Generalized Lasso Regularization for Regression Models

Diplomarbeit

Claudia Flexeder

Betreuung:

Prof. Dr. Gerhard Tutz

M.Sc. Sebastian Petry

Ludwig-Maximilians-Universität München
Institut für Statistik

Contents

1. Introduction

Lasso regularization [Tib96] is one of the most commonly used regularization methods, because it includes the technique of variable selection. All those coefficients whose corresponding predictors have vanishing or low influence on the response are shrunk to zero. By introducing the elastic net [ZH05], a regularization technique which additionally shows the grouping property was proposed. Thereby, the absolute values of the coefficients are estimated nearly equal if the corresponding predictors are highly correlated.

In this thesis, a new regularization method, the *pairwise fused lasso* [Pet09], is presented which has both the variable selection and the grouping property. The goal of this thesis is to examine the performance of the pairwise fused lasso and to select appropriate weights for its penalty term. Furthermore, the pairwise fused lasso solutions based on two different approximation procedures (LQA [Ulb10b] and LARS [EHJT04]) are compared.

The remainder of this thesis is organized as follows. Chapter 2 gives an overview of already established regularization techniques and proposes the LARS algorithm for solving penalized least squares problems. In Chapter 3 the pairwise fused lasso and its modifications with respect to additional weights are presented. Moreover, the computational approach for solving the penalized regression problem is discussed. Chapter 4 gives a brief summary of generalized linear model theory. Furthermore, the local quadratic approximation approach for fitting penalized generalized linear models is described. For studying the performance of the new regularization method, simulations based on the linear model are presented in Chapter 5 whereas simulations based on generalized linear models, especially the logit and Poisson model, are given in Chapter 6. Chapter 7 comprises real data examples. And finally we conclude in Chapter 8.
Tables and Figures which illustrate the simulation results described in Chapters 5 and 6 can be found in Appendix A and Appendices B, C, respectively.
Furthermore, the accompanying CD contains the R-code and functions for the simulation studies and the computation of the corresponding measures.

2. Regularization Methods

2.1. Motivation

Classical linear regression assumes that the response vector $\boldsymbol{y} = (y_1, \ldots, y_n)^{\mathrm{T}}$ is a linear combination of p regressors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p$ and an unknown parameter vector $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^{\mathrm{T}}$ as well as an additive error term $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)^{\mathrm{T}}$. Hence the normal regression model is given by

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{2.1}$$

where $\boldsymbol{\epsilon} \sim N_n(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$ and the design matrix $\boldsymbol{X} = \left(\boldsymbol{x}_1 | \ldots | \boldsymbol{x}_p\right)$ which is based on n iid observations.
The response is centered and the predictors are standardized. Consequently:

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

Since the response is centered and the predictors are standardized, no intercept has to be estimated.
The usual estimation procedure for the parameter vector $\boldsymbol{\beta}$ is the minimization of the residual sum of squares with respect to $\boldsymbol{\beta}$:

$$\widehat{\boldsymbol{\beta}}_{OLS} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\right)^{\mathrm{T}} \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\right). \tag{2.2}$$

Then, the ordinary least squares (OLS) estimator $\widehat{\boldsymbol{\beta}}_{OLS} = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1} \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}$ is obtained by solving the estimation equation

$$\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right) \boldsymbol{\beta} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}. \tag{2.3}$$

The OLS estimator is optimal within the class of linear unbiased estimators if the predictors are orthogonal. However, if there are highly correlated predictors in the regression model, multi-collinearity occurs. This can lead to problems in the computation of the OLS estimator. For this purpose, we distinguish the following two situations: *exact multi-collinearity* and *approximate multi-collinearity*. In the case of exact multi-collinearity, two (or more) predictors $\boldsymbol{x}_i$, $\boldsymbol{x}_j$ are linearly dependent. Consequently, both the design matrix $\boldsymbol{X}$ and the matrix $(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})$ no longer have full rank p. Thus, the inverse $\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}$ cannot be calculated, Equation (2.3) cannot be solved and the OLS estimator has not a unique solution. If there is only an approximate linear dependence between several variables, we have to deal with approximate multi-collinearity. In this case, $\boldsymbol{X}$ is of full rank and $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ is regular, such that Equation (2.3) has a unique solution. But due to this almost linear dependence, the determinant

$|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}|$ reaches a value near zero and the OLS estimator exhibits a very large variance, $\mathrm{var}(\widehat{\boldsymbol{\beta}}_{OLS}) = \sigma^2(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}$. According to Hoerl and Kennard [HK70], the length of the OLS vector tends to be longer than the length of the true parameter vector, i.e. $\|\boldsymbol{\beta}_{OLS}\| \geq \|\boldsymbol{\beta}_{true}\|$. Since the determinant is defined by $|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}| = \prod_{i=1}^{n} \lambda_i$ [FKL07], at least one eigenvalue λ_i tends to be very small in the case of approximate multi-collinearity. Thus, a measure of collinearity is the condition number κ [Tou03] which is given by

$$\kappa(\boldsymbol{X}) = \left(\frac{\lambda_{max}}{\lambda_{min}}\right)^{1/2}. \tag{2.4}$$

Clearly, $\kappa \geq 1$, with large values suggesting approximate multi-collinearity.
A further drawback of the usual estimation procedure is the lack of variable selection. Even coefficients whose corresponding predictors have vanishing or low influence on the response remain in the regression model. With a large number of predictors, we would like to determine a parsimonious model which is easier to interpret. Indeed, subset selection produces a sparse model but it is extremely variable because it is a discrete process [Bre96].
To overcome these problems [HK70, Bre96], regression modeling by regularization techniques was proposed. The regularization methods are based on penalty terms and should yield unique estimates of the parameter vector $\boldsymbol{\beta}$. Furthermore, an improvement of the prediction accuracy can be achieved by shrinking the coefficients or setting some of them to zero. Thereby we obtain regression models which should contain only the strongest effects and which are easier to interpret. In the following, an overview of some already established regularization techniques is given.

2.2. Penalized Least Squares

Regularization approaches for normal regression problems are based on penalized least squares

$$PLS\left(\lambda, \boldsymbol{\beta}\right) = \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\right)^{\mathrm{T}}\left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\right) + P\left(\lambda, \boldsymbol{\beta}\right) \tag{2.5}$$

and estimates of the parameter vector $\boldsymbol{\beta}$ are obtained by minimizing this equation, i.e.

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\mathrm{argmin}} \left\{PLS\left(\lambda, \boldsymbol{\beta}\right)\right\}. \tag{2.6}$$

The penalty term $P\left(\lambda, \boldsymbol{\beta}\right)$ depends on the tuning parameter λ which controls the shrinkage intensity. For the tuning parameter $\lambda = 0$ we obtain the ordinary least squares solution. On the contrary, for large values of λ the influence of the penalty term on the coefficient estimates increases. Hence, the penalty region determines the properties of the estimated parameter vector, whereas desirable features are variable selection and a grouping effect. An estimator shows the grouping property if it tends to estimate the absolute value of coefficients (nearly) equal if the corresponding predictors are highly correlated.

2.2.1. Ridge Regression

One of the most popular alternative solutions to ordinary least squares estimates is *ridge regression* introduced by Hoerl and Kennard [HK70].
Because of the $\|\boldsymbol{\beta}_{OLS}\| \geq \|\boldsymbol{\beta}_{true}\|$ problem described in section 2.1, ridge regression designs its penalty term in such a way that the length of the parameter vector $\boldsymbol{\beta}$ is restricted. Consequently the ridge estimate is defined by

$$\widehat{\boldsymbol{\beta}}_{ridge} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \Big\{ \sum_{i=1}^{n} \big(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\big)^2 \Big\}, \;\; s.t. \; \sum_{j=1}^{p} \beta_j^2 \leq t, \; t \geq 0. \tag{2.7}$$

Equivalent to this constrained notation of ridge regression is the following penalized regression problem:

$$\widehat{\boldsymbol{\beta}}_{ridge} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \Big\{ \sum_{i=1}^{n} \big(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\big)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \Big\}, \; \lambda \geq 0. \tag{2.8}$$

Thus, the parameter t is clearly related to the parameter λ. This means that for a specific value λ there exists a value t such that the estimation equations (2.7) and (2.8) exhibit the same solution, i.e.

$$\widehat{\boldsymbol{\beta}}_{ridge} = \big(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} + \lambda \boldsymbol{I}\big)^{-1} \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}, \tag{2.9}$$

where $\boldsymbol{I}$ is the $p \times p$ identity matrix. By adding $\lambda\boldsymbol{I}$ to $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$, this results in a regular and invertible matrix even in both cases of multi-collinearity. Thus, ridge regression provides unique estimates in such situations.
Contrary to the ordinary least squares estimates the ridge estimator is not unbiased. Hence this regularization method accepts a little bias to reduce the variance and the mean squared error, respectively of the estimates and possibly improves the prediction accuracy. Due to this, the resulting model is less sensitive to changes in the data. Summarizing, ridge regression yields more stable estimates by shrinking coefficients, but does not select predictors and therefore does not give an easily interpretable model.
Because of the missing variable selection, further regularization techniques were developed as e.g. lasso regularization.

2.2.2. Lasso

The *least absolute shrinkage and selection operator* (lasso), proposed by Tibshirani [Tib96], does both continuous shrinkage and automatic variable selection simultaneously. As with the ridge regression the lasso estimates are obtained by minimizing the residual sum of squares subject to a constraint. Instead of the L_2-penalty, the lasso imposes the L_1-norm on the regression coefficients, i.e. the sum of the absolute value of the coefficients is restricted:

$$\widehat{\boldsymbol{\beta}}_{lasso} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \Big\{ \sum_{i=1}^{n} \big(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\big)^2 \Big\}, \;\; s.t. \; \sum_{j=1}^{p} |\beta_j| \leq t, \; t \geq 0. \tag{2.10}$$

Or equivalently, the lasso determines the coefficient vector $\widehat{\boldsymbol{\beta}}_{lasso}$ satisfying

$$\widehat{\boldsymbol{\beta}}_{lasso} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \Big\{ \sum_{i=1}^{n} \big(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\big)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \Big\},\ \lambda \geq 0. \tag{2.11}$$

On the basis of the design of the constraint $\sum_{j=1}^{p} |\beta_j| \leq t$, values of $t < t_{OLS}$ with $t_{OLS} = \sum_{j=1}^{p} |\widehat{\beta}_{j,OLS}|$ cause a shrinkage of the coefficients. With decreasing values of the parameter t the estimated lasso coefficients are shrunk towards zero and some coefficients are exactly set to zero; for $t = 0$ all of them are equal to zero. Otherwise, a value of $t \geq t_{OLS}$ results in the unpenalized least squares estimates if the OLS estimator exists. In comparison to the parameter t, the parameter λ has the contrary effect on the estimation.

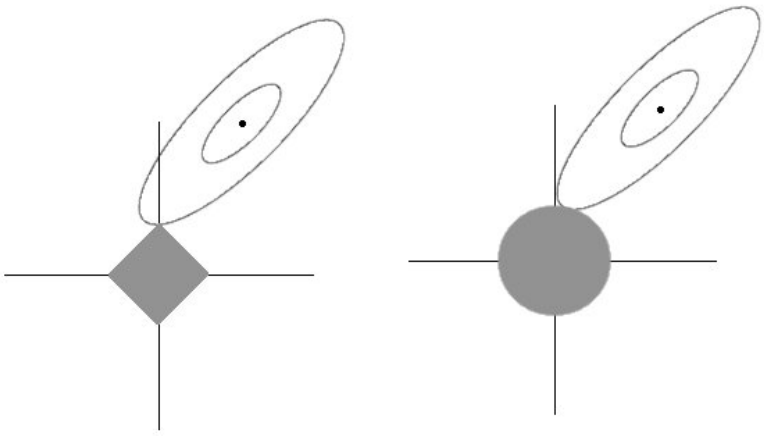

Figure 2.1.: Contours of the error and constraint regions for the lasso (left) and ridge regression (right).

For the two-dimensional case ($\mathbb{R}^2$), Figure 2.1 shows why the lasso exhibits the property to select predictors. The contours of the residual sum of squares are ellipses, centered at the ordinary least squares estimate. The constraint region for the lasso is the rotated square $|\beta_1| + |\beta_2| \leq t$, whereas that for ridge regression is the disk $\beta_1^2 + \beta_2^2 \leq t$. The first point where the elliptical contours touch the constraint region corresponds to the lasso and ridge solution, respectively. Since the first osculation point of the ellipses can be a vertex of the square, the lasso solution can have one coefficient β_j equal to zero. In contrast, ridge regression cannot produce zero solutions because there are no vertices in the constraint region that can be touched.

2.2.3. Elastic Net

Besides the advantage of variable selection, the lasso also has some limitations. As discussed by Tibshirani [Tib96] ridge regression dominates the lasso with regard to the prediction

accuracy in the usual $n > p$ case if there are high correlations among the variables. Another drawback of the lasso solution is the fact that in $p > n$ situations, it selects at most n variables. Moreover, the lasso does not group predictors as pointed out by Zou and Hastie [ZH05]. If there is a group of highly correlated predictors, the lasso tends to select only some arbitrary variables from this group.
A regularization and variable selection method which additionally shows the grouping property is the *elastic net*, presented by Zou and Hastie [ZH05]. The elastic net criterion is defined by

$$PLS_{enet}(\lambda_1, \lambda_2, \boldsymbol{\beta}) = \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \underbrace{\lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2}_{P_{enet}(\lambda_1, \lambda_2, \boldsymbol{\beta})}, \tag{2.12}$$

which depends on two tuning parameters $\lambda_1,\ \lambda_2 > 0$ and leads to the penalized least squares solution of the elastic net criterion

$$\widehat{\boldsymbol{\beta}}_{enet} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ PLS_{enet}(\lambda_1, \lambda_2, \boldsymbol{\beta}) \right\}. \tag{2.13}$$

The elastic net penalty is a convex combination of the lasso and ridge penalty and in constraint form given by $(1-\alpha)\sum_{j=1}^{p} |\beta_j| + \alpha \sum_{j=1}^{p} \beta_j^2 \leq t$ with $\alpha = \lambda_2 / (\lambda_1 + \lambda_2)$. For $\alpha = 1$ we obtain simple ridge regression, whereas for $\alpha = 0$ the lasso penalty is given. Equation (2.12) is called the naïve elastic net, because it is similar to either ridge regression or the lasso and tends to overshrink in regression problems. Thus, Zou and Hastie defined the elastic net estimates $\widehat{\boldsymbol{\beta}}$ as a rescaled naïve elastic net solution

$$\widehat{\boldsymbol{\beta}} = (1 + \lambda_2)\, \widehat{\boldsymbol{\beta}}_{enet}. \tag{2.14}$$

For various reasons [ZH05], $(1 + \lambda_2)$ is chosen as the scaling factor.
The interesting property of the elastic net is that it can select groups of correlated variables. According to Zou and Hastie the difference between the coefficient paths of predictors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is given by

$$\left| \widehat{\beta}_i(\lambda_1, \lambda_2) - \widehat{\beta}_j(\lambda_1, \lambda_2) \right| / \sum_{i=1}^{n} |y_i| \leq \frac{1}{\lambda_2} \sqrt{2\left(1 - \rho_{ij}\right)}, \tag{2.15}$$

where ρ_{ij} is the sample correlation and $\widehat{\beta}_i(\lambda_1, \lambda_2)$, $\widehat{\beta}_j(\lambda_1, \lambda_2)$ are the naïve elastic net estimates. If $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are highly correlated, i.e. $\rho_{ij} \to 1$, the coefficient paths of variable $\boldsymbol{x}_i$ and variable $\boldsymbol{x}_j$ are very close. In the case of negatively correlated predictors ($\rho_{ij} \to -1$), we consider $-\boldsymbol{x}_j$. Thus, the elastic net enforces that the regression coefficients of highly correlated variables tend to be equal, up to a sign if they are negatively correlated. Figure 2.2 illustrates the grouping effect and shows the coefficient paths of the lasso and the elastic net for the idealized example given by Zou and Hastie [ZH05]. The design matrix $\boldsymbol{X}$ in this

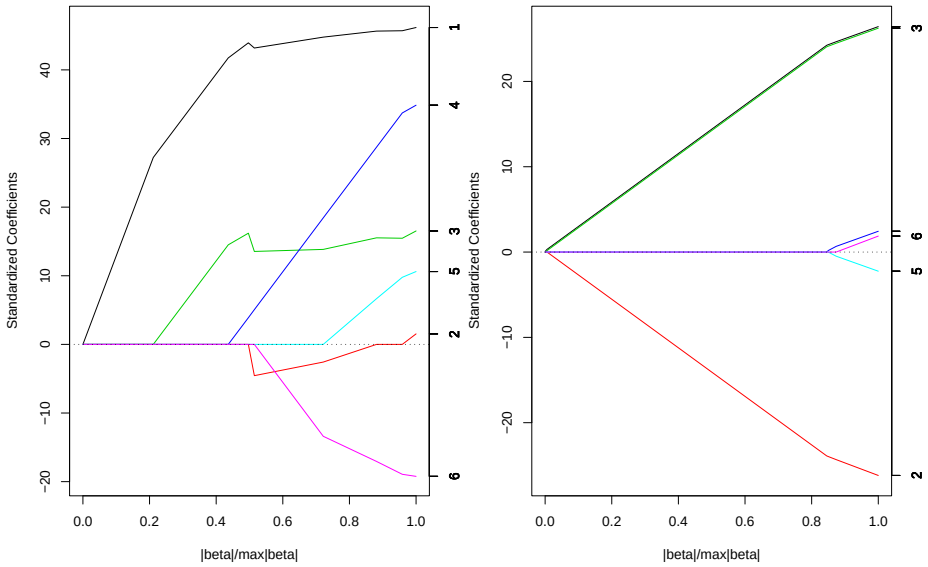

Figure 2.2.: Coefficient paths of lasso (left) and elastic net, $\lambda_2 = 0.5$ (right).

setting consists of

$$x_1 = Z_1 + \epsilon_1, \qquad x_2 = -Z_1 + \epsilon_2, \qquad x_3 = Z_1 + \epsilon_3,$$
$$x_4 = Z_2 + \epsilon_4, \qquad x_5 = -Z_2 + \epsilon_5, \qquad x_6 = Z_2 + \epsilon_6,$$

where Z_1, Z_2 are two independent $U(0,20)$ variables and ϵ_i, $i = 1,\ldots,6$ are independent and identically distributed $N(0, \frac{1}{16})$ for sample size $n = 100$. The response $\boldsymbol{y}$ for this model is generated as $N(Z_1 + 0.1Z_2, 1)$. The coefficient built-up of the elastic net in the right panel of figure 2.2 shows that this regularization method selects the influential predictors x_1, x_2, x_3 and yields the grouping effect. In contrast, from the solution paths in the left panel can be seen that the lasso paths are very unstable and that correlations within a group of predictors cannot be identified.
For solving the penalized least squares problem in equation (2.12), the given data set $(\boldsymbol{y}, \boldsymbol{X})$ is extended to an artificial data set $(\boldsymbol{y}^*, \boldsymbol{X}^*)$, whereas the $(n+p)$-dimensional vector $\boldsymbol{y}^*$ and the $(n+p) \times p$-matrix $\boldsymbol{X}^*$ are defined by

$$\boldsymbol{X}^* = (1+\lambda_2)^{-1/2} \begin{pmatrix} \boldsymbol{X} \\ \sqrt{\lambda_2}\boldsymbol{I} \end{pmatrix}, \qquad \boldsymbol{y}^* = \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{0} \end{pmatrix}. \tag{2.16}$$

Because of this augmentation the elastic net problem can be written as a lasso type problem and solved with the LARS algorithm [EHJT04]. Hence, the elastic net can select all p predictors in the $p > n$ case and not only n variables as the lasso, since the transformed matrix $\boldsymbol{X}^*$ has rank p. Further details to the algorithm LARS are given in section 2.3.

2.3. LARS Algorithm

Least angle regression [EHJT04] is a model selection algorithm which is associated with forward stepwise regression. According to Weisberg [Wei85], forward stepwise regression builds a model sequentially, adding one variable at a time. At the beginning, from a set of possible predictors we choose the one having largest absolute correlation with the response $\boldsymbol{y}$ and fit a linear model. Next, we add to the model that variable which has the largest absolute partial correlation with the response, adjusted for the predictors which are already included in the model. Thus, after k steps, k variables are selected according to this criterion for the partial correlations. Then they are used to build a k-parameter linear model.
The least angle regression (LARS) algorithm is based on a similar strategy, but "only enters as much of a predictor as it deserves" [HTF09]. Thus, it consists of the following steps:

1. We standardize the predictors to have mean zero and unit norm. Then, we start with the residual $\boldsymbol{r} = \boldsymbol{y} - \bar{\boldsymbol{y}}$ and all coefficients equal to zero $(\beta_1, \beta_2, \ldots, \beta_p = 0)$ as with forward stepwise regression.

2. We identify the predictor $\boldsymbol{x}_j$ most correlated with $\boldsymbol{r}$.

3. Instead of fitting $\boldsymbol{x}_j$ completely, we move the coefficient β_j continuously from 0 towards its least squares value $\langle \boldsymbol{x}_j, \boldsymbol{r} \rangle$, until some other competitor $\boldsymbol{x}_k$ has as much correlation with the current residual as does $\boldsymbol{x}_j$.

4. We move the corresponding coefficients β_j and β_k in the direction defined by their joint least squares coefficient of the current residual on $(\boldsymbol{x}_j, \boldsymbol{x}_k)$, until some other competitor $\boldsymbol{x}_l$ has as much correlation with the current residual. I.e. the algorithm proceeds in a direction *equiangular* between the predictors $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$.

5. We continue in this way until all the predictors are in the model and we arrive at the full least squares solution in $n > p$ situations after p steps. In the $p > n$ case, the algorithm reaches a zero residual solution after $n - 1$ steps. The number of steps is $n - 1$ rather than n because the design matrix has been centered and hence it has row rank $n - 1$.

