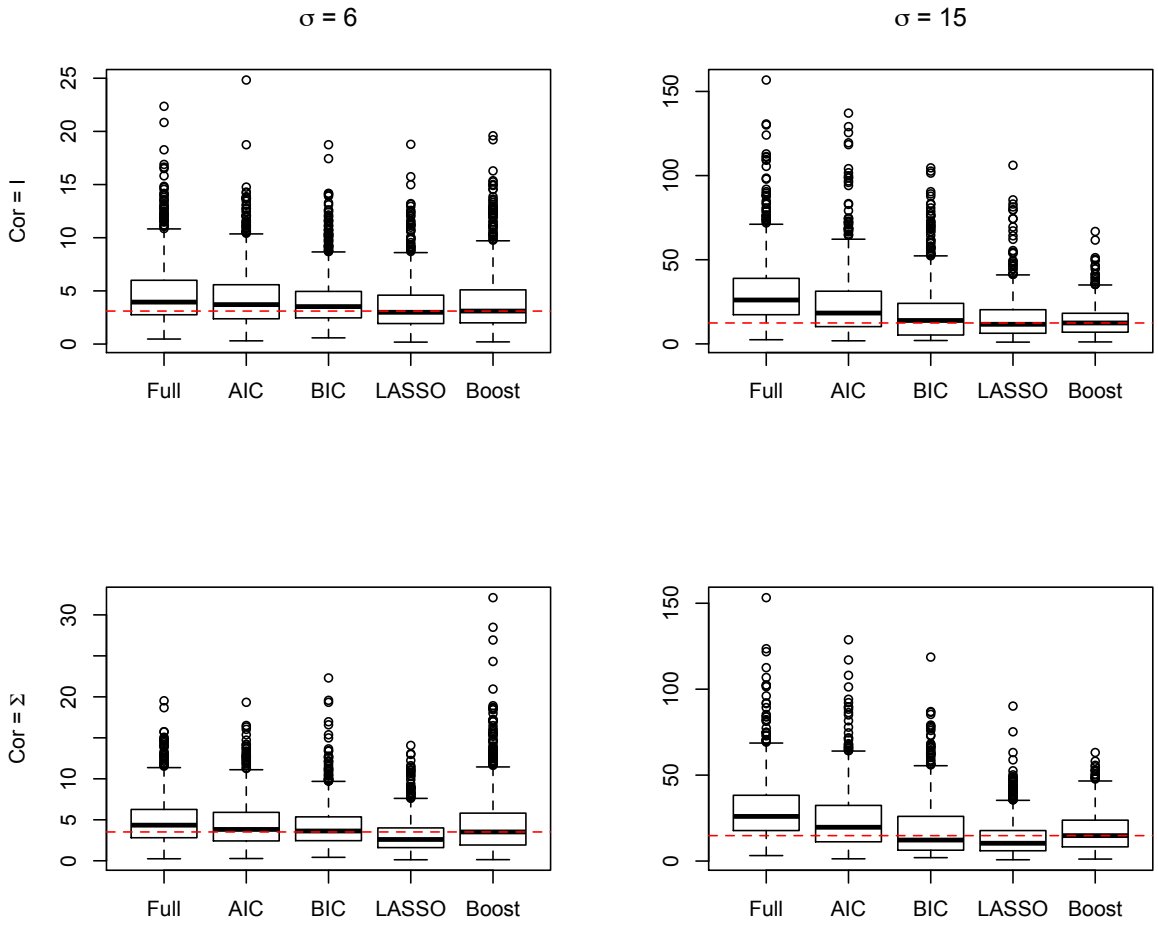

Figure 4.7: Simulation study results based on 1000 replications (Model C, N = 100). "Full" refers to fitting all interaction terms using a linear model; "Boosting" is the dedicated boosting algorithm.