To implement this equiangular strategy, Hastie et al. [HTF09] describe the algebra of the algorithm as follows: at the beginning of the kth step, $\mathcal{A}_k$ is the active set of variables and $\boldsymbol{\beta}_{\mathcal{A}_k}$ the coefficient vector for these variables; $k - 1$ coefficients are nonzero and only the just inserted coefficient equals zero. The direction δ_k for this step is defined by

$$\delta_k = \left(\boldsymbol{X}_{\mathcal{A}_k}^{\mathrm{T}} \boldsymbol{X}_{\mathcal{A}_k} \right)^{-1} \boldsymbol{X}_{\mathcal{A}_k}^{\mathrm{T}} \boldsymbol{r}_k, \tag{2.17}$$

where $\boldsymbol{r}_k = \boldsymbol{y} - \boldsymbol{X}_{\mathcal{A}_k} \boldsymbol{\beta}_{\mathcal{A}_k}$ is the current residual. Then, the coefficient profile turns out to be $\boldsymbol{\beta}_{\mathcal{A}_k}(\alpha) = \boldsymbol{\beta}_{\mathcal{A}_k} + \alpha \cdot \delta_k$. The parameter α denotes the step size which makes the current residual equally correlated with the variables in the active set and another competitor. Owing to the piecewise linearity of the algorithm and information of the covariance of the predictors, the step size can be exactly calculated at the beginning of each step. If the fit vector at the

beginning of this step is $\hat{f}_k$, then it evolves to

$$\hat{f}_k(\alpha) = f_k + \alpha \cdot u_k, \tag{2.18}$$

with $u_k = \boldsymbol{X}_{\mathcal{A}_k}\delta_k$ the new fit direction. This vector u_k denotes the smallest and equal angle with each of the variables in the active set $\mathcal{A}_k$. Therefore the name "least angle regression" was chosen.

By a simple modification of the LARS algorithm the entire lasso path can be generated. For this purpose, Hastie et al. [HTF09] proceed as follows:
If the predictors are standardized, the LARS algorithm can be equivalently stated in terms of inner products instead of correlations. Let $\mathcal{A}$ be the active set of variables at any step of the algorithm. The inner product of these variables with the current residual $\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}$ can be computed by

$$\boldsymbol{x}_j^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) = \alpha \cdot s_j, \ \forall j \in \mathcal{A}, \tag{2.19}$$

where $s_j \in \{-1, 1\}$ denotes the sign of the inner product and α indicates the common step size. Furthermore, $|\boldsymbol{x}_j^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})| \leq \alpha, \ \forall k \notin \mathcal{A}$. The lasso criterion in equation (2.11) can be written in vector form, i.e.

$$R(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1. \tag{2.20}$$

Let $\mathcal{B}$ be the active set of variables in equation (2.20) for a specified value of the tuning parameter λ. For these variables, $R(\boldsymbol{\beta})$ is differentiable and the stationary conditions yield

$$\boldsymbol{x}_j^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) = \lambda \cdot sign\left(\beta_j\right), \ \forall j \in \mathcal{B}. \tag{2.21}$$

If the sign of any nonzero coefficient β_j agrees with the sign s_j of the inner product, equation (2.19) and (2.21) are the same. However, if the coefficient of an active variable passes through zero, the LARS algorithm and lasso differ. Hence, constraint (2.21) is violated for the corresponding variable and this variable is excluded from the active set $\mathcal{B}$. The stationary conditions for the variables which are not in the active set require that

$$|\boldsymbol{x}_j^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})| \leq \lambda, \ \forall k \notin \mathcal{B}, \tag{2.22}$$

which again agrees with the unmodified LARS algorithm. Thus, the lasso modification allows the active set $\mathcal{B}$ to decrease, whereas the active set $\mathcal{A}$ grows monotonically as the unmodified LARS algorithm progresses. LARS is computationally very efficient since it requires only the same order of magnitude of computational effort as ordinary least squares applied to the full set of variables.

3. Pairwise Fused Lasso

In the subsequent chapters of this thesis a new regularization technique, the *pairwise fused lasso*, is presented. The fused lasso from Tibshirani et al. [TSR+05] can be considered as motivation for this new method. Besides the computational approach for solving the new penalized regression problem, modifications of the pairwise fused lasso penalty with regard to additional weights are discussed.

3.1. Fused Lasso

Another technique for regularized regression with both the grouping and the selection property is the *fused lasso*. Since the lasso ignores the order of predictors, Tibshirani et al. [TSR+05] generalized the lasso penalty to those ordered situations. The fused lasso penalty is defined by

$$P_{FL}(\lambda_1, \lambda_2, \boldsymbol{\beta}) = \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=2}^{p} |\beta_j - \beta_{j-1}|, \tag{3.1}$$

where the predictors $\boldsymbol{x}_j$, $j = 1, \ldots, p$ can be ordered in some meaningful way. For instance for functional data, the predictors $\boldsymbol{x}_j(t)$ are ordered according to some index variable t. Besides the absolute values of the coefficients, the fused lasso also penalizes the L_1-norm of their consecutive differences. Hence, the clustering property is motivated by the adjacency of predictors. The first term of the penalty encourages sparsity of the regression coefficients, whereas the second term encourages sparsity of their adjacent differences. Hence, some components β_j are exactly zero and the coefficient profile is piecewise constant. The contours of the sum of squares loss function and the contours of the constraint region for the fused lasso penalty are shown in Figure 3.1. As with ridge regression and lasso regularization, the first osculation point of the elliptical contours of the residual sum of squares with the constraint region corresponds to the fused lasso solution. The regression coefficients in Figure 3.1, determined by the ellipses, satisfy the lasso penalty $\sum_{j=1}^{p} |\beta_j| = t_1$ (gray square) and the difference penalty $\sum_{j=2}^{p} |\beta_j - \beta_{j-1}| = t_2$ (black rectangle), where t_1 and t_2 are the tuning parameters. As mentioned in section 2.2.2, the lasso selects no more than $min(n, p)$ variables. This is a drawback in the $p > n$ case. A similar behavior shows the fused lasso solution. In high dimensional situations, the fused lasso selects at most p sequences of non-zero coefficients instead of p non-zero components β_j like the lasso. Besides this sparsity property, the application of the least angle regression algorithm [EHJT04] is a further similarity to lasso regularization. The complete sequence of lasso and fusion problems can be solved with this algorithm. By transforming $\boldsymbol{X}$ to $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{L}^{-1}$, applying the LARS procedure and then transforming back, the fusion problem is solved. Thereby, the $p \times (p-1)$-matrix $\boldsymbol{L}$

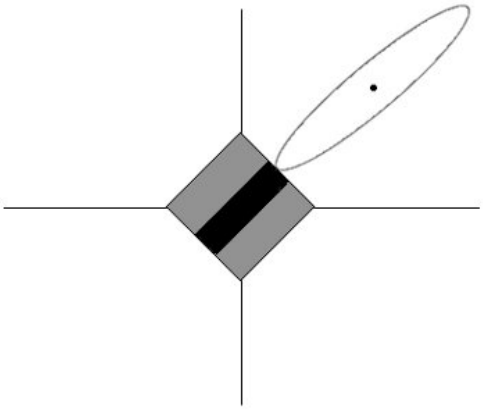

Figure 3.1.: Contours of the error and constraint region for the fused lasso.

has elements $L_{ii} = 1$, $L_{i+1,i} = -1$ and $L_{ij} = 0$ otherwise. Thus, in consequence $\boldsymbol{\theta} = \boldsymbol{L}\boldsymbol{\beta}$. For further details to this computational approach we refer to Tibshirani et al. [TSR+05].
In addition, the quantity of degrees of freedom used for fitting $\hat{\boldsymbol{y}}$ is of interest. Generally, the number of degrees of freedom is the difference between the number of cases and the number of parameters in the model [Wei85]. For the fused lasso one degree of freedom is defined as a sequence of one or more successive coefficients $\widehat{\beta}_j$ which are non-zero and equal to one another, i.e.

$$df\,(\hat{\boldsymbol{y}}) = p - \#\{\beta_j = 0\} - \#\{\beta_j - \beta_{j-1} = 0, \beta_j, \beta_{j-1} \neq 0\}. \tag{3.2}$$

More on degrees of freedom, especially effective degrees of freedom, will be described in chapter 5.

3.2. Pairwise Fused Lasso Penalty

The *pairwise fused lasso* [Pet09] is a generalization of the fused lasso penalty to situations where the predictors $\boldsymbol{x}_j$, and hence the corresponding parameters β_j, have no natural order in j. In this generalized formulation, the fusion refers to all possible pairs of predictors and not only to adjacent ones. Thus, the pairwise fused lasso penalty is defined by

$$P_{PFL}(\lambda_1, \lambda_2, \boldsymbol{\beta}) = \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=2}^{p} \sum_{k=1}^{j-1} |\beta_j - \beta_k|, \quad \lambda_1, \lambda_2 > 0. \tag{3.3}$$

An equivalent manner of representation is the following:

$$P_{PFL}(\lambda, \alpha, \boldsymbol{\beta}) = \lambda \left[\alpha \sum_{j=1}^{p} |\beta_j| + (1-\alpha) \sum_{j=2}^{p} \sum_{k=1}^{j-1} |\beta_j - \beta_k| \right], \quad \lambda > 0. \tag{3.4}$$

Thereby, λ and α with $\alpha \in [0,1]$ are the tuning parameters instead of λ_1 and λ_2 as in Equation (3.3). On the one hand, the first summand of the pairwise fused lasso penalty conforms to the lasso penalty and accounts for variable selection. On the other hand, the second summand is the sum of the absolute values of all $\binom{p}{2}$ pairwise differences of regression coefficients. This part of the penalty term induces clustering.

3.3. Selection of Weights

Possibly to achieve an improvement of the prediction accuracy of the model fitted on the test data or of the mean squared error of the estimated parameter vector, the pairwise fused lasso penalty can be modified by adding different weights. Accordingly, a modification of this penalty term is given by

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

where w_j and w_{jk} are the additional weights. One possibility is to choose $w_j = |\beta_j^{ML}|^{-1}$ and $w_{jk} = |\beta_j^{ML} - \beta_k^{ML}|^{-1}$, where β_j^{ML} denotes the maximum likelihood estimates of the regression coefficients. This choice is motivated by the *adaptive lasso* [Zou06] and its oracle properties. Further, an estimation procedure obeys the oracle properties if it identifies the correct subset model $\mathcal{A}$, i.e. $\{j : \hat{\beta}_j \neq 0\} = \mathcal{A}$, and if it has the optimal estimation rate. This rate is given by $\sqrt{n}(\widehat{\boldsymbol{\beta}}_{\mathcal{A}} - \boldsymbol{\beta}_{\mathcal{A}}^*) \xrightarrow{d} N(\mathbf{0}, \boldsymbol{\Sigma}^*)$, where $\boldsymbol{\Sigma}^*$ is the covariance matrix knowing the true subset model and $\boldsymbol{\beta}_{\mathcal{A}}^*$ denotes the subset of the true parameter vector elements. As pointed out by Zou [Zou06], these oracle properties do not hold for the lasso but for the adaptive lasso if data-dependent weights are chosen. Hence, in contrast to the primary definition of the pairwise fused lasso where all parameters have the same amount of shrinkage (3.4), the weighted formulation (3.5) forces the coefficients to be unequally penalized by assigning different weights to different components. Large values of $|\beta_j^{ML}|$ yield small weights w_j and consequently a decreasing shrinkage of the corresponding parameters. If the maximum likelihood estimates of the jth and the kth predictor have nearly the same value, the weight w_{jk} causes a large influence of the difference penalty term.
Another possibility is to convert the pairwise fused lasso to a correlation based penalty. For instance, the elastic net shows a relationship between correlation and grouping where strongly correlated covariates tend to be in or out of the model together, but the correlation structure is not used explicitly in the penalty term. A regularization method, which is based on the idea that highly correlated covariates should have (nearly) the same influence on the response except for their sign, is the *correlation based penalty* proposed by Tutz and Ulbricht [TU09]. Coefficients of two predictors are weighted according to their marginal correlation. As a result, the intensity of penalization depends on the correlation structure. Therefore, the penalty term of the pairwise fused lasso can be transformed into

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

where ρ_{jk} denotes the marginal correlation between the jth and the kth predictor. The marginal correlation is given by

$$\rho_{jk} = \rho\left(\boldsymbol{x}_j, \boldsymbol{x}_k\right) = \frac{\operatorname{cov}\left(\boldsymbol{x}_j, \boldsymbol{x}_k\right)}{\sqrt{\operatorname{var}\left(\boldsymbol{x}_j\right)}\sqrt{\operatorname{var}\left(\boldsymbol{x}_k\right)}}. \tag{3.7}$$

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

The marginal correlation, used in the penalty term, is the familiar measure of coherence. It measures the interaction between the predictors $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$ without taking further covariates into account. But the coherence can be of another type if all influential features are included in the analysis. In contrast to the marginal correlation, the partial correlation determines to what extent the correlation between two variables depends on the linear effect of the other covariates. Thereby, the aim is to eliminate this linear effect [Rei06]. For this reason, it makes sense to investigate the correlation based penalty in Equation (3.6) also with partial correlations instead of the marginal ones.
To define the partial correlations, we consider p regressors, $\boldsymbol{x} = \left(x_1, \ldots, x_p\right)$, with expectation $\mathrm{E}(\boldsymbol{x}) = \boldsymbol{\mu}$ and covariance matrix $\operatorname{cov}(\boldsymbol{x}) = \boldsymbol{\Sigma}$. The inverse covariance matrix $\operatorname{cov}(\boldsymbol{x})^{-1} = \boldsymbol{\Sigma}^{-1}$ is known as the concentration or precision matrix where c_{jk} denote the corresponding elements, $\boldsymbol{\Sigma}^{-1} = (c_{jk})$. According to Whittaker [Whi90], the following two definitions describe the relationship between the elements of the inverse covariance matrix and the partial correlations:

- Each diagonal element of $\boldsymbol{\Sigma}^{-1}$ is the reciprocal of the partial variance of the corresponding variable predicted from the rest,

$$c_{jj} = \frac{1}{\operatorname{var}\left(x_j | \boldsymbol{x}_{-j}\right)}, \tag{3.8}$$

  where $\boldsymbol{x}_{-j}$ denotes the vector $\boldsymbol{x}$ without the jth component, i.e.
  $\boldsymbol{x}_{-j} = \left(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_p\right)$.

- Each off-diagonal element of $\boldsymbol{\Sigma}^{-1}$ is the negative of the partial correlation between the two corresponding variables x_j and x_k after adjustment for all the remaining covariates and scaled by the associated inverse partial variances:

$$c_{jk} = -\sqrt{c_{jj}c_{kk}}\,\rho\left(x_j, x_k | \boldsymbol{x}_{-jk}\right) \tag{3.9}$$

  and

$$\rho\left(x_j, x_k | \boldsymbol{x}_{-jk}\right) = -\frac{c_{jk}}{\sqrt{c_{jj}c_{kk}}}, \tag{3.10}$$

  respectively. Thereby, $\boldsymbol{x}_{-jk}$ is the vector $\boldsymbol{x}$ without the jth and the kth predictor.

Thus, by scaling the elements of a matrix with its diagonal elements, the partial correlations can be calculated in a similar manner as the marginal correlations. Moreover, the precision matrix shows if two variables are partially uncorrelated. In such cases the elements next to the respective diagonal elements are zero.

In addition to the empirical partial correlations we consider regularized partial correlations. According to Ledoit and Wolf [LW04], when the number of variables is much larger than the number of observations, the covariance matrix is estimated with a lot of error. Thus, the empirical covariance matrix cannot be considered a good approximation of the true covariance matrix. An approach for a shrinkage estimator of the covariance matrix is given by Schäfer and Strimmer [SS05] and Opgen-Rhein and Strimmer [ORS07]. The procedure for obtaining regularized variances and covariances is described in the following:

$$\sigma^*_{jk} = \rho^*_{jk}\sqrt{\sigma^*_{jj}\sigma^*_{kk}} \tag{3.11}$$

with

$$\begin{aligned} \rho^*_{jk} &= (1-\lambda_1^*)\,\rho_{jk} \\ \sigma^*_{jj} &= \lambda_2^*\sigma_{median} + (1-\lambda_2^*)\,\sigma_{jj} \end{aligned}$$

and

$$\begin{aligned} \lambda_1^* &= \min\left(1, \frac{\sum_{j\neq k}\widehat{\mathrm{var}}\left(\rho_{jk}\right)}{\sum_{j\neq k}\rho_{jk}^2}\right) \\ \lambda_2^* &= \min\left(1, \frac{\sum_{j=1}^{p}\widehat{\mathrm{var}}\left(\sigma_{jj}\right)}{\sum_{j=1}^{p}\left(\sigma_{jj}-\sigma_{median}\right)^2}\right) \end{aligned}$$

Let x_{ij} be the ith observation of the predictor $\boldsymbol{x}_j$ and $\bar{x}_j$ its empirical mean. With

$$w_{ijk} = (x_{ij}-\bar{x}_j)(x_{ik}-\bar{x}_k) \quad \text{and} \quad \bar{w}_{jk} = \frac{1}{n}\sum\nolimits_{i=1}^{n} w_{ijk}, \tag{3.12}$$

the unbiased empirical covariance equals $\sigma_{jk} = \frac{n}{n-1}\bar{w}_{jk}$. Correspondingly, we obtain the empirical variance σ_{jj} and correlation $\rho_{jk} = \sigma_{jk}/\sqrt{\sigma_{jj}\sigma_{kk}}$. The variance of the empirical variances is defined by

$$\widehat{\mathrm{var}}(\sigma_{jj}) = \frac{n}{(n-1)^3}\sum_{i=1}^{n}(w_{ijj}-\bar{w}_{jj})^2 \tag{3.13}$$

and $\widehat{\mathrm{var}}(\rho_{jk})$ can be estimated by applying this formula to the standardized data matrix. The optimal correlation shrinkage intensity λ_1^* and the optimal variance shrinkage level λ_2^* are estimated using an analytic formula by Schäfer and Strimmer [SS05] and Opgen-Rhein and Strimmer [ORS07], respectively. Due to these two different tuning parameters, separate shrinkage for variances and correlations is performed. Thereby, the variances are shrunk towards σ_{median}, the median value of all σ_{jj}. The corresponding partial correlations

$\rho^*(x_j, x_k|\boldsymbol{x}_{-jk})$ can again be derived from the resulting inverse covariance matrix

$$\boldsymbol{\Sigma}^{*-1} = (\sigma^*_{jk})^{-1} = (c^*_{jk}). \tag{3.14}$$

This two-way shrinkage formula for the covariance matrix estimator is implemented in the R package `corpcor` [SORS09]. In particular, regularized estimates of partial correlations and partial variances can be computed using the function `pcor.shrink()` of this package.

3.4. Solving the Penalized Least Squares Problem

For solving the penalized regression problem

$$\widehat{\boldsymbol{\beta}}^* = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\right)^{\mathrm{T}} \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\right) + P_{PFL}\left(\boldsymbol{\beta}, \lambda, \alpha\right) \tag{3.15}$$

we have to transform this problem as proposed by Petry [Pet09] and described below. The approach is similar to the idea of Zou and Hastie [ZH05] where the representation of the elastic net as a lasso type problem was used. By reparametrization, the coefficients θ_{jk} and θ_{j0} are defined by

$$\begin{aligned} \theta_{jk} &= \beta_j - \beta_k, \quad 1 \le k < j \le p, \\ \theta_{j0} &= \beta_j, \quad 1 \le j \le p. \end{aligned} \tag{3.16}$$

Thus, the restriction

$$\theta_{jk} = \theta_{j0} - \theta_{k0}, \quad 1 \le k < j \le p \tag{3.17}$$

holds. Correspondingly let $\boldsymbol{\Delta X}$ denote the $n \times \binom{p}{2}$-matrix which contains the pairwise differences of all predictors. With $\Delta_{jk} = \boldsymbol{x}_j - \boldsymbol{x}_k$, $1 \le k < j \le p$, this is given by

$$\boldsymbol{\Delta X} = \left(\Delta_{21}|\Delta_{31}|\ldots|\Delta_{p1}|\Delta_{32}|\Delta_{42}|\ldots|\Delta_{p(p-1)}\right). \tag{3.18}$$

Further, let

$$\boldsymbol{\theta} = \left(\theta_{10}, \ldots, \theta_{p0}, \theta_{21}, \ldots, \theta_{p(p-1)}\right)^{\mathrm{T}} \tag{3.19}$$

be the corresponding coefficient vector for the expanded design matrix $(\boldsymbol{X}|\boldsymbol{\Delta X})$.
Then the pairwise fused lasso penalty has the form

$$P_{PFL}\left(\boldsymbol{\theta}, \lambda, \alpha\right) = \lambda \left[\alpha \sum_{j=1}^{p} w_{j0}|\theta_{j0}| + (1-\alpha) \sum_{j=1}^{p-1} \sum_{k=j+1}^{p} w_{jk}|\theta_{jk}|\right]. \tag{3.20}$$

The redundancy (3.17) is implied by using an additional penalty term with large γ yielding

$$\begin{aligned}\widehat{\boldsymbol{\theta}}^{*}(\gamma,\lambda,\alpha) &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left(\boldsymbol{y}-(\boldsymbol{X}|\boldsymbol{\Delta X})\,\boldsymbol{\theta}\right)^{\mathrm{T}}\left(\boldsymbol{y}-(\boldsymbol{X}|\boldsymbol{\Delta X})\,\boldsymbol{\theta}\right) \\ &\quad +\gamma\sum_{j=1}^{p-1}\sum_{k=j+1}^{p}\left(\theta_{j0}-\theta_{k0}-\theta_{jk}\right)^{2} \\ &\quad +\lambda\left[\alpha\sum_{j=1}^{p} w_{j0}|\theta_{j0}|+(1-\alpha)\sum_{j=1}^{p-1}\sum_{k=j+1}^{p} w_{jk}|\theta_{jk}|\right].\end{aligned} \tag{3.21}$$

For $\gamma\to\infty$ the restriction (3.17) is fulfilled. With reparameterization (3.16) the approximate estimator (3.21) can be formulated as a lasso type problem. Hence, criterion (3.21) can be written as

$$\begin{aligned}\widehat{\boldsymbol{\theta}}^{*}(\gamma,\lambda,\alpha) &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left(\boldsymbol{y}_0-\boldsymbol{D}\boldsymbol{\theta}\right)^{\mathrm{T}}\left(\boldsymbol{y}_0-\boldsymbol{D}\boldsymbol{\theta}\right) \\ &\quad +\lambda\left[\alpha\sum_{j=1}^{p} w_{j0}|\theta_{j0}|+(1-\alpha)\sum_{j=1}^{p-1}\sum_{k=j+1}^{p} w_{jk}|\theta_{jk}|\right],\end{aligned} \tag{3.22}$$

where $\boldsymbol{y}_0=\left(\boldsymbol{y}^{\mathrm{T}},\mathbf{0}_{\tilde{p}}^{\mathrm{T}}\right)^{\mathrm{T}}$ with $\mathbf{0}_{\tilde{p}}$ denoting a zero vector of length $\tilde{p}=\binom{p}{2}$ and

$$\boldsymbol{D}=\begin{pmatrix}\boldsymbol{X}|\boldsymbol{\Delta X}\\ \sqrt{\gamma}\widetilde{\boldsymbol{C}}\end{pmatrix} \tag{3.23}$$

Matrix $\widetilde{\boldsymbol{C}}$ is the $\tilde{p}\times(\tilde{p}+p)$-matrix which includes the constraints (3.17). Let δ_{jk}, $1\le k<j\le p$, denote a p-dimensional row vector with -1 at the kth and $+1$ at the jth component and all other components equal zero. With τ_m we define a $\tilde{p}$-dimensional row vector whose m-th component is -1 and zero otherwise. Then all constraints given by (3.17) can be summarized in matrix notation:

$$\widetilde{\boldsymbol{C}}=\begin{pmatrix}\delta_{21} & \tau_1\\ \delta_{31} & \tau_2\\ \vdots & \vdots\\ \delta_{p1} & \tau_{p-1}\\ \delta_{32} & \tau_p\\ \delta_{42} & \tau_{p+1}\\ \vdots & \vdots\\ \delta_{p(p-1)} & \tau_{\tilde{p}}\end{pmatrix} \tag{3.24}$$

Now Θ is the index set of the components of $\boldsymbol{\theta}$ given by (3.19) and we transform (3.22) to

$$\begin{aligned}\widehat{\boldsymbol{\theta}}^{*} &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left(\boldsymbol{y}_0 - \boldsymbol{D}\boldsymbol{\theta}\right)^{\mathrm{T}} \left(\boldsymbol{y}_0 - \boldsymbol{D}\boldsymbol{\theta}\right) + \lambda \left(\sum_{j=1}^{p} |\alpha \cdot w_{j0} \cdot \theta_{j0}| + \sum_{j=1}^{p-1} \sum_{k=j+1}^{p} |(1-\alpha) \cdot w_{jk} \cdot \theta_{jk}| \right) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left(\boldsymbol{y}_0 - \boldsymbol{D}\boldsymbol{\theta}\right)^{\mathrm{T}} \left(\boldsymbol{y}_0 - \boldsymbol{D}\boldsymbol{\theta}\right) + \lambda \left(\sum_{t \in \Theta} |\alpha \cdot w_t \cdot \theta_t| + |(1-\alpha) \cdot w_t \cdot \theta_t| \right). \qquad (3.25)\end{aligned}$$

Consequently, (3.25) is a lasso type problem on the expanded design matrix $\boldsymbol{D}$ where the components of θ are weighted by α and $(1-\alpha)$. This parametrization demands a rescaling of the constraints matrix $\widetilde{\boldsymbol{C}}$ (3.24). In (3.25) the parameters are multiplied with α, $(1-\alpha)$ respectively. Accordingly, the matrix $\widetilde{\boldsymbol{C}}$ in the design matrix $\boldsymbol{D}$ in (3.25) is of the form

$$\boldsymbol{C} = \widetilde{\boldsymbol{C}}\operatorname{diag}(\boldsymbol{l}),\ \boldsymbol{l} = \Big(\underbrace{\alpha^{-1}, \cdots, \alpha^{-1}}_{p}, \underbrace{(1-\alpha)^{-1}, \cdots, (1-\alpha)^{-1}}_{\tilde{p}} \Big)^{\mathrm{T}}. \qquad (3.26)$$

The ith component of the estimate $\widetilde{\boldsymbol{\beta}}_{PFL,\lambda,\alpha}$ is obtained by summing up all components of $\boldsymbol{\theta} = \left(\theta_{01}, \cdots, \theta_{0p}, \theta_{21}, \cdots, \theta_{p(p-1)}\right)^{\mathrm{T}}$ whose index contains i in at least one place.

This approach to solve the penalized least squares problem for the pairwise fused lasso is implemented in the R function `GFL.base()`. In addition, the modified penalty term with maximum likelihood estimates as weights (Eq. (3.5)) and the correlation based penalty term with marginal correlations (Eq. (3.6)) are implemented in the functions `GFL.base.kq()` and `GFL.base.cor()`, respectively [Pet09].

3.5. Preliminaries

In this section we introduce the basic notation and definitions for the different pairwise fused lasso penalties. This notation will be used throughout Chapters 5 and 6:

- *pfl*:
  PFL penalty according to Equation (3.4);
- *pfl.kq*:
  PFL penalty according to Equation (3.5) with OLS estimates as weights in the case of normal distribution;
- *pfl.ml*:
  PFL penalty according to Equation (3.5) with ML estimates as weights in the case of binomial and Poisson distribution, respectively;
- *pfl.cor*:
  correlation based PFL penalty according to Equation (3.6) with marginal correlations;

- *pcor.shrink*:
  correlation based PFL penalty according to Equation (3.6) with regularized partial correlations;
- *pcor.emp*:
  correlation based PFL penalty according to Equation (3.6) with empirical partial correlations;
- *kqpcor.shrink/mlpcor.shrink*:
  equals the penalty *pcor.shrink*, but with additional weights $w_j = |\beta_j^{ML}|^{-1}$ for the lasso term in the penalty;
- *kqpcor.emp/mlpcor.emp*:
  equals the penalty *pcor.emp*, but with additional weights $w_j = |\beta_j^{ML}|^{-1}$ for the lasso term in the penalty;

4. Penalized Generalized Linear Models

4.1. Generalized Linear Models

In practice, the errors ϵ_i are often not assumed to be normally distributed as proposed in Section 2.1. Therefore, generalized linear models are formulated in order to extend the theory of classical linear models by further distributions of the response variable. According to Fahrmeir and Tutz [FT01], a generalized linear model is characterized by two assumptions: the *distributional assumption* constitutes that, given $\boldsymbol{x}_i$, the y_i are (conditionally) independent and that the (conditional) distribution of the response variable belongs to a simple exponential family, i.e. the density of y_i can be written as

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

In this context θ_i is the natural parameter, ϕ is an additional scale or dispersion parameter and $b(\cdot)$, $c(\cdot)$ are functions corresponding to the type of the exponential family. The (conditional) expectation is $E(y_i|\boldsymbol{x}_i) = \mu_i$. Note that the scale parameter ϕ does not depend on the observation i.
The relationship between the mean of the response variable μ_i and the linear predictor $\eta_i = \boldsymbol{x}_i^T\beta$ is determined by the *structural assumption*. Thus, we have:

$$\mu_i = h(\eta_i), \qquad \eta_i = g(\mu_i) = h^{-1}(\mu_i), \qquad \text{resp.} \tag{4.2}$$

where $h(\cdot)$ is the response function and $g(\cdot)$, the inverse of h, is the so-called link function. The natural parameter θ_i is a function of the expectation μ_i, i.e. $\theta_i = \theta(\mu_i)$. Furthermore, the mean is of the form $\mu_i = b'(\theta_i) = \partial b(\theta_i)/\partial\theta_i$ and $v(\mu_i) = b''(\theta_i) = \partial^2 b(\theta_i)/\partial\theta_i^2$. The variance function $v(\mu_i)$ results from $var(y_i|\mathbf{x}_i) = \phi v(\mu_i)$, the variance of the response y_i. The specification of the mean $\mu_i = h(\eta_i)$ implies a certain variance structure since both are based on derivatives of $b(\theta_i)$. If the natural parameter is directly related to the linear predictor, the link function is called *natural* or *canonical* link function and is given by

$$\theta(\mu_i) = \eta_i, \qquad g(\mu_i) = \theta(\mu_i), \qquad \text{resp.} \tag{4.3}$$

Besides the normal distribution, the binomial, the Poisson and the gamma distribution are also members of the exponential family. These distributions have the following natural link functions:

$\eta_i = \mu_i$	for the normal distribution,
$\eta_i = -1/\mu_i$	for the gamma distribution,
$\eta_i = \log \mu_i$	for the Poisson distribution,
$\eta_i = \log\left(\mu_i/(1-\mu_i)\right)$	for the Bernoulli distribution.

Further details on the distributions just mentioned are given in the following:

Models for continuous response
In the terminology of generalized linear models, the normal regression model is given by

$$y_i|\boldsymbol{x}_i \sim N(\mu_i, \sigma^2), \qquad \mu_i = \eta_i = \boldsymbol{x}_i^{\mathrm{T}}\beta. \tag{4.4}$$

Thereby, the link function is the canonical link. The components of the exponential family for this distribution are $\theta(\mu_i) = \mu_i$, $b(\theta_i) = \theta_i^2/2$ and $\phi = \sigma^2$. If the response y_i is non-negative another link function has to be chosen. This involves a non-negative mean μ_i, e.g. the log-link:

$$\mu_i = \exp(\eta_i) \Leftrightarrow \log(\mu_i) = \eta_i \tag{4.5}$$

A further distribution for continuous non-negative responses is the gamma distribution. Besides its expectation $\mu_i > 0$, the density includes the shape parameter $\nu > 0$ which causes the greater flexibility of the gamma distribution. Thus, we have $\theta(\mu_i) = -1/\mu_i$, $b(\theta_i) = -\log(-\theta_i)$ and $\phi = -1/\nu$. With $var(y_i) = \mu_i^2/\nu$, an increasing expectation implicates an increasing variance. In addition to the natural link $\eta_i = 1/\mu_i$, two other important link functions for the gamma distribution are the identity link $\eta_i = \mu_i$ and the log-link $\eta_i = \log(\mu_i)$.

Models for binary response
A binary outcome $y_i \in \{0, 1\}$ can be modeled by the Bernoulli distribution, i.e. $y_i \sim B(1, \pi_i)$. Thereby, the response probability is defined by

$$E(y_i|\boldsymbol{x}_i) = P(y_i = 1|\boldsymbol{x}_i) = \mu_i = \pi_i \tag{4.6}$$

and $\theta(\pi_i) = \log(\pi_i/(1-\pi_i))$, $b(\theta_i) = \log(1+\exp(\theta_i)) = -\log(1-\pi_i)$ and $\phi = 1$ are the components of the exponential family. This implies the variance structure $var(y_i|\boldsymbol{x}_i) = \pi_i(1-\pi_i)$. The following three models and their corresponding link functions are the most common to relate the response probability π_i to the linear predictor $\eta_i = \boldsymbol{x}_i^{\mathrm{T}}\beta$ [FT01].

1. The *logit* model is determined by the canonical link function

$$g(\pi_i) = \log\left(\frac{\pi_i}{1-\pi_i}\right) = \eta_i. \tag{4.7}$$

   Therefore, as the resulting response function we obtain the logistic distribution function

$$\pi_i = h(\eta_i) = \frac{\exp(\eta_i)}{1+\exp(\eta_i)}. \tag{4.8}$$

2. In the *probit* model, the response function conforms to the standard normal distribution function, i.e.

$$\pi_i = h(\eta_i) = \Phi(\eta_i), \qquad g(\pi_i) = \Phi^{-1}(\pi_i) = \eta_i, \qquad \text{resp.} \tag{4.9}$$

3. Furthermore, the *complementary log-log* model is based on the link function

$$g(\pi_i) = \log(-\log(1-\pi)) = \eta_i. \tag{4.10}$$

Models for count data
A common distribution for count data is the Poisson distribution with parameter $\lambda_i > 0$ and responses $y_i \in \{0, 1, 2, \ldots\}$. The parameters of the exponential family are given by $\theta(\mu_i) = \log(\mu_i)$, $b(\theta_i) = \exp(\theta_i)$ and $\phi = 1$ with $\mu_i = \lambda_i$. Due to the non-negativity of the response, a frequently used link function is again the log-link as for the normal distribution.

4.2. Maximum Likelihood Estimation

Maximum likelihood is the estimation procedure for generalized linear models. Since the responses belong to an exponential family (4.1), the log-likelihood for independent observations $y_1, \ldots, y_n$ is given by

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} l_i(\theta_i) = \sum_{i=1}^{n} \log f(y_i \mid \theta_i, \phi) = \sum_{i=1}^{n} \frac{y_i\theta_i - b(\theta_i)}{\phi}. \tag{4.11}$$

Here, function $c(y_i, \phi)$ is omitted because it does not depend on θ_i and therefore not on $\boldsymbol{\beta}$. The first derivative of the log-likelihood is the score function

$$s(\boldsymbol{\beta}) = \frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}. \tag{4.12}$$

Using the relations $\theta_i = \theta(\mu_i)$, $\mu_i = h(\eta_i)$ and $\eta_i = \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}$, the score function is given by

$$\begin{aligned} s(\boldsymbol{\beta}) = \sum_{i=1}^{n} s_i(\boldsymbol{\beta}) &= \sum_{i=1}^{n} \frac{\partial l_i(\theta_i)}{\partial \theta} \frac{\partial \theta(\mu_i)}{\partial \mu} \frac{\partial h(\eta_i)}{\partial \eta} \frac{\partial \eta_i}{\partial \boldsymbol{\beta}} \\ &= \sum_{i=1}^{n} \boldsymbol{x}_i D_i \sigma_i^{-2} (y_i - \mu_i) \end{aligned} \tag{4.13}$$

where D_i is the first derivative of the response function $h(\eta_i)$ evaluated at $\eta_i = \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}$, i.e. $D_i = \partial h(\eta_i)/\partial \eta$, and $\sigma_i^2 = \phi v(h(\eta_i))$. The *observed information matrix* is

$$F_{obs}(\boldsymbol{\beta}) = -\frac{\partial^2 l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^{\mathrm{T}}} = \left(-\frac{\partial^2 l(\boldsymbol{\beta})}{\partial \beta_i \partial \beta_j} \right)_{ij}, \tag{4.14}$$

i.e. the matrix of negative second derivatives which depends on the observations. Thus, the *expected information matrix* or *Fisher matrix* is given by

$$\begin{aligned} F(\boldsymbol{\beta}) &= E(F_{obs}(\boldsymbol{\beta})) = \sum_{i=1}^{n} F_i(\boldsymbol{\beta}) \\ &= \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}} D_i^2 \sigma_i^{-2}. \end{aligned} \tag{4.15}$$

For the canonical link function (4.3), the expected and the observed information matrix are identical:

$$F(\boldsymbol{\beta}) = F_{obs}(\boldsymbol{\beta}). \tag{4.16}$$

In matrix notation, the score function and the Fisher matrix have the following form:

$$s(\boldsymbol{\beta}) = \boldsymbol{X}^{\mathrm{T}} \boldsymbol{D} \boldsymbol{\Sigma}^{-1} (\boldsymbol{y} - \boldsymbol{\mu}) \tag{4.17}$$

and

$$F(\boldsymbol{\beta}) = \boldsymbol{X}^{\mathrm{T}} \boldsymbol{D} \boldsymbol{\Sigma}^{-1} \boldsymbol{D}^{\mathrm{T}} \boldsymbol{X} = \boldsymbol{X}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{X}, \tag{4.18}$$

respectively. Thereby, $\boldsymbol{D} = \mathrm{diag}(\partial h(\eta_1)/\partial\eta, \ldots, \partial h(\eta_n)/\partial\eta)$ denotes the diagonal matrix of derivatives, $\boldsymbol{\Sigma} = (\sigma_1^2, \ldots, \sigma_n^2)$ is the covariance matrix and $\boldsymbol{W} = \boldsymbol{D}\boldsymbol{\Sigma}^{-1}\boldsymbol{D}^{\mathrm{T}}$ is the diagonal weight matrix.
The matrix notation is useful for the computation of the maximum likelihood solution. The maximum likelihood estimates are obtained by solving the equation

$$s(\widehat{\boldsymbol{\beta}}) = 0. \tag{4.19}$$

According to Fahrmeir and Tutz [FT01], Equation (4.19) is in general non-linear and thus has to be solved iteratively. A widely used iteration procedure for this is *Fisher scoring*. Starting with an initial estimate $\widehat{\boldsymbol{\beta}}^{(0)}$, Fisher scoring iterations are defined by

$$\widehat{\boldsymbol{\beta}}^{(k+1)} = \widehat{\boldsymbol{\beta}}^{(k)} + F^{-1}(\widehat{\boldsymbol{\beta}}^{(k)}) s(\widehat{\boldsymbol{\beta}}^{(k)}), \quad k = 0, 1, 2, \ldots \tag{4.20}$$

where $\widehat{\boldsymbol{\beta}}^{(k)}$ denotes the estimate in the kth step. Iteration is stopped if some termination criterion is reached, e.g.

$$\frac{\|\widehat{\boldsymbol{\beta}}^{(k+1)} - \widehat{\boldsymbol{\beta}}^{(k)}\|}{\|\widehat{\boldsymbol{\beta}}^{(k)}\|} \leq \epsilon. \tag{4.21}$$

This means that the iteration is stopped if the changes between successive steps are smaller than the specified threshold ϵ.
The Fisher scoring iterations in Equation (4.20) can be alternatively seen as iteratively weighted least squares. Therefore, we define working observations as

$$\tilde{y}_i(\widehat{\boldsymbol{\beta}}) = \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\beta} + \boldsymbol{D}_i^{-1} (y_i - \mu_i) \tag{4.22}$$

and the corresponding vector of working observations $\tilde{\boldsymbol{y}}(\widehat{\boldsymbol{\beta}}) = (\tilde{y}_1(\widehat{\boldsymbol{\beta}}), \ldots, \tilde{y}_n(\widehat{\boldsymbol{\beta}}))^{\mathrm{T}}$. Then, the Fisher scoring iterations can be written as

$$\widehat{\boldsymbol{\beta}}^{(k+1)} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}(\widehat{\boldsymbol{\beta}}^{(k)})\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}(\widehat{\boldsymbol{\beta}}^{(k)})\tilde{\boldsymbol{y}}(\widehat{\boldsymbol{\beta}}^{(k)}). \tag{4.23}$$

Another iterative procedure to solve Equation (4.19) is the *Newton Raphson method*. Thereby, the expected information matrix $F(\boldsymbol{\beta})$ is replaced by the observed information matrix $F_{obs}(\boldsymbol{\beta})$ in the Fisher scoring iteration in Equation (4.20).