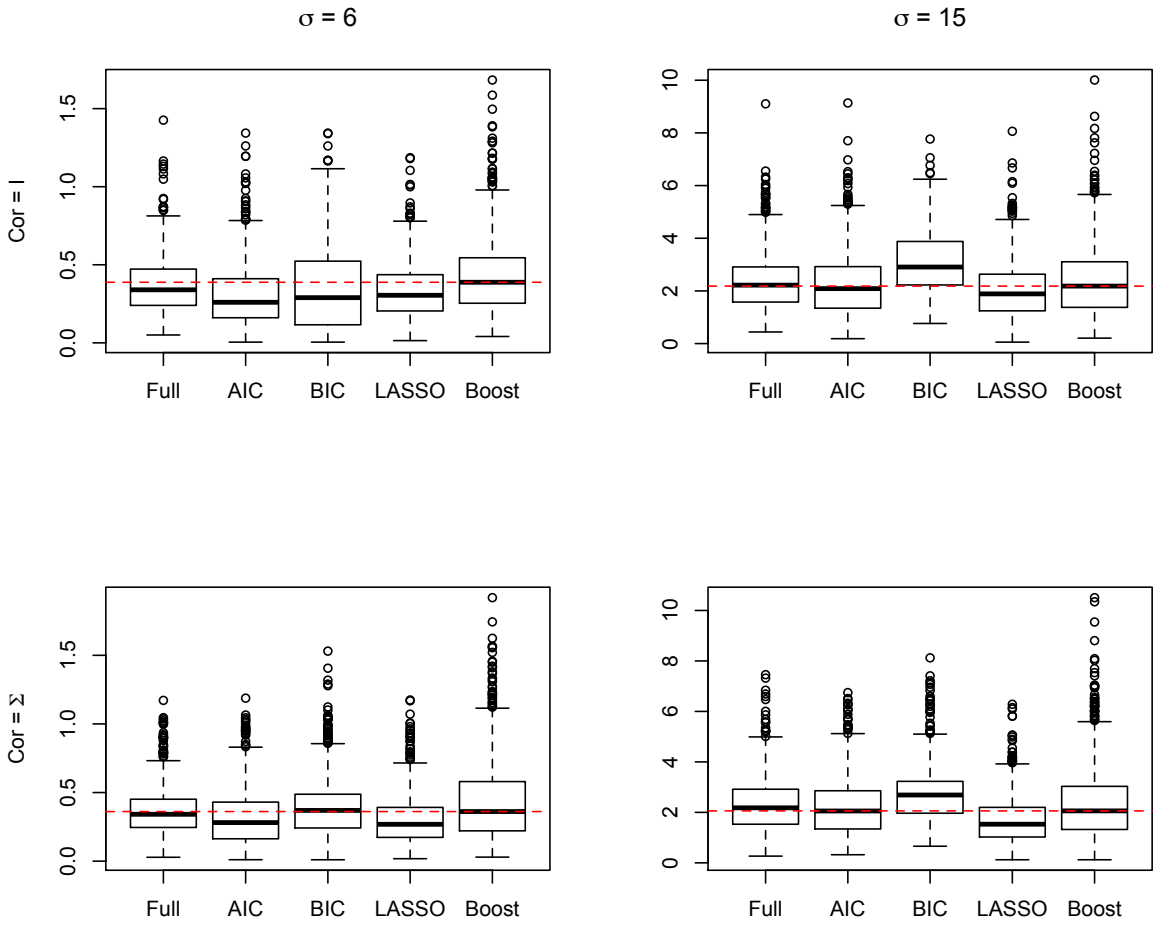

Figure 4.8: Simulation study results based on 1000 replications (Model C, N = 1000). "Full" refers to fitting all interaction terms using a linear model; "Boosting" is the dedicated boosting algorithm.

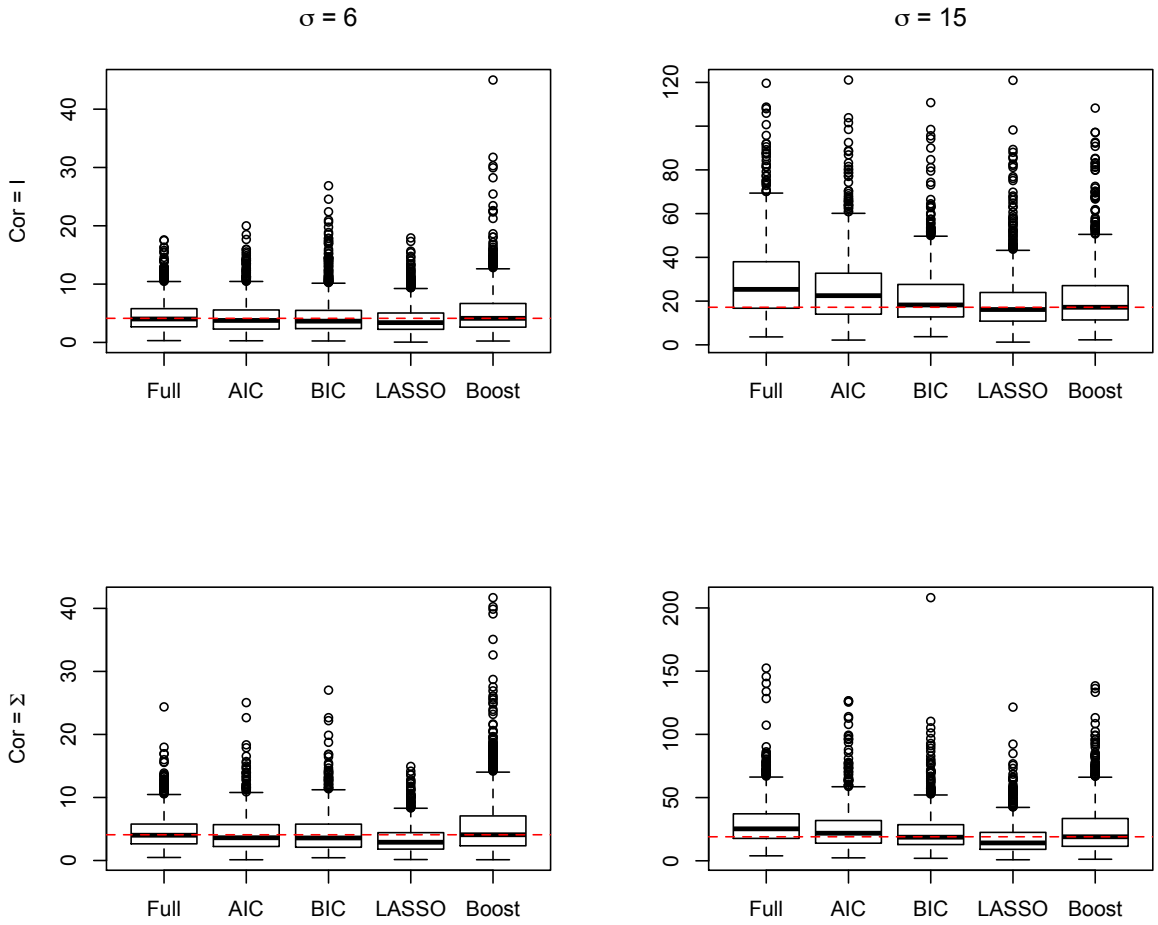

Figure 4.9: Simulation study results based on 1000 replications (Model D, N = 100). "Full" refers to fitting all interaction terms using a linear model; "Boosting" is the dedicated boosting algorithm.