4.3. Local Quadratic Approximations

In this section we describe the *local quadratic approximation approach*, introduced by Ulbricht [Ulb10b]. This algorithm fits penalized generalized linear models and thereby comprises a large class of penalties.
In the following, $\boldsymbol{b} = (\beta_0, \boldsymbol{\beta}^{\mathrm{T}})^{\mathrm{T}}$ denotes the vector of unknown parameters in the predictor. In addition to the coefficients of the p regressors, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^{\mathrm{T}}$, the vector $\boldsymbol{b}$ contains an intercept β_0. We want to solve the penalized minimization problem

$$\min_{\boldsymbol{b}} -l(\boldsymbol{b}) + P_\delta(\lambda, \boldsymbol{\beta}), \tag{4.24}$$

where $l(\boldsymbol{b})$ is the log-likelihood of the underlying generalized linear model. The penalty term is the sum of J penalty functions and is given by

$$P_\delta(\lambda, \boldsymbol{\beta}) = \sum_{j=1}^{J} p_{\lambda,j}(|\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}|). \tag{4.25}$$

Thereby, $\boldsymbol{a}_j$ is a known vector of constants and $p_{\lambda,j} : \mathbb{R}_+^* \to \mathbb{R}_+^*$. The subscript δ represents the specific penalty family, e.g. $P_{PFL}(\lambda, \boldsymbol{\beta})$ denotes the pairwise fused lasso penalty in Equation (3.3). The penalty proposed by Fan and Li [FL01] is of the structure $P_\delta(\lambda, \boldsymbol{\beta}) = \sum_{j=1}^{p} p_\lambda(|\beta_j|)$. Since vector $\boldsymbol{a}_j$ has only one non-zero element in this case, this penalty term does not take into account any interactions between the regression coefficients. Hence, the approach of Fan and Li [FL01] can be only applied to few penalty families such as ridge and lasso, but not to the fused lasso or pairwise fused lasso.
The sum of all J penalty functions $p_{\lambda,j}(|\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}|)$ in Equation (4.25) determines the penalty region, whereas the number J of penalty functions does not have to conform to the number of regressors p. Furthermore, the type of the penalty function and the tuning parameter λ do not have to be the same for all J penalty functions. For illustration purposes, we consider the pairwise fused lasso penalty

$$P_{PFL}(\lambda_1, \lambda_2, \boldsymbol{\beta}) = \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=2}^{p} \sum_{k=1}^{j-1} |\beta_j - \beta_k|.$$

According to the notation of Equation (4.25), the pairwise fused lasso penalty can be described by

$$P_{PFL}(\lambda_1, \lambda_2, \boldsymbol{\beta}) = \sum_{j=1}^{\tilde{p}+p} p_{\lambda,j}(|\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}|), \tag{4.26}$$

with $\tilde{p} = \binom{p}{2}$. The p penalty functions for the lasso penalty term are given by

$$p_{\lambda,j}(\cdot) = \lambda_1 |\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}|, \quad j = 1, \ldots, p, \tag{4.27}$$

where $\boldsymbol{a}_j = (0, \ldots, 0, 1, 0, \ldots, 0)^{\mathrm{T}}$ with a one at the jth position. We obtain for the $\tilde{p}$ penalty functions for the difference penalty term:

$$p_{\lambda,j}(\cdot) = \lambda_2 |\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}|, \quad j = p+1, \ldots, \tilde{p}+p \tag{4.28}$$

with the p-dimensional vector $\boldsymbol{a}_j = (0, \ldots, -1, 0, \ldots, 1, 0, \ldots, 0)$ with a one at the kth position for $k = p+2, \ldots, \tilde{p}+p$, and a minus one at the lth position for $l = p+1, \ldots, k-1$.
An often applied principle in solving a convex optimization problem is to use a quadratic approximation of the objective function. If the latter is twice continuously differentiable iterative procedures of the Newton type are appropriate (Sec. 4.2). Therefore, we need the gradient and the Hessian of the objective function. Since the first term of Equation (4.24) is the negative log-likelihood, we can use the corresponding score function and expected Fisher information matrix. For the second term, we cannot proceed the same way because it includes L_1-norm terms. Therefore, Ulbricht [Ulb10b] developed a quadratic approximation of the penalty term (4.25) which is described in the following. Based on this approximation, Newton-type algorithms (Sec. 4.2) can be applied.
Let $\xi_j = |\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}|$ and

$$p'_{\lambda,j} = \frac{dp_{\lambda,j}}{d\xi_j}. \tag{4.29}$$

We assume that $p_{\lambda,j}$ is continuously differentiable for all $\xi_j > 0$. Due to this assumption we set

$$p'_{\lambda,j} \equiv \lim_{\xi_j \downarrow 0} p'_{\lambda,j}(\xi_j). \tag{4.30}$$

Using Equation (4.30), for $\xi_j \geq 0$ we obtain the gradient of the jth penalty function

$$\nabla p_{\lambda,j} = \frac{\partial p_{\lambda,j}}{\partial \boldsymbol{\beta}} = p'_{\lambda,j}(\xi_j)\mathrm{sgn}(\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta})\boldsymbol{a}_j, \tag{4.31}$$

which implies that

$$\mathrm{sgn}(\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}) = \frac{\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}}{|\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}|}. \tag{4.32}$$

If the update $\boldsymbol{\beta}_{(k)}$ of the Newton-type algorithm is close to $\boldsymbol{\beta}$, we can use the approximation

$$\mathrm{sgn}(\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}_{(k)}) \approx \frac{\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}}{|\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}_{(k)}|}. \tag{4.33}$$

However a drawback of this approximation is that it is restricted to $|\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}_{(k)}| \neq 0$. Therefore, if $|\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}_{(k)}|$ appears in the denominator, we use the approximation $|\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}_{(k)}| \approx \zeta_j$. The parameter ζ_j is given by

$$\zeta_j = \sqrt{\left(\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}_{(k)}\right)^2 + c}, \tag{4.34}$$

where c is a small positive integer (for the computations in chapters 5 and 6 we choose $c = 10^{-8}$). Thus, we can overcome the restriction in Equation (4.33) and improve on the numerical stability.
Next, we consider the following equation:

$$\begin{aligned}
\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}\boldsymbol{a}_j^{\mathrm{T}}\left(\boldsymbol{\beta}-\boldsymbol{\beta}_{(k)}\right) &= \left(\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}\right)^2 - \boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}_{(k)} \\
&= \frac{1}{2}\left[\left(\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}\right)^2 - 2\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}_{(k)} + \left(\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}_{(k)}\right)^2\right] + \frac{1}{2}\left[\left(\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}\right)^2 - \left(\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}_{(k)}\right)^2\right] \\
&= \frac{1}{2}\left[\boldsymbol{a}_j^{\mathrm{T}}\left(\boldsymbol{\beta}-\boldsymbol{\beta}_{(k)}\right)\right]^2 + \frac{1}{2}\left(\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{a}_j\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta} - \boldsymbol{\beta}_{(k)}^{\mathrm{T}}\boldsymbol{a}_j\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}_{(k)}\right)
\end{aligned} \tag{4.35}$$

If $\boldsymbol{\beta}_{(k)}$ is close to $\boldsymbol{\beta}$, the first term of the right hand side of Equation (4.35) is nearly zero. Consequently, we obtain the approximation

$$\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}\boldsymbol{a}_j^{\mathrm{T}}\left(\boldsymbol{\beta}-\boldsymbol{\beta}_{(k)}\right) \approx \frac{1}{2}\left(\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{a}_j\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta} - \boldsymbol{\beta}_{(k)}^{\mathrm{T}}\boldsymbol{a}_j\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}_{(k)}\right). \tag{4.36}$$

Using the approximations in Equations (4.33) and (4.36), the first order Taylor expansion of the jth penalty function in the neighborhood of $\boldsymbol{\beta}_{(k)}$ can be written as

$$\begin{aligned}
p_{\lambda,j}\left(|\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}|\right) &\approx p_{\lambda,j}\left(|\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}_{(k)}|\right) + \nabla p_{\lambda,j}^{\mathrm{T}}\left(\boldsymbol{\beta}-\boldsymbol{\beta}_{(k)}\right) \\
&\approx p_{\lambda,j}\left(|\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}_{(k)}|\right) + p'_{\lambda,j}\left(|\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}_{(k)}|\right)\frac{\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}}{\zeta_j}\boldsymbol{a}_j^{\mathrm{T}}\left(\boldsymbol{\beta}-\boldsymbol{\beta}_{(k)}\right) \\
&\approx p_{\lambda,j}\left(|\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}_{(k)}|\right) + \frac{1}{2}\frac{p'_{\lambda,j}\left(|\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}_{(k)}|\right)}{\zeta_j}\left(\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{a}_j\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta} - \boldsymbol{\beta}_{(k)}^{\mathrm{T}}\boldsymbol{a}_j\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}_{(k)}\right)
\end{aligned} \tag{4.37}$$

Approximation (4.37) is a quadratic function of $\boldsymbol{\beta}$. Using matrix notation and summation over all J penalty functions it is equivalent to

$$\sum_{j=1}^{J} p_{\lambda,j}\left(|\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}|\right) \approx \sum_{j=1}^{J} p_{\lambda,j}\left(|\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}_{(k)}|\right) + \frac{1}{2}\left(\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{A}_\lambda\boldsymbol{\beta} - \boldsymbol{\beta}_{(k)}^{\mathrm{T}}\boldsymbol{A}_\lambda\boldsymbol{\beta}_{(k)}\right), \tag{4.38}$$

with

$$\boldsymbol{A}_\lambda = \sum_{j=1}^{J}\frac{p'_{\lambda,j}\left(|\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{\beta}_{(k)}|\right)}{\zeta_j}\boldsymbol{a}_j\boldsymbol{a}_j^{\mathrm{T}} \tag{4.39}$$

which does not depend on the parameter vector β. Since an intercept is included in the model, the penalty matrix is extended to

$$\boldsymbol{A}_\lambda^* = \begin{bmatrix} 0 & \mathbf{0}^\mathrm{T} \\ \mathbf{0} & \boldsymbol{A}_\lambda \end{bmatrix}, \tag{4.40}$$

where $\mathbf{0}$ is the p-dimensional zero vector. The quadratic approximation of the penalty term (4.38) and a second order Taylor expansion of the negative log-likelihood at $\boldsymbol{b}_{(k)} = (\beta_{0,(k)}, \boldsymbol{\beta}_{(k)}^\mathrm{T})^\mathrm{T}$ result in a local approximation of the penalized minimization problem in Equation (4.24), which is defined by

$$\begin{aligned} Q(\boldsymbol{b}) \;:=\; & -l\left(\boldsymbol{b}_{(k)}\right) - s\left(\boldsymbol{b}_{(k)}\right)^\mathrm{T}\left(\boldsymbol{b}-\boldsymbol{b}_{(k)}\right) - \frac{1}{2}\left(\boldsymbol{b}-\boldsymbol{b}_{(k)}\right)^\mathrm{T}\boldsymbol{H}\left(\boldsymbol{b}_{(k)}\right)\left(\boldsymbol{b}-\boldsymbol{b}_{(k)}\right) \\ & + \sum_{j=1}^{J} p_{\lambda,j}\left(\left|\boldsymbol{a}_j^\mathrm{T}\boldsymbol{\beta}_{(k)}\right|\right) + \frac{1}{2}\left(\boldsymbol{b}^\mathrm{T}\boldsymbol{A}_\lambda^*\boldsymbol{b} + \boldsymbol{b}_{(k)}^\mathrm{T}\boldsymbol{A}_\lambda^*\boldsymbol{b}_{(k)}\right). \end{aligned} \tag{4.41}$$

Here, $\boldsymbol{s}$ and $\boldsymbol{H}$ denote the score function and the Hessian of the log-likelihood, respectively. To apply a Newton-type algorithm to this local quadratic minimization problem, we have to compute the gradient and the Hessian of Equation (4.41).
The gradient is

$$\frac{\partial Q}{\partial \boldsymbol{b}} = -\boldsymbol{s}\left(\boldsymbol{b}_{(k)}\right) - \boldsymbol{H}\left(\boldsymbol{b}-\boldsymbol{b}_{(k)}\right) + \boldsymbol{A}_\lambda^*\boldsymbol{b} \tag{4.42}$$

and for $\boldsymbol{b} = \boldsymbol{b}_{(k)}$ it evaluates to

$$\left.\frac{\partial Q}{\partial \boldsymbol{b}}\right|_{\boldsymbol{b}=\boldsymbol{b}_{(k)}} = -\boldsymbol{s}\left(\boldsymbol{b}_{(k)}\right) + \boldsymbol{A}_\lambda^*\boldsymbol{b}. \tag{4.43}$$

The corresponding Hessian is given by

$$\frac{\partial^2 Q}{\partial \boldsymbol{b}\partial \boldsymbol{b}^\mathrm{T}} = \left.\frac{\partial^2 Q}{\partial \boldsymbol{b}\partial \boldsymbol{b}^\mathrm{T}}\right|_{\boldsymbol{b}=\boldsymbol{b}_{(k)}} = -\boldsymbol{H}\left(\boldsymbol{b}_{(k)}\right) + \boldsymbol{A}_\lambda^*. \tag{4.44}$$

In order that it is not necessary to compute the second order derivative in each iteration, we use $-E\left(\boldsymbol{H}(\boldsymbol{b})\right) = \boldsymbol{F}(\boldsymbol{b})$ as in the Fisher-scoring algorithm (q.v. Sec. 4.2). Thereby, $\boldsymbol{F}$ denotes the Fisher information matrix corresponding to the log-likelihood $l(\boldsymbol{b})$. Then, starting with the initial value $\boldsymbol{b}_{(0)}$ and extending Equation (4.44) by the Fisher information $\boldsymbol{F}(\boldsymbol{b}_{(k)})$, the update step of the quasi-Newton method is

$$\boldsymbol{b}_{(k+1)} = \boldsymbol{b}_{(k)} - \left(\boldsymbol{F}(\boldsymbol{b}_{(k)}) + \boldsymbol{A}_\lambda^*\right)^{-1} - \left\{\boldsymbol{s}(\boldsymbol{b}_{(k)}) + \boldsymbol{A}_\lambda^*\boldsymbol{b}_{(k)}\right\}. \tag{4.45}$$

Ulbricht [Ulb10b] calls this method "quasi"-Newton because an approximation of the Hessian is used.
Iterations are carried out until the relative distance moved during the kth step is less or equal

to a specified threshold ϵ, i.e. the termination condition is

$$\frac{\|\boldsymbol{b}_{(k+1)} - \boldsymbol{b}_{(k)}\|}{\|\boldsymbol{b}_{(k)}\|} \leq \epsilon, \;\; \epsilon > 0. \tag{4.46}$$

Since the approximation of the penalty term is a quadratic function, the update in Equation (4.45) is equivalent to the update of a quadratically penalized generalized linear model estimation problem. Accordingly, the update can be rewritten as a penalized iteratively re-weighted least squares (P-IRLS) problem

$$\boldsymbol{b}_{(k+1)} = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{X} + \boldsymbol{A}_{\lambda}^{*}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\tilde{\boldsymbol{y}}. \tag{4.47}$$

With the following derivation, Ulbricht [Ulb10b] shows that $\tilde{\boldsymbol{y}} = \boldsymbol{D}^{-1}(\boldsymbol{y} - \mu) + \boldsymbol{X}\boldsymbol{b}_{(k)}$ holds:

$$\begin{aligned}
\boldsymbol{b}_{(k+1)} &= \boldsymbol{b}_{(k)} - \left(\boldsymbol{F}(\boldsymbol{b}_{(k)}) + \boldsymbol{A}_{\lambda}^{*}\right)^{-1}\left\{-\boldsymbol{s}(\boldsymbol{b}_{(k)}) + \boldsymbol{A}_{\lambda}^{*}\boldsymbol{b}_{(k)}\right\} \\
&= \boldsymbol{b}_{(k)} - \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{X} + \boldsymbol{A}_{\lambda}^{*}\right)^{-1}\left\{-\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\left[\underbrace{\boldsymbol{D}^{-1}(\boldsymbol{y} - \mu) + \boldsymbol{X}\boldsymbol{b}_{(k)}}_{=:\tilde{\boldsymbol{y}}} - \boldsymbol{X}\boldsymbol{b}_{(k)}\right] + \boldsymbol{A}_{\lambda}^{*}\boldsymbol{b}_{(k)}\right\} \\
&= \boldsymbol{b}_{(k)} - \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{X} + \boldsymbol{A}_{\lambda}^{*}\right)^{-1}\left\{-\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\tilde{\boldsymbol{y}} + \left[\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{X} + \boldsymbol{A}_{\lambda}^{*}\right]\boldsymbol{b}_{(k)}\right\} \\
&= \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{X} + \boldsymbol{A}_{\lambda}^{*}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\tilde{\boldsymbol{y}}
\end{aligned} \tag{4.48}$$

The update in Equation (4.47) is iterated until convergence. Hence, at convergence the estimate is of the form $\hat{\boldsymbol{b}} = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{X} + \boldsymbol{A}_{\lambda}^{*}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\tilde{\boldsymbol{y}}$. Thus, $\hat{\boldsymbol{b}}$ is a weighted quadratically penalized least squares solution. With $\boldsymbol{X}^{*} := \boldsymbol{W}^{\mathrm{T}/2}\boldsymbol{X}$ and $\tilde{\boldsymbol{y}}^{*} := \boldsymbol{W}^{\mathrm{T}/2}\tilde{\boldsymbol{y}}$, the estimate can be written as

$$\hat{\boldsymbol{b}} = \left(\boldsymbol{X}^{*\mathrm{T}}\boldsymbol{X}^{*} + \boldsymbol{A}_{\lambda}^{*}\right)^{-1}\boldsymbol{X}^{*\mathrm{T}}\tilde{\boldsymbol{y}}^{*} \tag{4.49}$$

which is a quadratically penalized solution to the linear model $\tilde{\boldsymbol{y}}^{*} = \boldsymbol{X}^{*}\boldsymbol{b} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim N_n(\boldsymbol{0}, \sigma^2\boldsymbol{I})$. An alternative formulation for the solution is

$$S(\boldsymbol{b}) = \min_{\boldsymbol{b}}\|\tilde{\boldsymbol{y}}^{*} - \boldsymbol{X}^{*}\boldsymbol{b}\|^2 + \frac{1}{2}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{A}_{\lambda}^{*}\boldsymbol{b} \tag{4.50}$$

for a given value of λ. Using Equation (4.49) we obtain

$$\hat{\tilde{\boldsymbol{y}}}^{*} = \boldsymbol{X}^{*}\left(\boldsymbol{X}^{*\mathrm{T}}\boldsymbol{X}^{*} + \boldsymbol{A}_{\lambda}^{*}\right)^{-1}\boldsymbol{X}^{*\mathrm{T}}\tilde{\boldsymbol{y}}^{*} \tag{4.51}$$

as an estimate of $\tilde{\boldsymbol{y}}^{*}$.

5. Simulation Study I

5.1. Simulation Settings

In this section we present some simulation settings to investigate the performance of the pairwise fused lasso. Furthermore we compare this new method with already established ones. All simulations in this chapter are based on the linear model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}_{true} + \boldsymbol{\epsilon},\ \boldsymbol{\epsilon} \sim N_n(\boldsymbol{0}, \sigma^2 \boldsymbol{I}). \tag{5.1}$$

50 replications are performed for every simulation scenario and in each replication we generate a training, a validation and a test data set. The observation numbers of the corresponding data sets are denoted by $n_{train}/n_{vali}/n_{test}$.

Since we investigate a regularization method with both variable selection and grouping property, we use similar simulation scenarios as Zou and Hastie [ZH05].

1. The first setting is specified by the parameter vector $\boldsymbol{\beta}_{true} = (3, 1.5, 0, 0, 0, 2, 0, 0)^{\mathrm{T}}$ and standard error $\sigma = 3$. The correlation between the i-th and the j-th predictor follows from
$$\mathrm{corr}(i, j) = 0.5^{|i-j|},\ \forall i, j \in \{1, \ldots, 8\}. \tag{5.2}$$
The observation numbers are 20/20/200.

2. In this simulation scenario the parameter vector is $\boldsymbol{\beta}_{true} = (0.85, 0.85, 0.85, 0.85, 0.85, 0.85, 0.85, 0.85)^{\mathrm{T}}$ and the correlation is given by Equation (5.2). The observation numbers are 20/20/200 again.

3. In this setting we have $p = 20$ predictors. The parameter vector is structured into blocks:
$$\boldsymbol{\beta}_{true} = \big(\underbrace{0, \ldots, 0}_{5}, \underbrace{2, \ldots, 2}_{5}, \underbrace{0, \ldots, 0}_{5}, \underbrace{2, \ldots, 2}_{5}\big)^{\mathrm{T}}.$$
The standard error σ is 15 and the correlation between two predictors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is given by $\mathrm{corr}(i, j) = 0.5$. The observation numbers are 50/50/400.

4. This setting consists of $p = 20$ predictors. The parameter vector is given by
$$\boldsymbol{\beta}_{true} = \big(\underbrace{3, \ldots, 3}_{9}, \underbrace{0, \ldots, 0}_{11}\big)^{\mathrm{T}}$$
and $\sigma = 15$. The design matrix $\boldsymbol{X}$ is specified by the following procedure. First we generate three auxiliary predictors $Z_j \sim N_n(\boldsymbol{0}, \boldsymbol{I})$, $j \in \{1, 2, 3\}$. With these predictors

we generate

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

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

5. This setting is the same as Setting 4 except for the parameter vector which is given by:

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

The first three settings are also performed for the following correlation structure: the correlation between two predictors x_i and x_j in the first and the second scenario is given by

$$\mathrm{corr}(i,j) = 0.9^{|i-j|},\ \forall i,j \in \{1,\ldots,8\}, \tag{5.3}$$

whereas the third setting is realized for the correlation $\mathrm{corr}(i,j) = 0.9$.

5.2. Technical Issues

In the following, we present some technical issues for the computational approach for fitting penalized regression models. The computations in this work are realized with the statistical software package R [R D09]. The local quadratic approximation approach as described in Section 4.3 is implemented in the package `lqa` [Ulb10a]. Predominantly we used the function `cv.lqa()` of this package to fit for each simulation setting a set of models for different tuning parameter candidates and different penalties. The procedure measuring the model performance is specified in more details on the basis of the function `cv.lqa()` and its arguments.

- The function argument `family` identifies the exponential family of the response and the link function of the model. In this simulation study, a normal distribution and its canonical link are chosen (cf. Sec. 4.1).
- We investigate the pairwise fused lasso penalty and all modifications of this penalty term as described in Section 3.5. Furthermore, already established regularization methods like ridge regression, the lasso and the elastic net are considered. These different penalty terms are specified by the argument `penalty.family`.
- The tuning parameter candidates are given via the argument `lambda.candidates`. For this purpose, we generate a sequence s of equidistant points in $[-11,11]$. For the tuning parameter λ we generate a sequence $\exp(s)$ of length 100 for the distinct

pairwise fused lasso penalties as well as for the ridge, lasso and elastic net penalty in all simulation settings. In contrast, the length of the sequence for the tuning parameter α (which can only take values from 0 to 1) differs depending on the particular setting. In the first and the second simulation setting with only 8 regressors, we have a sequence of length 101. In the other three settings (each with 20 predictors) we choose only a sequence of length 11 to keep the computation time within bounds.

- The training data set is used for model fitting. Thus, we fit on these training data a penalized linear model for each tuning parameter candidate or each possible combination of tuning parameters if the penalty term is specified by two parameters. Then, the validation data are used for evaluating the performance of this set of models according to a specific loss function (argument `loss.func`). In this study, we choose the squared loss as loss function, i.e. $PE_{y,vali} = \frac{1}{n_{vali}} \|\boldsymbol{y}_{vali} - \boldsymbol{X}_{vali}\widehat{\boldsymbol{\beta}}\|^2$. We determine the model with the parameter vector $\widehat{\boldsymbol{\beta}}_{opt}$ which minimizes the squared loss $PE_{y,vali}$.
- Finally, we measure the performance of the selected model on the test data set. We compute the prediction error of $\widehat{\boldsymbol{\beta}}_{opt}$ on the test data set, $PE_{y,test} = \frac{1}{n_{test}} \|\boldsymbol{y}_{test} - \boldsymbol{X}_{test}\widehat{\boldsymbol{\beta}}_{opt}\|^2$. Furthermore, we determine the mean squared error of this coefficient vector, $MSE_\beta = \|\widehat{\boldsymbol{\beta}}_{opt} - \boldsymbol{\beta}_{true}\|^2$.

5.3. Results Setting 1

In this section, the results for the first simulation setting are presented. Before we go into detail on these results, we comment on the subsequent tables which contain the medians of the measured values.