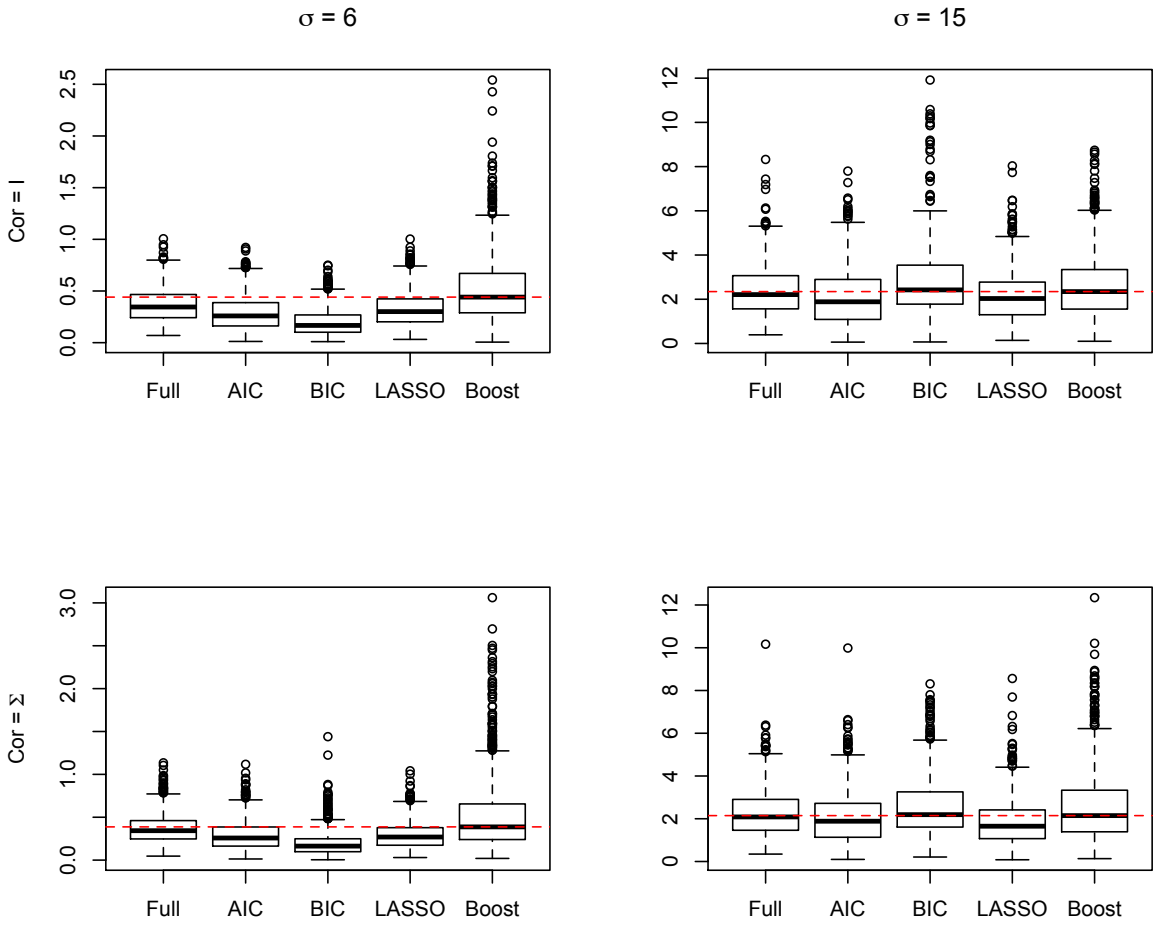

Figure 4.10: Simulation study results based on 1000 replications (Model D, N = 1000). "Full" refers to fitting all interaction terms using a linear model; "Boosting" is the dedicated boosting algorithm.