Since 50 replications are performed for each simulation scenario, we can calculate the standard deviation of the medians by bootstrapping. According to Efron and Tibshirani [ET98], we therefore select $B = 500$ independent samples $\boldsymbol{x}^{*1}, \boldsymbol{x}^{*2}, \ldots, \boldsymbol{x}^{*B}$, each consisting of $n = 50$ data values sampled with replacement from $\boldsymbol{x}$, which denotes the corresponding vector of estimates for the 50 replications. We evaluate the median corresponding to each bootstrap sample, i.e. $\hat{\theta}^*(b) = \text{median}(\boldsymbol{x}^{*b}),\ b = 1, 2, \ldots, B$. Then, the standard error can be estimated by the sample standard deviation of the B replications

$$\widehat{\text{se}}_B = \left\{ \sum_{b=1}^{B} \left[\hat{\theta}^*(b) - \hat{\theta}^*(\cdot)\right]^2 / (B-1) \right\}^{1/2}, \tag{5.4}$$

where $\hat{\theta}^*(\cdot) = \sum_{b=1}^{B} \hat{\theta}^*(b)/B$. This algorithm is implemented in the function `bootstrap` of the same-named package [Tib09]. In the following tables, the resulting estimates of the standard deviation are quoted in parentheses.

In addition to the prediction error and the mean squared error, the *hits* and the *false positives* [TU09] are calculated. Since a model should include all influential variables, these criteria measure the performance of procedures to identify the relevant variables.

Thereby, *hits* denotes the number of correctly identified influential variables. The number of non-influential predictors which are wrongly specified as influential are given by the *false positives*. Since we are using approximation procedures, a threshold ϵ_1 is specified to distinguish influential and non-influential predictors. This threshold is set to 0.0001 in all simulation settings. This means that all those variables whose absolute value of the corresponding coefficient is greater than ϵ_1 are considered influential.

To evaluate the grouping property of regularization methods, we compute the *effective degrees of freedom*. For this purpose, we consider the estimated standardized regression coefficients. Standardized coefficients whose absolute values differ from each other by at most ϵ_2 are considered to be one cluster. The approach to investigate the absolute values of the coefficients is motivated by the definition of the grouping property (c.f. Sec. 2.2). Then, the number of *effective degrees of freedom* corresponds to the number of influential clusters which are represented by those coefficients whose absolute values are again greater than the specified threshold ϵ_2. The choice of ϵ_2 can be made dependent on the true parameter vectors $\boldsymbol{\beta}_{true}$ of the particular settings. In this study, we choose $\epsilon_2 = 0.001$ for all simulation scenarios because all the true regression coefficients in the given settings are greater or equal 0.85. Consequently, in settings with smaller coefficients, a smaller threshold ϵ_2 for clustering and identification of the effective degrees of freedom has to be chosen. A problem with clustering is that a coefficient possibly cannot be clearly assigned to a cluster. This is the case for example, if the absolute value of the difference between $|\beta_1|$ and $|\beta_2|$ corresponds to the absolute value of the difference between $|\beta_2|$ and $|\beta_3|$, but the absolute value of the difference of $|\beta_1|$ and $|\beta_3|$ exceeds the threshold ϵ_2. Then, the question would be whether β_1 and β_2, β_2 and β_3 or all three coefficients form a cluster. In this work, we do not consider this problem but assign the concerned coefficients to any cluster and determine the number of effective degrees of freedom.

The simulation results for the first setting with correlation $\rho = 0.5$ are given in Tables 5.1, 5.2 and Figures 5.1, 5.2. The boxplots of the predictors for the different pairwise fused lasso (PFL) penalties as well as for ridge regression, the lasso and the elastic net are shown in Figure 5.2. The corresponding medians of the predictors are presented in Table 5.2. Figure 5.1 illustrates both the boxplots of the prediction error (PE) on the test data set and the boxplots of the mean squared error (MSE) of the parameter vector $\boldsymbol{\beta}$. Furthermore, Table 5.1 contains the medians of the prediction error, the mean squared error, the hits and the false positives and the effective degrees of freedom.

Initially, we consider the first ten regularization methods in Table 5.1. For these methods we use the function `cv.lqa()` (Sec. 5.2). As Figure 5.2 and Table 5.2 illustrate, all ten procedures identify the relevant variables β_1, β_2 and β_6. Accordingly, the median of the hits in Table 5.1 equals 3 for each penalty. The PFL methods using partial correlations, the elastic net and the lasso estimate 4 of 5 non-influential predictors influential according to the threshold $\epsilon_1 = 0.0001$, the other methods identify all of them wrong. Except ridge regression, the effective degrees of freedom of all procedures have values between 4 and 7. Since ridge regression shows neither the variable selection property nor the grouping property, ridge regression has 8 effective degrees of freedom. If we consider the prediction error

Method	Median						
	PE		MSE		hits	false.pos	df.eff
pfl	12.10	(0.39)	4.09	(0.36)	3	5	6
pfl.kq	12.19	(0.45)	4.33	(0.57)	3	5	4
pfl.cor	12.18	(0.53)	4.01	(0.45)	3	5	6
pcor.shrink	12.26	(0.38)	3.82	(0.34)	3	4	6
pcor.emp	12.16	(0.45)	3.70	(0.38)	3	4	6
kqpcor.shrink	12.10	(0.44)	3.60	(0.37)	3	4	5.5
kqpcor.emp	**11.99**	(0.44)	3.84	(0.41)	3	4	6
enet	12.35	(0.44)	**3.52**	(0.29)	3	4	7
ridge	12.74	(0.67)	4.12	(0.65)	3	5	8
lasso	12.87	(0.50)	3.98	(0.44)	3	4	6
pfl(lars)	12.01	(0.48)	4.40	(0.42)	3	5	5
pfl.kq(lars)	**11.82**	(0.32)	3.75	(0.40)	3	4	4
pfl.cor(lars)	12.27	(0.44)	**3.72**	(0.41)	3	4	5
kq*	17.47	(1.13)	10.39	(1.39)	3	5	8
enet*	12.57	(0.52)	**3.83**	(0.53)	3	3	6
ridge*	12.63	(0.64)	4.03	(0.59)	3	5	8
lasso*	**12.53**	(0.56)	4.10	(0.57)	3	3	5

Table 5.1.: Results for the 1st simulation setting and correlation $\rho = 0.5$, based on 50 replications.

and the mean squared error (Fig. 5.1), the performance does not strongly differ between these procedures. Considering the medians in Table 5.1, kqpcor.emp has the best prediction, followed by kqpcor.shrink and pfl. With respect to the accuracy of the parameter estimate, the elastic net dominates kqpcor.shrink and pcor.emp.

One aim of this thesis is to compare the pairwise fused lasso solutions based on the local quadratic approximation approach (LQA) and the solutions based on the LARS algorithm. Thus, additionally we compute pfl, pfl.kq and pfl.cor with the functions `GFL.base()`, `GFL.base.kq()` and `GFL.base.cor()`, respectively (Sec. 3.4). In Table 5.1, these procedures are denoted by pfl(lars), pfl.kq(lars) and pfl.cor(lars). Comparing the medians of the measured values, the two approaches lead to similar but not identical results. One reason for this could be that the LARS algorithm has the property to set some coefficients exactly to zero whereas the LQA approach yields solutions which are near but not exact zero. On the other hand, the procedures work with different approximations and thus have similar but not identical solutions.

Furthermore, we compute ridge regression, the lasso and the elastic net with already established functions and packages. Ridge regression (ridge*) is implemented in function `lm.ridge()` [VR02]. For the computation of lasso estimates (lasso*) and elastic net estimates (enet*), the packages `lasso2` [LVTM09] and `elasticnet` [ZH08] can be used. With respect to the prediction error and the mean squared error, ridge*, lasso* and enet* show nearly the same performance as those based on the LQA approach. The OLS estimator (kq*) has the largest prediction error and mean squared error among all procedures in this setting.

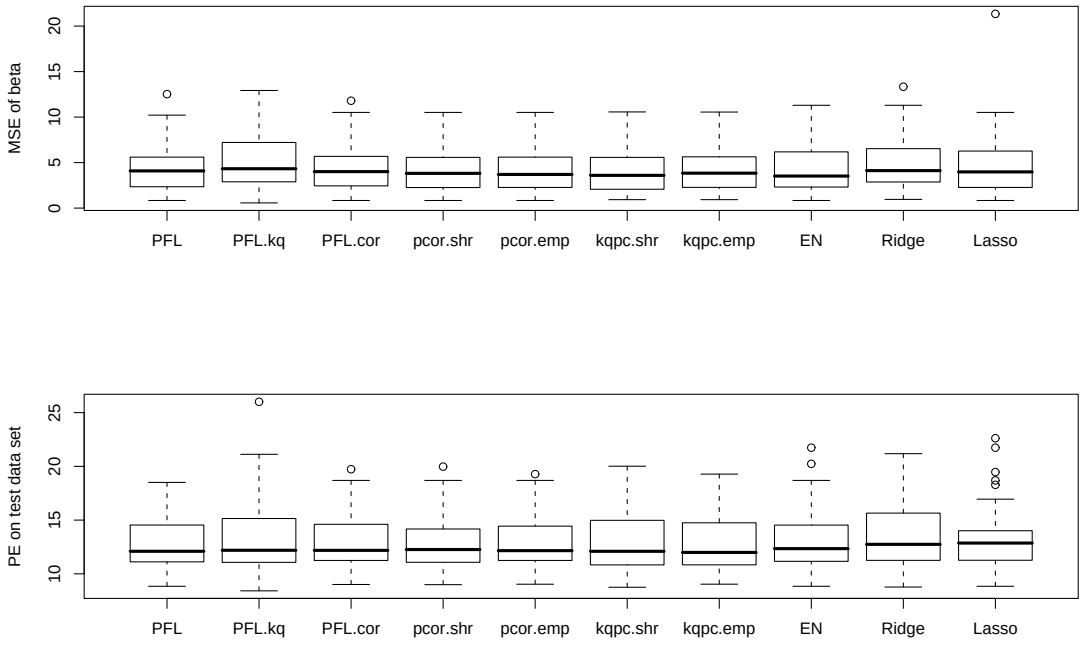

Figure 5.1.: Boxplots of the prediction error on the test data set and MSE of $\boldsymbol{\beta}$ for the 1st simulation setting and correlation $\rho = 0.5$

The first setting is also realized for the correlation $\rho = 0.9$. The results are given in Tables 5.3, A.1 and Figures A.1, A.2 from the appendix. If we consider the mean squared error and the prediction error, pfl.cor has the best performance. However in comparison to the setting for correlation $\rho = 0.5$, the accuracy of the parameter estimates decreases. By means of the minimal number of effective degrees of freedom, pfl.kq has the strongest clustering. Although pfl.kq considers all variables as relevant (3 hits, 5 false positives), there are only 3 effective degrees of freedom. This means that the coefficients of the 8 predictors form 3 clusters in the case of highly correlated predictors.

To draw a comparison, in the following sections the results for the methods pfl(lars), pfl.kq(lars), pfl.cor(lars) and kq*, ridge*, lasso*, enet* are listed in the tables. We will explicitly highlight basic differences compared to the corresponding methods based on the LQA approach. For all other cases, the description of the results refers to the procedures computed by function `cv.lqa()`. Therefore, only for these procedures the boxplots of the prediction error and the mean squared error as well as the boxplots of the predictors are illustrated in the figures.

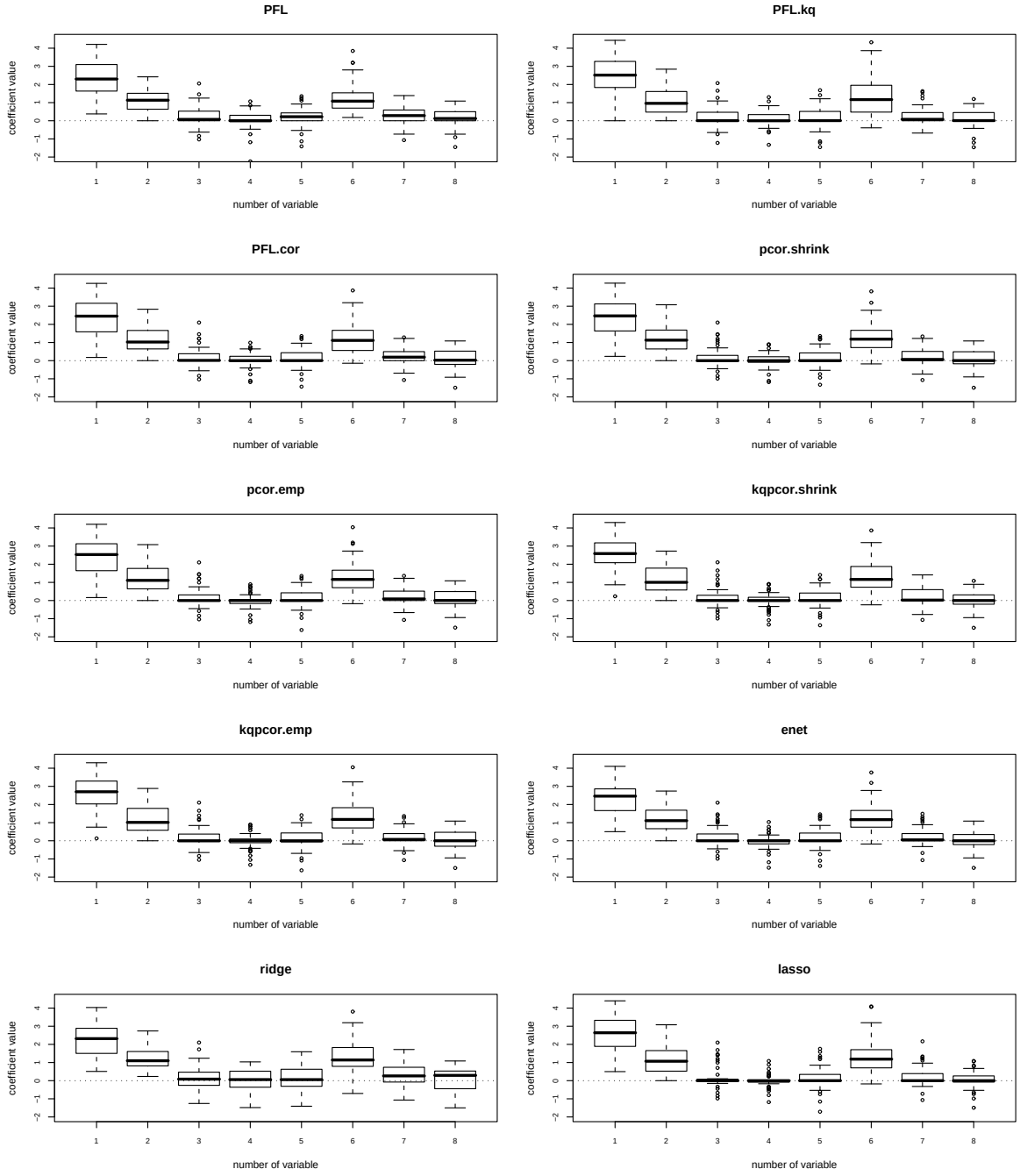

Figure 5.2.: Boxplots of the predictors for the 1st simulation setting and correlation $\rho = 0.5$

Predictor	Method									
	pfl		pfl.kq		pfl.cor		pcor.shrink		pcor.emp	
β_1	2.30	(0.23)	2.52	(0.18)	2.45	(0.28)	2.47	(0.25)	2.53	(0.25)
β_2	1.13	(0.10)	0.96	(0.18)	1.03	(0.12)	1.13	(0.15)	1.11	(0.13)
β_3	0.07	(0.08)	0.01	(0.06)	0.01	(0.05)	0.00	(0.04)	0.00	(0.03)
β_4	0.01	(0.03)	0.00	(0.03)	0.00	(0.02)	0.00	(0.00)	0.00	(0.01)
β_5	0.22	(0.10)	0.01	(0.08)	0.00	(0.09)	0.00	(0.05)	0.00	(0.03)
β_6	1.08	(0.11)	1.17	(0.17)	1.11	(0.14)	1.18	(0.14)	1.16	(0.13)
β_7	0.28	(0.06)	0.08	(0.10)	0.19	(0.11)	0.07	(0.10)	0.09	(0.09)
β_8	0.12	(0.11)	0.01	(0.09)	0.02	(0.08)	0.00	(0.07)	0.00	(0.07)

Predictor	Method									
	kqpcor.shr		kqpcor.emp		enet		ridge		lasso	
β_1	2.59	(0.22)	2.70	(0.26)	2.46	(0.17)	2.32	(0.21)	2.64	(0.20)
β_2	1.00	(0.15)	1.01	(0.13)	1.11	(0.16)	1.10	(0.10)	1.07	(0.17)
β_3	0.00	(0.04)	0.00	(0.03)	0.00	(0.03)	0.08	(0.10)	0.00	(0.00)
β_4	0.00	(0.01)	0.00	(0.01)	0.00	(0.02)	0.06	(0.16)	0.00	(0.00)
β_5	0.00	(0.03)	0.00	(0.01)	0.00	(0.04)	0.06	(0.19)	0.00	(0.00)
β_6	1.16	(0.21)	1.18	(0.22)	1.17	(0.10)	1.14	(0.12)	1.19	(0.11)
β_7	0.02	(0.08)	0.08	(0.11)	0.05	(0.08)	0.26	(0.12)	0.00	(0.07)
β_8	0.00	(0.02)	0.00	(0.04)	0.00	(0.05)	0.29	(0.18)	0.00	(0.01)

Table 5.2.: Medians of the predictors for the 1st simulation setting and correlation $\rho = 0.5$ based on 50 replications.

Method	Median						
	PE		MSE		hits	false.pos	df.eff
pfl	11.26	(0.26)	7.90	(0.88)	3	5	5
pfl.kq	11.63	(0.33)	10.54	(0.68)	3	5	3
pfl.cor	**11.21**	(0.24)	**7.64**	(0.94)	3	5	5
pcor.shrink	11.32	(0.23)	8.85	(0.70)	3	5	5
pcor.emp	11.29	(0.20)	8.64	(0.61)	3	5	5
kqpcor.shrink	11.24	(0.37)	8.78	(0.90)	3	5	4
kqpcor.emp	11.26	(0.29)	8.89	(0.86)	3	5	4
enet	11.51	(0.27)	8.95	(0.57)	3	5	8
ridge	11.49	(0.31)	8.81	(0.62)	3	5	8
lasso	12.10	(0.44)	11.64	(1.83)	3	3	6
pfl(lars)	11.24	(0.25)	8.05	(1.00)	3	5	4
pfl.kq(lars)	**11.16**	(0.31)	8.69	(1.14)	3	4	3
pfl.cor(lars)	11.28	(0.24)	**7.90**	(1.03)	3	5	4
kq*	17.47	(1.13)	55.22	(8.13)	3	5	8
enet*	11.47	(0.25)	**8.21**	(0.79)	3	4	7
ridge*	**11.45**	(0.25)	8.91	(0.73)	3	5	8
lasso*	11.98	(0.30)	10.72	(1.89)	3	3	5

Table 5.3.: Results for the 1st simulation setting and correlation $\rho = 0.9$, based on 50 replications.

5.4. Results Setting 2 and 3

Although we investigate regularization methods with variable selection property we perform setting 2 which consists solely of relevant variables. For correlation $\rho = 0.5$, the results are given in Tables 5.4, A.2 and Figures A.3 and A.4 from the appendix.

Method	Median						
	PE		MSE		hits	false.pos	df.eff
pfl	10.25	(0.17)	**0.39**	(0.10)	8	–	3
pfl.kq	**10.08**	(0.23)	0.42	(0.15)	8	–	2
pfl.cor	10.94	(0.34)	0.93	(0.20)	8	–	5
pcor.shrink	10.97	(0.29)	1.12	(0.19)	8	–	4
pcor.emp	11.24	(0.35)	1.19	(0.28)	8	–	5
kqpcor.shrink	11.16	(0.28)	1.22	(0.19)	8	–	4
kqpcor.emp	11.24	(0.26)	1.42	(0.23)	8	–	5
enet	11.62	(0.26)	1.91	(0.14)	8	–	8
ridge	11.60	(0.22)	1.76	(0.14)	8	–	8
lasso	13.35	(0.48)	4.47	(0.36)	8	–	7
pfl(lars)	**10.23**	(0.22)	**0.40**	(0.10)	8	–	2
pfl.kq(lars)	10.30	(0.18)	0.42	(0.13)	8	–	2
pfl.cor(lars)	11.06	(0.35)	1.23	(0.29)	8	–	4
kq*	17.47	(1.13)	10.39	(1.39)	8	–	8
enet*	12.58	(0.37)	3.01	(0.27)	7	–	7
ridge*	**11.69**	(0.24)	**1.83**	(0.16)	8	–	8
lasso*	13.38	(0.41)	4.62	(0.46)	6	–	6

Table 5.4.: Results for the 2nd simulation setting and correlation $\rho = 0.5$, based on 50 replications.

In this setting, pfl and pfl.kq have the best performance with respect to the prediction error and the mean squared error. Furthermore, they have only 3 and 2 effective degrees of freedom, respectively. The lasso has both the worst prediction and the worst mean squared error. This results from the fact that the lasso has the selection but not the grouping property. All methods identify the 8 relevant variables.

The results for the same setting but with highly correlated predictors ($\rho = 0.9$) are illustrated in Tables 5.5, A.3 and Figures A.5, A.6. Considering the prediction accuracy and the accuracy of the coefficient estimates, pfl.cor shows the best performance. Compared to the setting with medium correlated predictors (Tab. 5.4), the number of effective degrees of freedom in Table 5.5 decreases or remains the same. None of the pairwise fused lasso methods is able to estimate all coefficients to be equal and thus to form only one cluster. Instead, at least two groups are identified. Ridge regression and the elastic net estimate all parameters different (8 effective degrees of freedom). Because of the performed variable selection (6 hits), the lasso has only 6 effective degrees of freedom. This means that the lasso selects only some variables from the group of highly correlated predictors.

Method	Median						
	PE		MSE		hits	false.pos	df.eff
pfl	10.22	(0.27)	0.63	(0.24)	8	–	3
pfl.kq	10.17	(0.20)	0.96	(0.39)	8	–	2
pfl.cor	**10.13**	(0.24)	**0.27**	(0.10)	8	–	2
pcor.shrink	10.22	(0.20)	0.47	(0.29)	8	–	3
pcor.emp	10.36	(0.25)	0.97	(0.35)	8	–	3
kqpcor.shrink	10.35	(0.20)	0.98	(0.35)	8	–	3
kqpcor.emp	10.52	(0.28)	1.17	(0.36)	8	–	4
enet	10.65	(0.19)	1.49	(0.40)	8	–	8
ridge	10.61	(0.25)	1.28	(0.44)	8	–	8
lasso	11.87	(0.31)	10.20	(1.18)	6	–	6
pfl(lars)	**10.23**	(0.26)	0.38	(0.17)	8	–	2
pfl.kq(lars)	10.25	(0.21)	0.88	(0.18)	8	–	2
pfl.cor(lars)	**10.23**	(0.24)	**0.28**	(0.17)	8	–	2
kq*	17.47	(1.13)	55.22	(8.13)	8	–	8
enet*	**10.47**	(0.23)	1.97	(0.22)	8	–	8
ridge*	10.68	(0.24)	**1.39**	(0.42)	8	–	8
lasso*	11.84	(0.40)	10.90	(1.17)	5	–	5

Table 5.5.: Results for the 2nd simulation setting and correlation $\rho = 0.9$, based on 50 replications.

The third simulation setting contains two groups of relevant variables. The simulation results for this setting with correlation $\rho = 0.5$ are given in Tables 5.6, A.4 and Figures A.7, A.8, A.9. Note that pfl and pfl.cor perform best but the performance does not diverge for these methods, except for the lasso since the lasso is the only procedure which shrinks coefficients exactly to zero in this setting. Indeed the lasso does not identify all relevant variables, but its median of false positives equals 4.5. By contrast, the other procedures estimate the 10 non-influential variables influential. As illustrated in Figures A.8, A.9 and Table A.4, all procedures except the lasso estimate the 20 coefficients nearly equal. For instance, pfl.cor estimates all coefficients equal since it has only one effective degree of freedom. However, this value is only desirable if all variables are correctly specified. Note that in this setting enet* and the elastic net based on the LQA approach differ with regard to the hits and the false positives as well as the median of the mean squared error.

Method	Median						
	PE		MSE		hits	false.pos	df.eff
pfl	**239.77**	(2.97)	20.39	(0.25)	10	10	1.5
pfl.kq	240.50	(4.13)	20.82	(0.52)	10	10	2.5
pfl.cor	239.81	(2.99)	**20.35**	(0.20)	10	10	1
pcor.shrink	243.54	(4.03)	20.60	(0.59)	10	10	6
pcor.emp	244.44	(3.66)	21.53	(1.50)	10	10	7
kqpcor.shrink	245.59	(4.59)	21.38	(1.18)	10	10	6.5
kqpcor.emp	247.44	(4.39)	21.78	(1.84)	10	10	6
enet	247.28	(4.32)	22.37	(1.57)	10	10	19.5
ridge	242.92	(4.56)	21.63	(1.24)	10	10	20
lasso	261.56	(5.26)	54.01	(3.77)	7	4.5	10
pfl(lars)	**239.81**	(3.00)	20.38	(0.28)	10	10	1.5
pfl.kq(lars)	240.21	(4.21)	**20.32**	(0.38)	10	10	1
pfl.cor(lars)	**239.81**	(3.04)	20.48	(0.28)	10	10	1.5
kq*	379.96	(10.06)	284.16	(22.11)	10	10	20
enet*	251.85	(3.17)	35.26	(2.39)	7	5	12
ridge*	**246.75**	(3.62)	**21.98**	(1.79)	10	10	20
lasso*	266.87	(4.65)	54.82	(3.91)	6	4	9

Table 5.6.: Results for the 3rd simulation setting and correlation $\rho = 0.5$, based on 50 replications.

The simulation results for the third setting and correlation $\rho = 0.9$ are given in Tables 5.7, A.5 and Figures A.10, A.11, A.12. The regularization methods show the same behavior as for the case with correlation $\rho = 0.5$. Note that for this correlation structure lasso* has a much smaller mean squared error than the lasso based on the LQA approach. Furthermore, the PFL methods based on the LARS algorithm and those based on the LQA approach differ with regard to the prediction error.

In order to investigate whether the procedures identify grouped variables if the difference between influential and non-influential variables is much larger, we present another simulation scenario (Setting 7). Except for the true parameter vector, this scenario equals the third

Method	Median						
	PE		MSE		hits	false.pos	df.eff
pfl	233.96	(3.00)	20.54	(0.33)	10	10	1.5
pfl.kq	234.00	(3.78)	20.32	(0.30)	10	10	2
pfl.cor	233.96	(3.82)	**20.35**	(0.17)	10	10	1
pcor.shrink	**233.93**	(3.35)	20.80	(1.49)	10	10	4.5
pcor.emp	234.75	(2.69)	20.91	(2.45)	10	10	4
kqpcor.shrink	236.60	(3.97)	25.39	(2.97)	10	10	4
kqpcor.emp	235.39	(3.56)	26.99	(4.88)	10	10	4
enet	234.95	(2.72)	28.93	(3.17)	10	10	19
ridge	234.90	(3.01)	24.81	(1.83)	10	10	20
lasso	241.19	(4.37)	92.99	(6.60)	5	4	8
pfl(lars)	**249.81**	(2.81)	20.21	(0.39)	10	10	3
pfl.kq(lars)	253.26	(3.11)	**19.78**	(0.47)	10	10	3.5
pfl.cor(lars)	**249.81**	(2.96)	20.21	(0.38)	10	10	3
kq*	379.96	(10.06)	157.87	(12.28)	10	10	20
enet*	257.63	(6.02)	30.07	(2.79)	8	5	13
ridge*	**249.59**	(5.46)	**18.63**	(0.68)	10	10	20
lasso*	279.61	(4.92)	39.29	(2.15)	7	4	11

Table 5.7.: Results for the 3rd simulation setting and correlation $\rho = 0.9$, based on 50 replications.

setting and is also performed for both correlation $\rho = 0.5$ and correlation $\rho = 0.9$. Thereby, the parameter vector is given by

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

As illustrated in Table A.9 and Figures A.18, A.19, each procedure identifies the two groups of relevant variables in this simulation scenario with correlation $\rho = 0.5$. Note that indeed the group structure of the non-influential predictors is identified, but the corresponding coefficients are not set to zero. Considering the prediction error in Table 5.8 and Figure A.17, kqpcor.emp has the best prediction followed by the other PFL methods using partial correlations.

The simulation results for this setting with highly correlated predictors are given in Tables 5.9, A.10 and Figures A.20, A.21, A.22. The regularization methods again show the same behavior as described for the third simulation scenario with correlation $\rho = 0.5$. All PFL procedures estimate the 20 coefficients nearly equal 5 which is the mean of the true parameter vector in setting 7.

Method	Median						
	PE		MSE		hits	false.pos	df.eff
pfl	319.56	(5.62)	149.35	(10.04)	10	9	16
pfl.kq	329.58	(6.79)	168.78	(19.13)	10	10	7
pfl.cor	319.65	(6.71)	152.94	(11.62)	10	9	16
pcor.shrink	315.80	(4.27)	**140.83**	(12.00)	10	9	16
pcor.emp	315.88	(4.70)	145.50	(15.21)	10	9	16
kqpcor.shrink	315.56	(7.41)	149.01	(13.51)	10	9	14
kqpcor.emp	**312.22**	(6.51)	144.64	(13.22)	10	9	14
enet	317.95	(7.30)	142.71	(7.59)	10	9.5	19
ridge	324.15	(7.06)	160.41	(7.10)	10	10	20
lasso	324.79	(7.89)	163.79	(11.73)	10	8	17

Table 5.8.: Results for the 7th simulation setting and correlation $\rho = 0.5$, based on 50 replications.

Method	Median						
	PE		MSE		hits	false.pos	df.eff
pfl	279.95	(3.74)	460.86	(15.32)	10	10	8.5
pfl.kq	286.33	(4.91)	476.22	(12.58)	10	10	4
pfl.cor	279.92	(5.46)	477.59	(16.81)	10	10	8
pcor.shrink	283.60	(4.71)	426.46	(25.75)	10	10	9
pcor.emp	279.70	(4.95)	436.51	(23.09)	10	10	8
kqpcor.shrink	279.16	(4.73)	425.07	(16.73)	10	10	8
kqpcor.emp	277.94	(3.24)	413.13	(12.85)	10	10	8
enet	278.06	(5.45)	375.06	(14.53)	10	10	20
ridge	**274.85**	(3.69)	**367.99**	(9.68)	10	10	20
lasso	302.41	(4.86)	600.53	(29.57)	10	8	16

Table 5.9.: Results for the 7th simulation setting and correlation $\rho = 0.9$, based on 50 replications.

5.5. Results Setting 4 and 5

In the following, the results for the fourth and the fifth simulation scenario are presented. Due to their design, these two settings are well suited for studying the selection and the grouping property of regularization methods.

Method	Median						
	PE		MSE		hits	false.pos	df.eff
pfl	261.00	(5.45)	41.64	(8.67)	9	10	10.5
pfl.kq	**250.61**	(5.45)	107.32	(12.19)	9	7.5	6
pfl.cor	265.75	(3.82)	38.51	(16.84)	9	8	10.5
pcor.shrink	263.02	(2.98)	37.42	(13.23)	9	9	11
pcor.emp	260.16	(3.20)	34.81	(14.35)	9	8.5	11
kqpcor.shrink	253.09	(3.53)	112.98	(9.69)	5	2	6.5
kqpcor.emp	253.74	(3.78)	111.19	(10.79)	5	2.5	7
enet	260.64	(2.69)	33.72	(9.09)	9	8.5	16
ridge	264.33	(3.49)	**24.83**	(0.88)	9	11	20
lasso	264.90	(4.02)	100.67	(6.20)	7	5	11
pfl(lars)	260.32	(6.41)	47.37	(9.03)	9	9.5	8
pfl.kq(lars)	**253.61**	(4.66)	94.07	(22.72)	7	4	5
pfl.cor(lars)	262.84	(3.18)	**38.82**	(15.82)	9	7	10
kq*	369.87	(7.82)	4057.24	(315.11)	9	11	20
enet*	**262.27**	(5.41)	**20.40**	(5.79)	9	4	11
ridge*	266.21	(3.07)	24.90	(0.88)	9	11	20
lasso*	265.05	(3.80)	105.96	(7.96)	4	5	8

Table 5.10.: Results for the 4th simulation setting, based on 50 replications.

The prediction errors and the mean squared errors given in Tables 5.10, 5.11 and Figures A.13, 5.3 show an oppositional behavior for the PFL procedures using weights based on the ordinary least squares estimates. Both in setting 4 and in setting 5, pfl.kq has the best prediciton, followed by kqpcor.shrink and kqpcor.emp. However, considering the accuracy of the parameter estimates, pfl.kq, kqpcor.shrink and kqpcor.emp have maximum mean squared errors. Since the ordinary least squares estimates are not correctly determined, the mean squared error of the corresponding PFL methods tends to be much larger.
Note that in setting 5 pfl.kq(lars) and pfl.cor(lars) have almost the same prediction errors as the corresponding PFL methods based on the LQA approach, but the mean squared errors differ strongly. Thereby, pfl.kq(lars) performs better than pfl.kq, whereas pfl.cor performs better than pfl.cor(lars) with respect to the mean squared error.
As illustrated in Figures A.14, A.15 and Table A.6 for setting 4 and in Figures 5.4, 5.5 and Table A.8 for setting 5, pfl, pfl.cor, pcor.shrink, pcor.emp, the elastic net and ridge regression work quite well in identifying the groups of relevant predictors. Except for ridge regression and pfl in setting 4 and additionally pfl.kq in setting 5, the procedures show good performance in selecting variables.

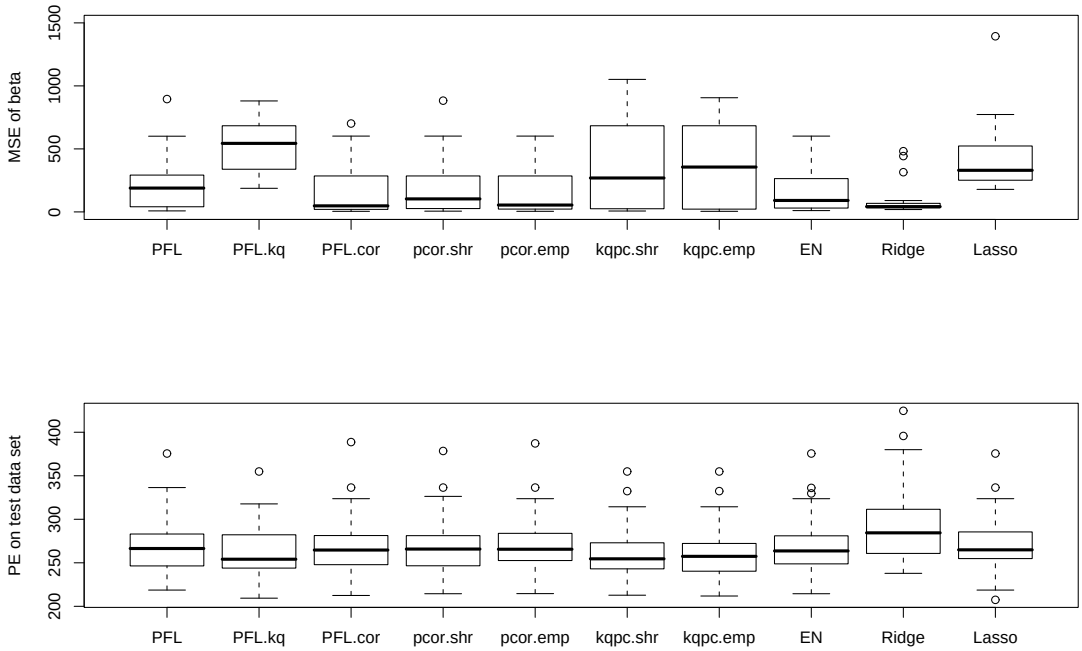

Figure 5.3.: Boxplots of the prediction error on the test data set and MSE of $\boldsymbol{\beta}$ for the 5th simulation setting

In order to detect whether the procedures identify grouped variables even if the predictors which form one group are not in adjacent columns in the design matrix, we present another simulation scenario (Setting 6). This setting equals setting 5 except for the parameter vector which is a permutation of the parameter vector in setting 5 and given by

$$\boldsymbol{\beta}_{true} = (5, 2, 0, 0, 0, 10, 0, 10, 2, 0, 0, 0, 0, 5, 10, 2, 0, 0, 5, 0)^{\mathrm{T}}.$$

The simulation results are given in Tables 5.12, A.7 and Figures A.16, 5.6, 5.7. The red squares in the boxplots of the predictors in Figures 5.6, 5.7 indicate the true parameters. As in setting 4 and 5, pfl.kq, kqpcor.shrink and kqpcor.emp have maximum mean squared errors, but the best performance with respect to the prediction error. Considering the boxplots of the predictors, ridge regression exceeds the other procedures, although it is not able to select predictors at all. Therefore, ridge regression has indeed the largest prediction error but has the best performance with respect to the accuracy of the parameter estimates.

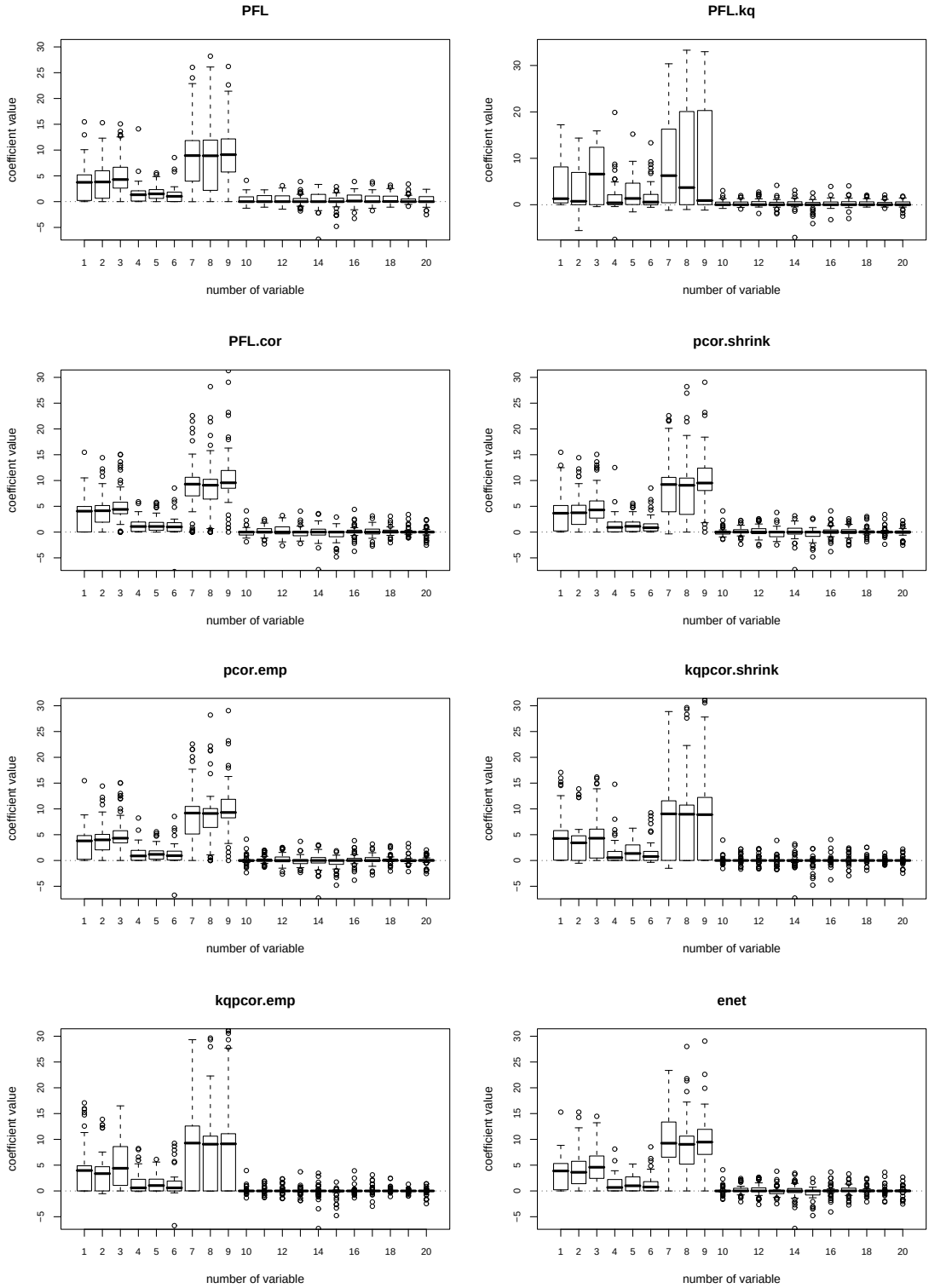

Figure 5.4.: Boxplots of the predictors for the 5th simulation setting

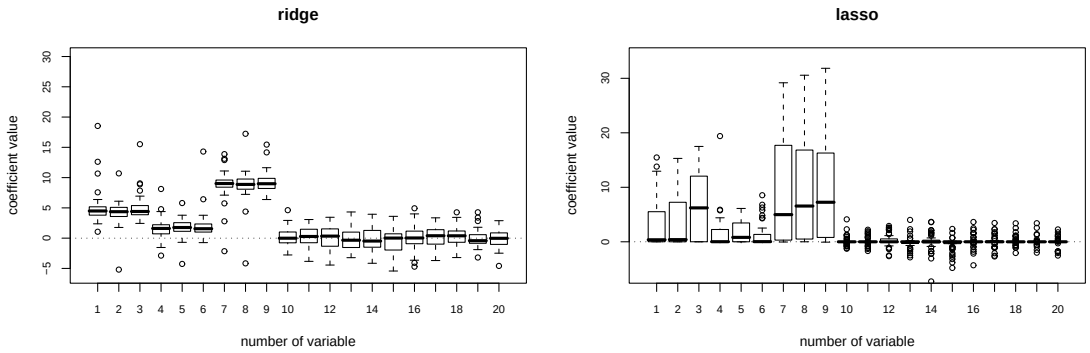

Figure 5.5.: Boxplots of the predictors for the 5th simulation setting

Method	Median						
	PE		MSE		hits	false.pos	df.eff
pfl	266.43	(4.86)	189.15	(44.89)	9	9	10.5
pfl.kq	**254.11**	(5.66)	543.81	(51.63)	9	10	6
pfl.cor	264.67	(4.36)	48.07	(76.95)	9	9	11.5
pcor.shrink	265.80	(3.98)	103.23	(72.95)	9	8.5	11
pcor.emp	265.59	(3.51)	54.40	(70.67)	9	8.5	11
kqpcor.shrink	254.58	(3.97)	269.23	(159.51)	9	5.5	8
kqpcor.emp	257.42	(3.37)	355.94	(156.73)	7	4.5	7.5
enet	263.65	(4.05)	90.80	(58.76)	9	7	15
ridge	284.43	(7.90)	**42.12**	(4.05)	9	11	20
lasso	264.90	(3.92)	330.20	(26.06)	7	5	11.5
pfl(lars)	266.99	(5.57)	**156.19**	(59.03)	9	7.2	8
pfl.kq(lars)	**257.02**	(5.15)	367.24	(140.86)	5	2	6
pfl.cor(lars)	265.76	(4.51)	207.66	(63.98)	8.5	7.5	10
kq*	369.87	(7.82)	4057.23	(315.11)	9	11	20
enet*	**262.96**	(4.16)	**36.58**	(9.38)	8	3.5	11
ridge*	283.97	(7.70)	43.36	(3.94)	9	11	20
lasso*	269.72	(3.75)	336.35	(37.31)	5	5	9

Table 5.11.: Results for the 5th simulation setting, based on 50 replications.