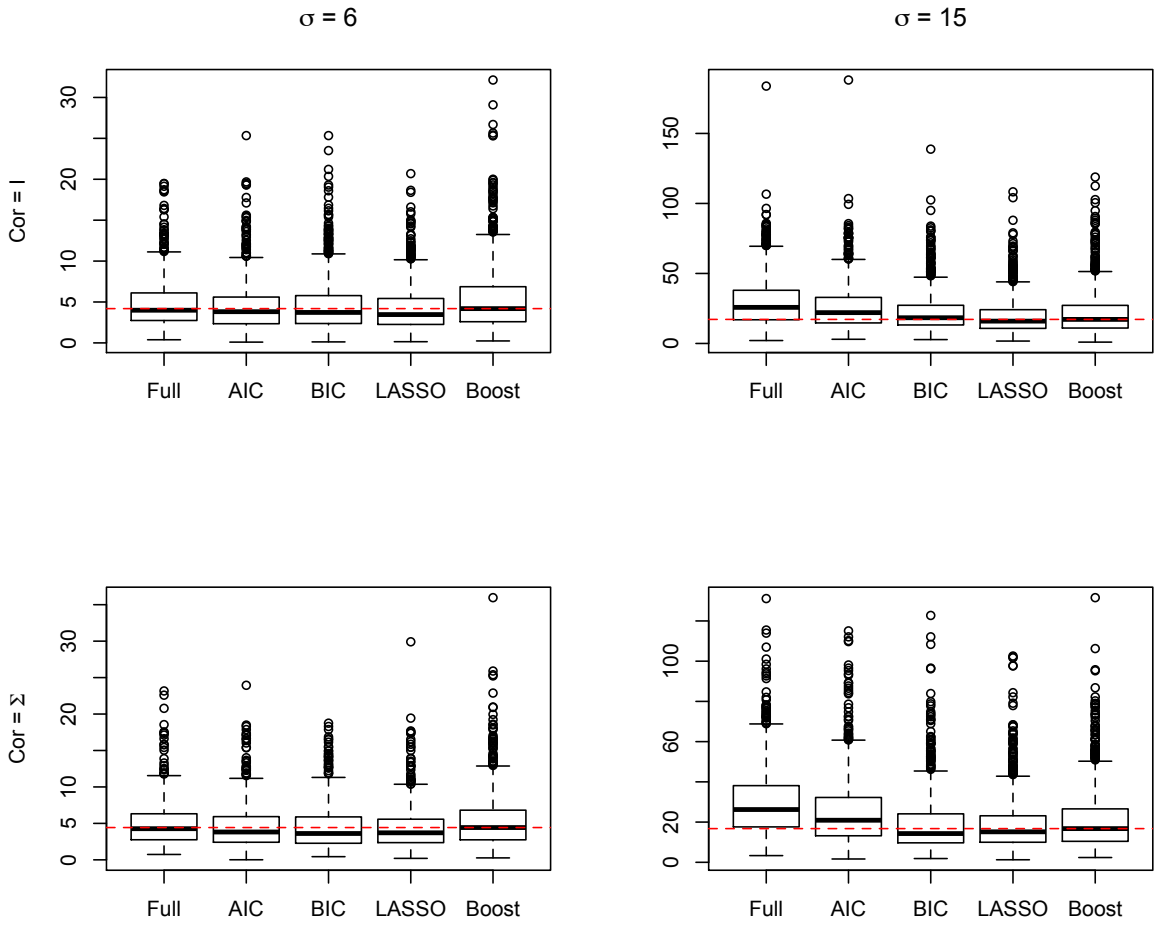

Figure 4.11: Simulation study results based on 1000 replications (Model E, N = 100). "Full" refers to fitting all interaction terms using a linear model; "Boosting" is the dedicated boosting algorithm.

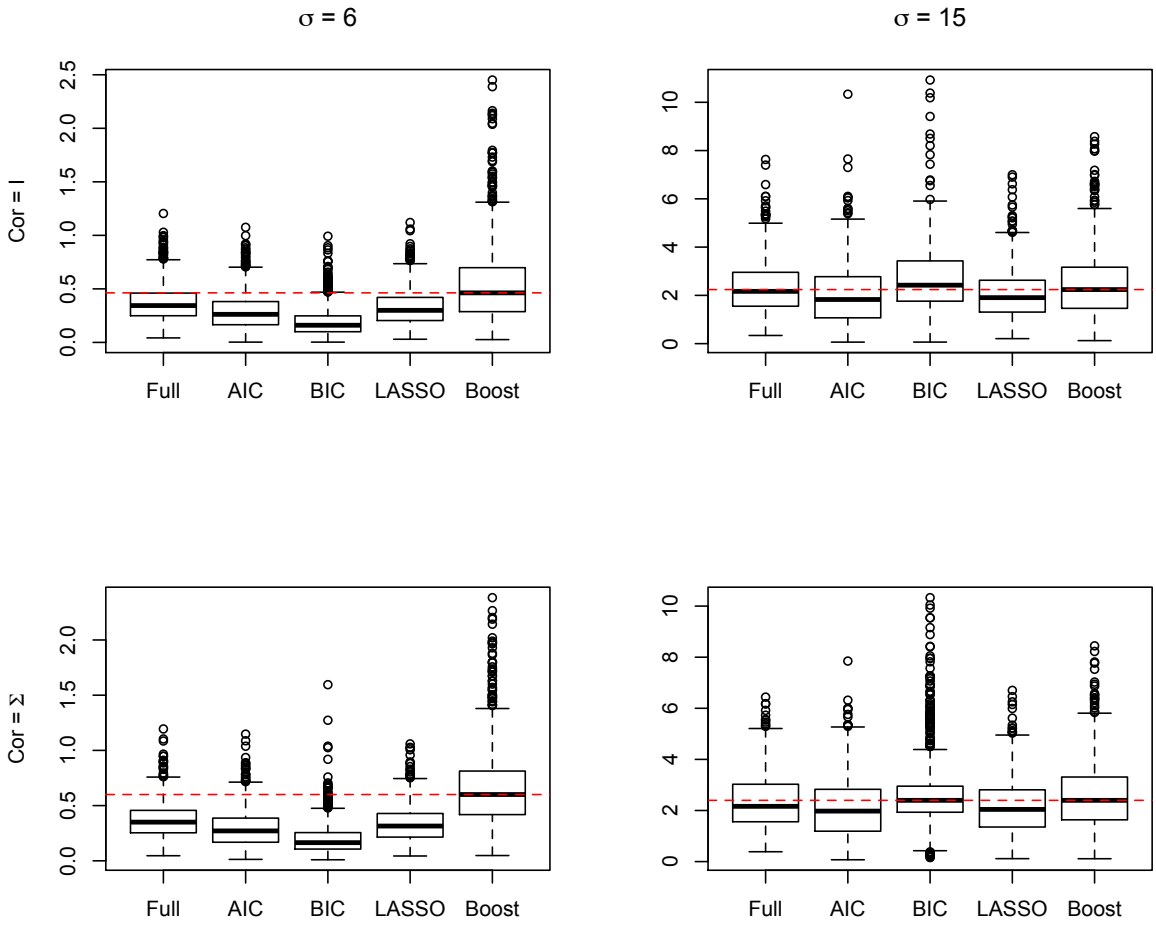

Figure 4.12: Simulation study results based on 1000 replications (Model E, N = 1000). "Full" refers to fitting all interaction terms using a linear model; "Boosting" is the dedicated boosting algorithm.