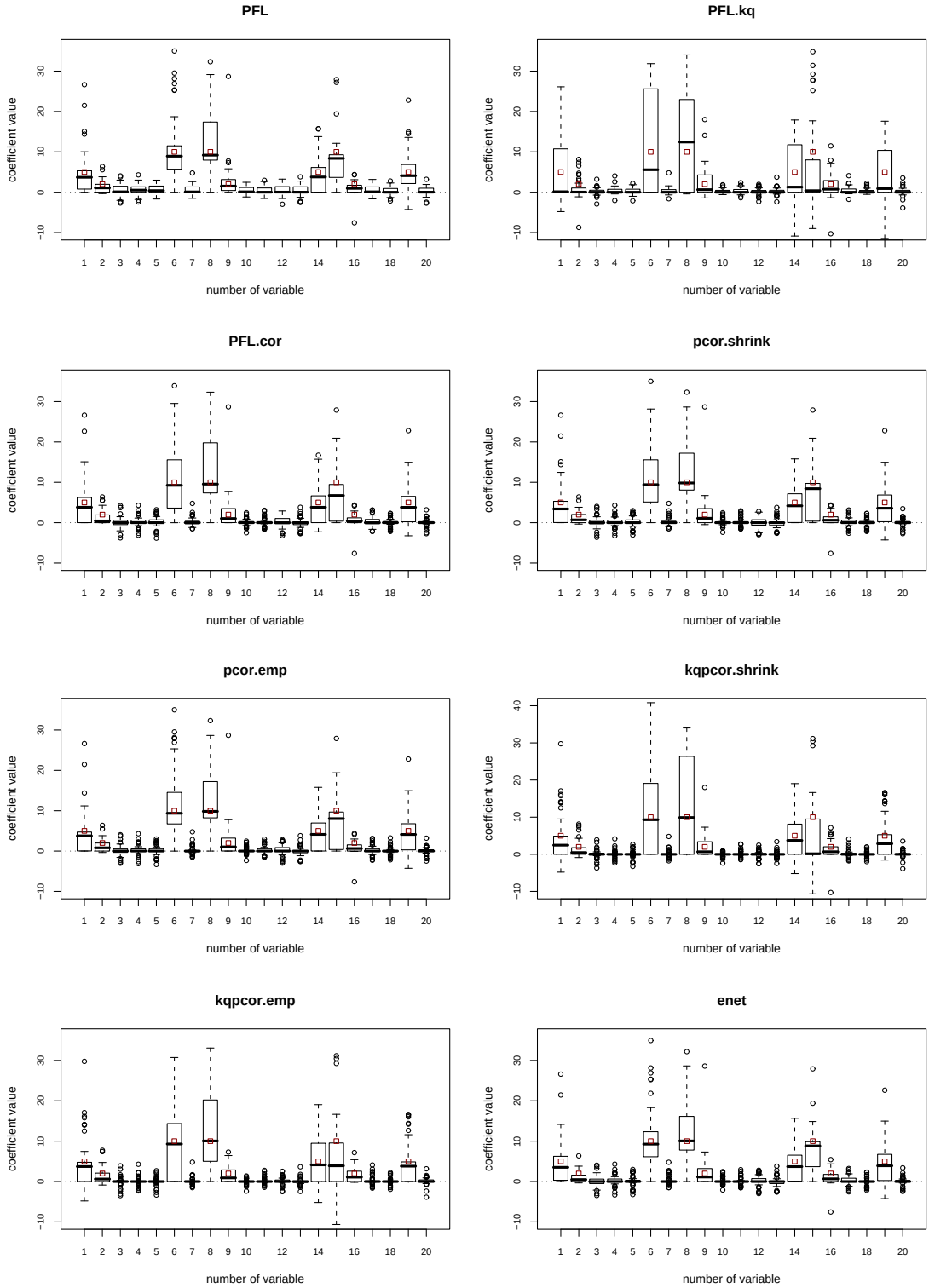

Figure 5.6.: Boxplots of the predictors for the 6th simulation setting

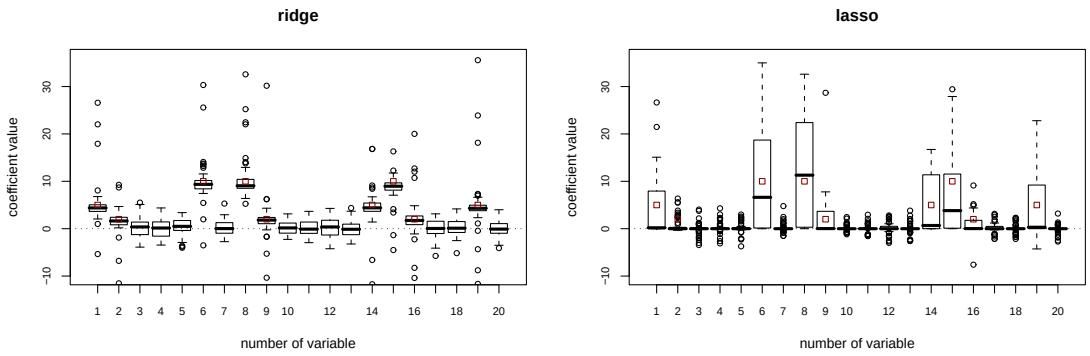

Figure 5.7.: Boxplots of the predictors for the 6th simulation setting

Method	Median						
	PE		MSE		hits	false.pos	df.eff
pfl	266.72	(4.75)	137.38	(89.23)	9	9	11
pfl.kq	258.30	(3.99)	610.57	(53.32)	8	6.5	6
pfl.cor	267.60	(4.87)	278.79	(38.50)	8	8	12
pcor.shrink	269.98	(4.88)	277.04	(47.84)	8	8	11
pcor.emp	265.01	(4.39)	254.21	(72.05)	8	8	11
kqpcor.shrink	258.40	(4.16)	607.10	(118.57)	6	4.5	8
kqpcor.emp	**256.36**	(4.09)	438.55	(161.53)	7.5	6	9
enet	267.40	(3.53)	143.10	(50.65)	8	7	15
ridge	283.67	(4.83)	**51.70**	(5.53)	9	11	20
lasso	264.08	(4.48)	377.02	(49.35)	7	5.5	12

Table 5.12.: Results for the 6th simulation setting, based on 50 replications.

6. Simulation Study II

6.1. Binary Regression

In this section, we present some simulations based on generalized linear models to investigate the performance of the pairwise fused lasso. For binary responses we choose the logit model (c.f. Sec. 4.1), i.e.

$$y_i \sim B(1, \pi_i) \;\text{ with }\; \pi_i = \frac{\exp(\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\beta}_{true})}{1 + \exp(\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\beta}_{true})}.$$

Basically, we use the same simulation settings as proposed in Section 5.1, except for the observation numbers of the training, the validation and the test data set. In each simulation scenario of this study, the observation numbers $n_{train}/n_{vali}/n_{test}$ correspond to $100/100/400$. Furthermore, the predictor $\eta_i = \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\beta}_{true}$ from the Normalcase is multiplied by a factor a in order to realize an appropriate domain for the logistic response function. The value range of the predictor should be approximately the interval $[-4, 4]$. Thus, for each setting we determine a factor a and multiply the true parameter vector from the Normalcase by this factor. The corresponding value of this factor and the modified parameter vector for each simulation setting are given by:

- *Setting 1:*
  $a = 0.40 \;\rightarrow\; \boldsymbol{\beta}_{true} = (1.2, 0.6, 0, 0, 0, 0.8, 0, 0)^{\mathrm{T}}$
- *Setting 2:*
  $a = 0.55 \;\rightarrow\; \boldsymbol{\beta}_{true} = \big(\underbrace{0.47, \ldots, 0.47}_{8}\big)^{\mathrm{T}}$
- *Setting 3:*
  $a = 0.15 \;\rightarrow\; \boldsymbol{\beta}_{true} = \big(\underbrace{0, \ldots, 0}_{5}, \underbrace{0.3, \ldots, 0.3}_{5}, \underbrace{0, \ldots, 0}_{5}, \underbrace{0.3, \ldots, 0.3}_{5}\big)^{\mathrm{T}}$
- *Setting 4:*
  $a = 0.15 \;\rightarrow\; \boldsymbol{\beta}_{true} = \big(\underbrace{0.45, \ldots, 0.45}_{9}, \underbrace{0, \ldots, 0}_{11}\big)^{\mathrm{T}}$
- *Setting 5:*
  $a = 0.10 \;\rightarrow\; \boldsymbol{\beta}_{true} = \big(0.75, 0.75, 0.75, 0.3, 0.3, 0.3, 1.5, 1.5, 1.5, \underbrace{0, \ldots, 0}_{11}\big)^{\mathrm{T}}$

The procedure to measure the model performance is basically the same as described in Section 5.2, thus we again use function `cv.lqa()` for model fitting. Additionally we make the following changes:

- Instead of the squared loss we use the deviance to evaluate the model performance on the validation data.
- The tuning parameter candidates are generated in the same manner as proposed in Section 5.2. However in contrast to the first simulation study, the length of the sequence for the tuning parameter α is 11 in each simulation scenario.
- To measure the performance of the selected model on the test data, we compute the pearson statistic. According to Fahrmeir and Tutz [FT01], the pearson statistic is defined by

$$\chi^2 = \sum_{i=1}^{n} \frac{(y_{i_{test}} - \widehat{\mu}_{i_{test}})^2}{v(\widehat{\mu}_{i_{test}})},$$

  where $\widehat{\mu}_{i_{test}} = h(\boldsymbol{x}_{i_{test}}^{\mathrm{T}} \widehat{\boldsymbol{\beta}}_{opt})$ and $v(\widehat{\mu}_{i_{test}})$ are the estimated mean and variance function, respectively.

In the following we present the simulation results. The table for each simulation scenario contains the medians of the pearson statistic (PS), the mean squared error (MSE), the hits, the false positives and the effective degrees of freedom. Both the threshold ϵ_1 which is used to distinguish influential and non-influential predictors and the threshold ϵ_2 which is used for the computation of the effective degrees of freedom are set to 0.0001 in this simulation study.

6.1.1. Results Setting 1, 2 and 3

Method	Median						
	PS		MSE		hits	false.pos	df.eff
pfl	380.70	(8.77)	0.51	(0.06)	3	5	7
pfl.ml	374.69	(9.16)	0.50	(0.06)	3	3.5	5
pfl.cor	376.67	(13.13)	0.51	(0.05)	3	5	6
pcor.shrink	375.28	(12.57)	0.48	(0.06)	3	5	7
pcor.emp	371.95	(11.93)	0.48	(0.06)	3	4	7
mlpcor.shrink	371.37	(11.93)	0.47	(0.05)	3	3	5
mlpcor.emp	376.48	(11.83)	0.48	(0.05)	3	3	5
enet	**368.29**	(9.08)	**0.46**	(0.05)	3	4	7
ridge	374.95	(13.14)	0.55	(0.04)	3	5	8
lasso	374.09	(10.72)	0.51	(0.05)	3	4	7

Table 6.1.: Results for the 1st simulation setting and correlation $\rho = 0.5$, based on 50 replications.

The simulation results for the first setting with correlation $\rho = 0.5$ are given in Table 6.1 and Figures 6.1, 6.2. Considering the median of the pearson statistic as measure for the adequacy of a model, the elastic net performs best. Furthermore, the elastic net has the smallest mean squared error followed by the PFL methods using partial correlations. Both

the median of the false positives and the effective degrees of freedom are the smallest for the PFL procedures using maximum likelihood estimates.
Comparing these results with the ones for correlation $\rho = 0.9$ given in Table 6.2 and Figures B.1, B.2 we observe that except for the lasso all procedures estimate the five non-influential predictors influential. The lasso has the worst performance both in terms of the prediction accuracy and in terms of the accuracy of the parameter estimates. As in the simulation study for normal distribution, the median of the prediction error and the pearson statistic, respectively, decreases for highly correlated predictors, but the median of the mean squared error increases for all regularization methods.

Method	Median						
	PS		MSE		hits	false.pos	df.eff
pfl	358.17	(11.07)	**0.97**	(0.12)	3	5	5
pfl.ml	360.05	(15.95)	1.27	(0.13)	3	5	4
pfl.cor	361.31	(14.49)	1.01	(0.11)	3	5	5
pcor.shrink	353.24	(11.14)	1.00	(0.10)	3	5	5
pcor.emp	352.69	(11.06)	1.04	(0.11)	3	5	5
mlpcor.shrink	357.38	(13.29)	1.00	(0.12)	3	5	5
mlpcor.emp	356.32	(12.63)	1.06	(0.12)	3	5	5
enet	354.26	(9.26)	1.06	(0.14)	3	5	7
ridge	**350.99**	(16.75)	1.15	(0.14)	3	5	8
lasso	366.78	(9.41)	1.42	(0.15)	3	4	6

Table 6.2.: Results for the 1st simulation setting and correlation $\rho = 0.9$, based on 50 replications.

For the second simulation scenario and correlation $\rho = 0.5$, the results are shown in Table 6.3 and Figures B.3, B.4, whereas the results for correlation $\rho = 0.9$ are illustrated in Table 6.4 and Figures B.5, B.6. Considering the median of the mean squared error and the effective degrees of freedom, the PFL methods dominate the elastic net, the lasso as well as ridge regression for both correlation structures. Thereby, pfl, pfl.ml and pfl.cor have the best performance amongst the PFL methods. Due to two effective degrees of freedom, pfl.ml shows the strongest grouping in each case. With respect to the pearson statistic, for correlation $\rho = 0.5$ the PFL methods using regularized partial correlations have the smallest median, whereas for correlation $\rho = 0.9$ the PFL methods using empirical partial correlations perform best. The lasso has in both cases the worst prediction and the worst mean squared error because of the missing grouping property. Furthermore, the lasso selects again only 6 of 8 relevant variables if they are highly correlated ($\rho = 0.9$).

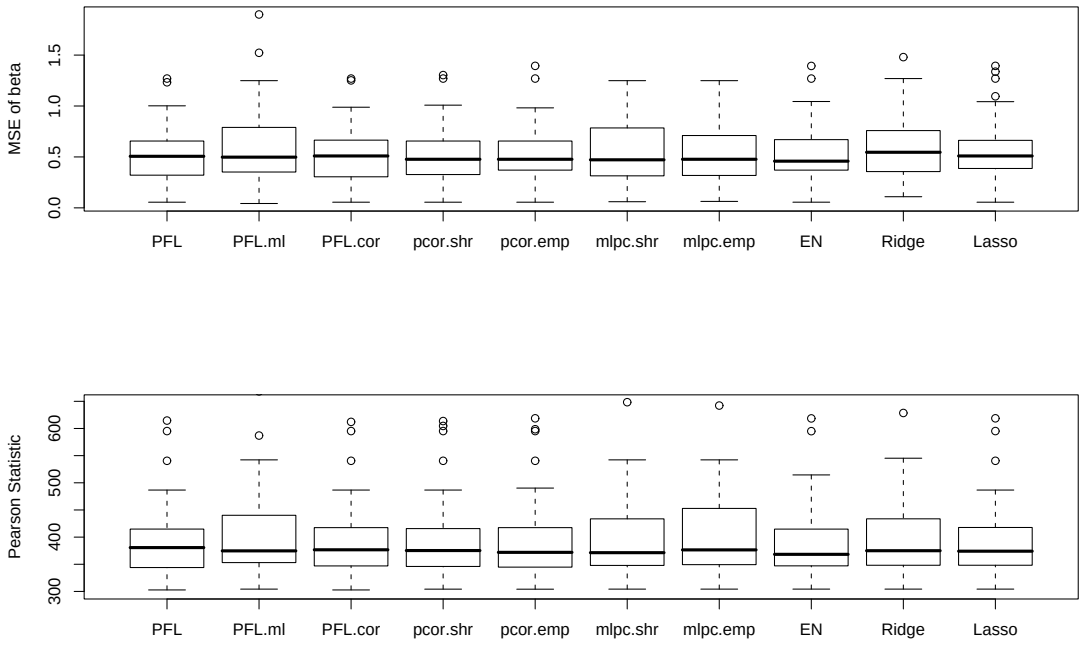

Figure 6.1.: Boxplots of the pearson statistic on the test data set and MSE of β for the 1st simulation setting and correlation $\rho = 0.5$

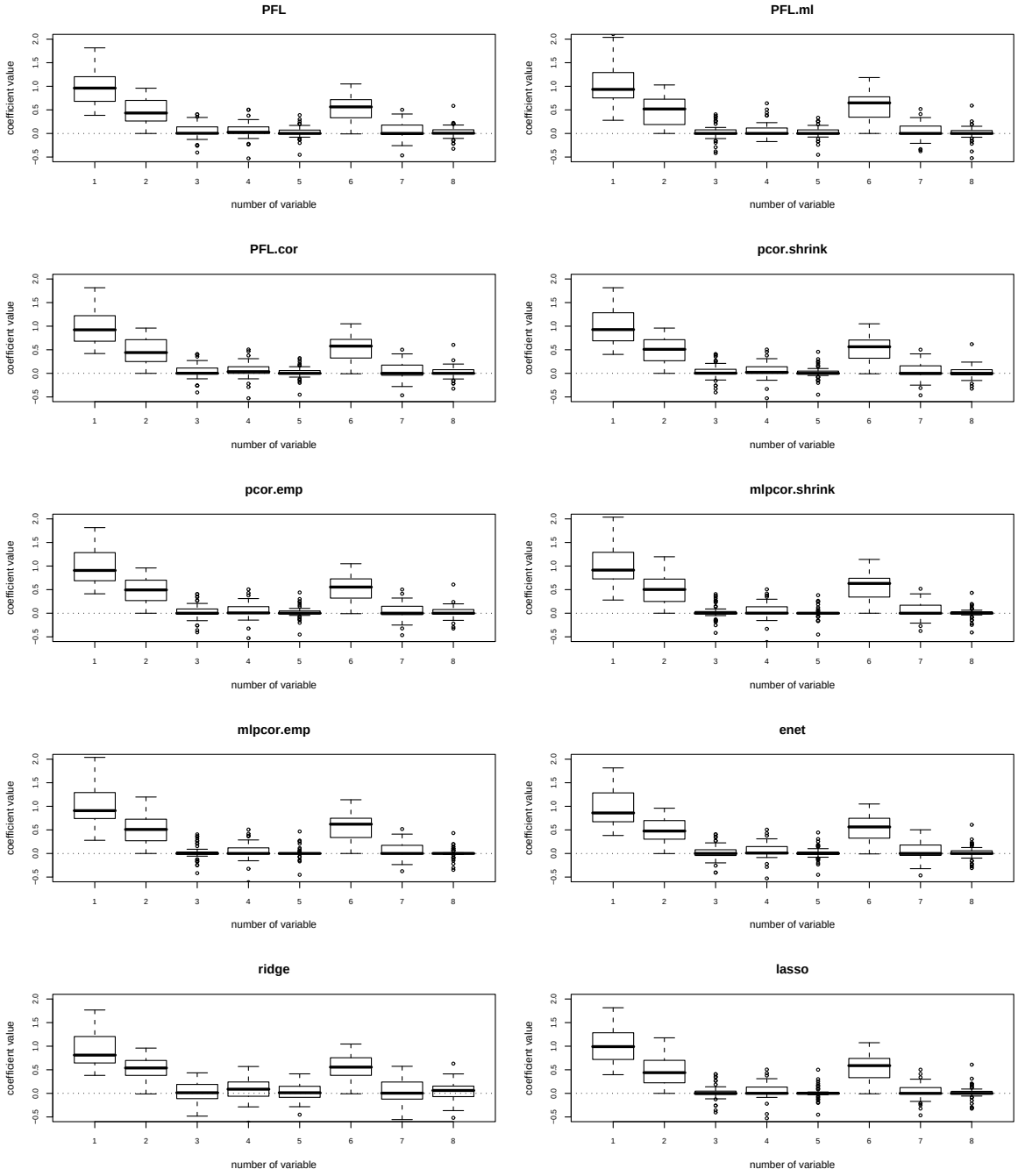

Figure 6.2.: Boxplots of the predictors for the 1st simulation setting and correlation $\rho = 0.5$

Method	Median						
	PS		MSE		hits	false.pos	df.eff
pfl	379.36	(12.01)	**0.08**	(0.02)	8	–	3.5
pfl.ml	375.72	(11.71)	**0.08**	(0.02)	8	–	2
pfl.cor	374.51	(11.78)	0.12	(0.01)	8	–	4.5
pcor.shrink	357.05	(8.89)	0.26	(0.03)	8	–	5
pcor.emp	360.93	(7.95)	0.29	(0.02)	8	–	6
mlpcor.shrink	**355.89**	(7.25)	0.27	(0.02)	8	–	5
mlpcor.emp	360.30	(7.93)	0.28	(0.02)	8	–	6
enet	379.29	(8.71)	0.43	(0.03)	8	–	8
ridge	379.88	(9.55)	0.43	(0.02)	8	–	8
lasso	429.54	(15.72)	0.65	(0.04)	8	–	8

Table 6.3.: Results for the 2nd simulation setting and correlation $\rho = 0.5$, based on 50 replications.

Method	Median						
	PS		MSE		hits	false.pos	df.eff
pfl	343.12	(21.53)	0.16	(0.06)	8	–	3
pfl.ml	336.35	(23.26)	0.17	(0.07)	8	–	2
pfl.cor	349.22	(18.81)	**0.12**	(0.05)	8	–	3
pcor.shrink	327.51	(20.70)	0.21	(0.05)	8	–	4
pcor.emp	**310.93**	(22.45)	0.22	(0.10)	8	–	4.5
mlpcor.shrink	316.72	(17.46)	0.21	(0.04)	8	–	4
mlpcor.emp	311.09	(17.81)	0.21	(0.05)	8	–	4
enet	320.02	(15.26)	0.63	(0.16)	8	–	8
ridge	316.78	(12.20)	0.51	(0.11)	8	–	8
lasso	361.90	(23.33)	1.86	(0.21)	6	–	6

Table 6.4.: Results for the 2nd simulation setting and correlation $\rho = 0.9$ based on 50 replications.

For the third setting, the regularization methods show the same behavior as described in the previous simulation study for normal distribution (Sec. 5.4). As illustrated in Figures B.8, B.9 for correlation $\rho = 0.5$ and Figures B.11, B.12 for correlation $\rho = 0.9$, all procedures except the lasso estimate the 20 regression coefficients nearly equal. Thus, the median of the hits and false positives (Tab. 6.5, 6.6) corresponds to 10 for these methods. For the setting with highly correlated predictors, the PFL methods, especially pfl, pfl.ml and pfl.cor, have a small number of effective degrees of freedom. This indicates a strong clustering. The boxplots of the pearson statistic and the mean squared error are shown in Figures B.7, B.10.

Method	Median						
	PS		MSE		hits	false.pos	df.eff
pfl	389.38	(13.31)	**0.47**	(0.01)	10	10	8
pfl.ml	369.52	(15.88)	**0.47**	(0.01)	10	10	4
pfl.cor	380.79	(11.67)	0.48	(0.01)	10	10	6
pcor.shrink	363.02	(9.77)	0.49	(0.02)	10	10	9
pcor.emp	363.83	(12.13)	0.52	(0.02)	10	10	9
mlpcor.shrink	357.82	(10.10)	0.51	(0.02)	10	10	8.5
mlpcor.emp	351.02	(8.93)	0.51	(0.02)	10	10	9.5
enet	345.36	(6.30)	0.53	(0.03)	10	9.5	19
ridge	**343.69**	(8.91)	0.48	(0.03)	10	10	20
lasso	376.37	(9.33)	0.84	(0.05)	7	5	12

Table 6.5.: Results for the 3rd simulation setting and correlation $\rho = 0.5$, based on 50 replications.

Method	Median						
	PS		MSE		hits	false.pos	df.eff
pfl	350.77	(26.92)	**0.47**	(0.05)	10	10	1
pfl.ml	358.77	(26.06)	0.48	(0.08)	10	10	2
pfl.cor	354.33	(26.12)	**0.47**	(0.07)	10	10	1.5
pcor.shrink	338.23	(25.39)	0.48	(0.07)	10	10	4.5
pcor.emp	345.62	(20.77)	0.48	(0.10)	10	10	4
mlpcor.shrink	338.45	(23.66)	0.49	(0.09)	10	10	4
mlpcor.emp	355.23	(20.07)	0.49	(0.12)	10	10	4
enet	343.74	(21.04)	0.83	(0.17)	10	10	19.5
ridge	**325.06**	(18.77)	0.60	(0.06)	10	10	20
lasso	355.22	(19.02)	2.24	(0.12)	5	4	9

Table 6.6.: Results for the 3rd simulation setting and correlation $\rho = 0.9$ based on 50 replications.

6.1.2. Results Setting 4 and 5

For the fourth simulation setting, the results are given in Table 6.7 and Figures B.13, B.14, B.15, whereas the results for the fifth setting are illustrated in Table 6.8 and Figures 6.3, 6.4, 6.5. Both in setting 4 and in setting 5, besides the lasso the PFL methods using weights based on maximum likelihood estimates have the worst performance with respect to the mean squared errors. However, these PFL procedures have the smallest medians of both the false positives and the effective degrees of freedom. Considering the boxplots of the predictors, ridge regression works best in identifying the groups of relevant predictors although it has neither the variable selection nor the grouping property.

Method	Median						
	PS		MSE		hits	false.pos	df.eff
pfl	353.62	(8.61)	0.70	(0.24)	9	9	11
pfl.ml	359.78	(16.37)	2.24	(0.23)	8.5	6	6
pfl.cor	**343.64**	(11.66)	1.03	(0.33)	9	8	11
pcor.shrink	358.33	(12.22)	0.98	(0.24)	9	8	11
pcor.emp	347.01	(7.85)	0.65	(0.18)	9	8.5	11
mlpcor.shrink	361.86	(11.30)	1.69	(0.47)	5	3	7
mlpcor.emp	352.60	(8.12)	1.87	(0.49)	5	2.5	7
enet	349.43	(7.13)	1.17	(0.24)	9	8	15
ridge	357.84	(13.48)	**0.50**	(0.04)	9	11	20
lasso	356.27	(7.54)	2.00	(0.11)	6	4	11

Table 6.7.: Results for the 4th simulation setting based on 50 replications.

Method	Median						
	PS		MSE		hits	false.pos	df.eff
pfl	304.60	(11.09)	2.13	(0.33)	8	8	11
pfl.ml	334.68	(11.53)	4.17	(0.26)	8	4.5	6
pfl.cor	296.58	(13.55)	1.48	(0.43)	6	8	12
pcor.shrink	**292.04**	(11.93)	1.94	(0.36)	8	8	12
pcor.emp	292.98	(9.04)	1.75	(0.34)	8	7.5	12
mlpcor.shrink	324.67	(20.89)	4.95	(0.43)	5	2	7
mlpcor.emp	325.82	(20.02)	4.97	(0.36)	5	2	6.5
enet	294.25	(11.64)	1.73	(0.24)	8	6	14
ridge	307.36	(12.30)	**0.96**	(0.05)	9	11	20
lasso	301.22	(12.25)	3.30	(0.44)	7	5	12

Table 6.8.: Results for the 5th simulation setting based on 50 replications.

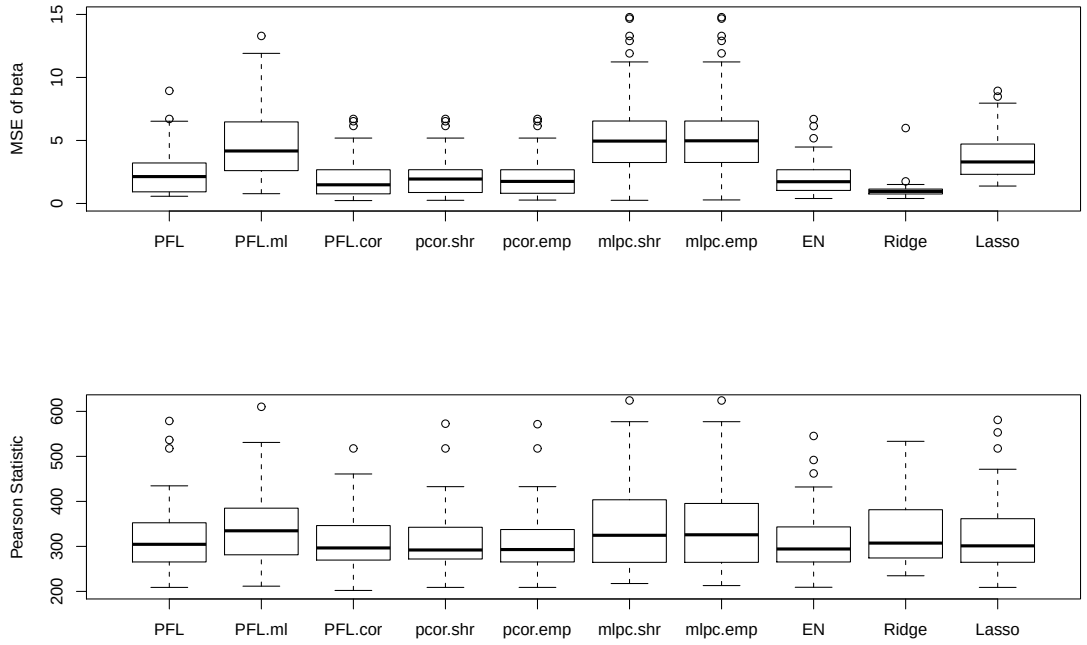

Figure 6.3.: Boxplots of the pearson statistic on the test data set and MSE of β for the 5th simulation setting

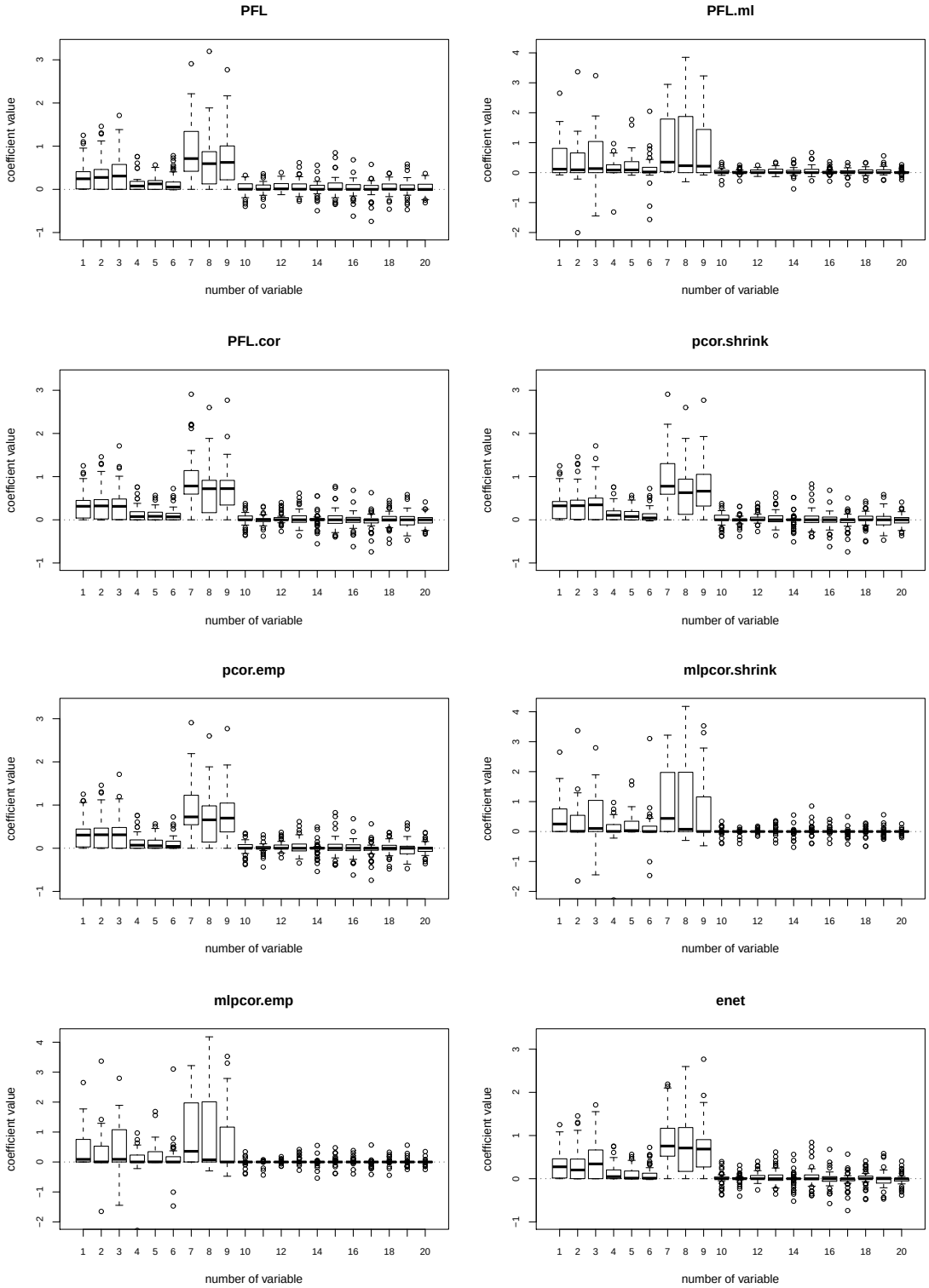

Figure 6.4.: Boxplots of the predictors for the 5th simulation setting

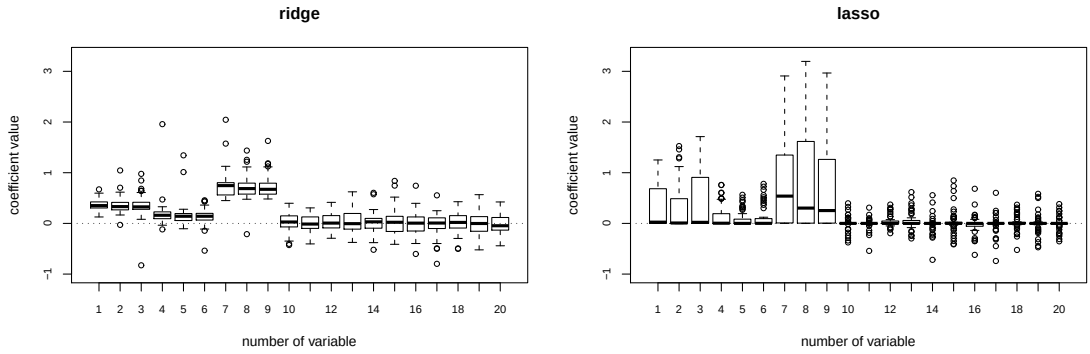

Figure 6.5.: Boxplots of the predictors for the 5th simulation setting

6.2. Poisson Regression

The simulations in this section are based on the Poisson model with the log-link as link function (c.f. Sec. 4.1), i.e.

$$y_i \sim Po(\lambda_i) \text{ with } \lambda_i = \exp(\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\beta}_{true}).$$

Analogously to the simulation study on binary responses (Sec. 6.1), the predictor $\eta_i = \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\beta}_{true}$ is multiplied by a factor a. Since the value range of the mean $\mu_i = \lambda_i$ should be approximately in the interval $[0, 8]$, we again determine for each setting (as proposed in Section 5.1) the corresponding factor a and multiply the true parameter vector by this factor. The observation numbers for the training, the validation and the test data set remain the same as in section 5.1. Thus, the modified parameter vectors are given by

- *Setting 1:*
  $a = 0.15 \quad \rightarrow \quad \boldsymbol{\beta}_{true} = (0.45, 0.225, 0, 0, 0, 0.3, 0, 0)^{\mathrm{T}}$
- *Setting 2:*
  $a = 0.20 \quad \rightarrow \quad \boldsymbol{\beta}_{true} = \big(\underbrace{0.17, \ldots, 0.17}_{8}\big)^{\mathrm{T}}$
- *Setting 3:*
  $a = 0.05 \quad \rightarrow \quad \boldsymbol{\beta}_{true} = \big(\underbrace{0, \ldots, 0}_{5}, \underbrace{0.1, \ldots, 0.1}_{5}, \underbrace{0, \ldots, 0}_{5}, \underbrace{0.3, \ldots, 0.3}_{5}\big)^{\mathrm{T}}$
- *Setting 4:*
  $a = 0.05 \quad \rightarrow \quad \boldsymbol{\beta}_{true} = \big(\underbrace{0.15, \ldots, 0.15}_{9}, \underbrace{0, \ldots, 0}_{11}\big)^{\mathrm{T}}$
- *Setting 5:*
  $a = 0.03 \quad \rightarrow \quad \boldsymbol{\beta}_{true} = \big(0.15, 0.15, 0.15, 0.06, 0.06, 0.06, 0.3, 0.3, 0.3, \underbrace{0, \ldots, 0}_{11}\big)^{\mathrm{T}}$

Note that the procedure and statistics to evaluate the model performance described for the simulations based on the logit model in Section 6.1 remain the same for this simulation study here.

In the following we present the simulation results. As in the simulation study for binary responses, the tables contain the medians of the pearson statistic (PS), the mean squared error (MSE), the hits, the false positives and the effective degrees of freedom. The figures which illustrate the boxplots of the predictors, the pearson statistic and the mean squared error are presented in Appendix C.

6.2.1. Results Setting 1, 2 and 3

Tables 6.9 and 6.10 show that the lasso identifies only two of the three relevant variables. However, the lasso has the smallest median of false positives. Considering the median of the mean squared error, the performance does not diverge for the regularization techniques except for the lasso in the setting with correlation $\rho = 0.9$.

Method	Median						
	PS		MSE		hits	false.pos	df.eff
pfl	**274.05**	(18.84)	0.20	(0.02)	3	5	4
pfl.ml	275.52	(15.16)	0.22	(0.01)	3	5	3
pfl.cor	282.06	(24.94)	0.20	(0.02)	3	5	4
pcor.shrink	280.86	(26.66)	0.20	(0.02)	3	4	4.5
pcor.emp	285.11	(17.87)	0.19	(0.02)	3	4	4
mlpcor.shrink	276.97	(18.13)	0.21	(0.03)	3	4	4
mlpcor.emp	277.99	(20.17)	0.22	(0.02)	3	4	5
enet	285.29	(18.81)	**0.18**	(0.02)	3	4	6
ridge	292.48	(15.80)	0.21	(0.01)	3	5	8
lasso	304.16	(18.73)	0.20	(0.04)	2	2	4

Table 6.9.: Results for the 1st simulation setting and correlation $\rho = 0.5$ based on 50 replications.

Method	Median						
	PS		MSE		hits	false.pos	df.eff
pfl	218.03	(8.08)	0.23	(0.01)	3	5	4
pfl.ml	223.00	(10.11)	0.25	(0.02)	3	5	2.5
pfl.cor	221.69	(11.31)	0.22	(0.01)	3	5	4
pcor.shrink	215.39	(7.45)	0.23	(0.01)	3	5	4
pcor.emp	212.82	(7.48)	0.23	(0.01)	3	5	4
mlpcor.shrink	218.38	(13.49)	0.24	(0.02)	3	5	4
mlpcor.emp	212.52	(13.42)	0.25	(0.02)	3	5	4
enet	**205.77**	(8.55)	0.22	(0.02)	3	5	8
ridge	210.32	(9.80)	**0.21**	(0.02)	3	5	8
lasso	221.06	(12.04)	0.32	(0.05)	2	3	5

Table 6.10.: Results for the 1st simulation setting and correlation $\rho = 0.9$ based on 50 replications.

Considering the effective degrees of freedom in Tables 6.11 and 6.12, pfl, pfl.ml and pfl.cor show the strongest grouping, especially in that setting with highly correlated predictors. In contrast to the simulation on binary responses, the lasso selects only six of eight relevant variables not only in the setting with correlation $\rho = 0.9$ but also in the setting with correlation $\rho = 0.5$.

Method	Median						
	PS		MSE		hits	false.pos	df.eff
pfl	**249.46**	(15.07)	**0.04**	(0.01)	8	–	3
pfl.ml	**249.46**	(14.99)	**0.04**	(0.01)	8	–	2
pfl.cor	266.02	(16.24)	0.11	(0.03)	8	–	4
pcor.shrink	264.94	(13.52)	0.11	(0.03)	8	–	5
pcor.emp	278.67	(18.59)	0.14	(0.02)	8	–	5
mlpcor.shrink	273.77	(11.04)	0.13	(0.04)	8	–	5
mlpcor.emp	284.64	(18.71)	0.16	(0.03)	8	–	5
enet	281.64	(10.08)	0.16	(0.02)	8	–	8
ridge	279.74	(12.91)	0.14	(0.01)	8	–	8
lasso	331.07	(20.20)	0.21	(0.02)	6	–	6

Table 6.11.: Results for the 2nd simulation setting and correlation $\rho = 0.5$ based on 50 replications.

Method	Median						
	PS		MSE		hits	false.pos	df.eff
pfl	225.12	(11.27)	**0.01**	(0.01)	8	–	1
pfl.ml	227.63	(10.48)	0.02	(0.02)	8	–	2
pfl.cor	222.48	(11.34)	**0.01**	(0.01)	8	–	1
pcor.shrink	216.48	(8.95)	**0.01**	(0.01)	8	–	3
pcor.emp	**216.36**	(11.67)	0.02	(0.01)	8	–	4
mlpcor.shrink	217.43	(7.52)	0.02	(0.01)	8	–	3
mlpcor.emp	216.86	(8.78)	0.03	(0.02)	8	–	4
enet	221.84	(11.05)	0.10	(0.02)	8	–	8
ridge	221.84	(11.50)	0.07	(0.01)	8	–	8
lasso	255.36	(18.37)	0.33	(0.05)	5.5	–	5

Table 6.12.: Results for the 2nd simulation setting and correlation $\rho = 0.9$ based on 50 replications.

The results for the third setting are given in Tables 6.13 and 6.14. Note that the behavior of the regularization procedures is again the same as described for the simulations based on the linear model (Sec. 5.4) as well as the logit model (Sec. 6.1.1). This means that again the procedures except the lasso estimate all regression coefficients nearly equal and thus consider all non-influential predictors influential (10 false positives).

Method	Median						
	PS		MSE		hits	false.pos	df.eff
pfl	407.92	(15.58)	**0.05**	(0.00)	10	10	2.5
pfl.ml	403.88	(16.15)	**0.05**	(0.00)	10	10	3
pfl.cor	415.39	(17.81)	**0.05**	(0.00)	10	10	2
pcor.shrink	398.62	(16.24)	**0.05**	(0.00)	10	10	5
pcor.emp	400.90	(16.02)	0.06	(0.00)	10	10	6
mlpcor.shrink	408.52	(17.64)	0.06	(0.01)	10	10	5.5
mlpcor.emp	402.22	(16.81)	0.07	(0.01)	10	10	6
enet	404.81	(18.30)	0.07	(0.01)	10	10	19
ridge	**392.28**	(14.67)	0.06	(0.01)	10	10	20
lasso	460.23	(25.61)	0.15	(0.02)	6	4	10

Table 6.13.: Results for the 3rd simulation setting and correlation $\rho = 0.5$ based on 50 replications.

Method	Median						
	PS		MSE		hits	false.pos	df.eff
pfl	420.23	(16.43)	**0.05**	(0.00)	10	10	1
pfl.ml	423.40	(16.59)	**0.05**	(0.00)	10	10	2
pfl.cor	423.97	(14.54)	**0.05**	(0.00)	10	10	1
pcor.shrink	415.71	(12.92)	0.06	(0.01)	10	10	4.5
pcor.emp	420.67	(14.46)	0.07	(0.01)	10	10	5
mlpcor.shrink	428.39	(12.17)	0.06	(0.01)	10	10	4
mlpcor.emp	425.47	(13.62)	0.06	(0.01)	10	10	4
enet	416.38	(15.01)	0.08	(0.01)	10	10	19
ridge	**409.55**	(16.66)	0.06	(0.00)	10	10	20
lasso	431.51	(16.20)	0.27	(0.02)	4	3	8

Table 6.14.: Results for the 3rd simulation setting and correlation $\rho = 0.9$ based on 50 replications.

6.2.2. Results Setting 4 and 5

Tables 6.15 and 6.16 illustrate that similarly for the fourth and fifth simulation setting the results for the Poisson model corresponds to those for the logit model (Sec. 6.1.2). Thus, with respect to the accuracy of the parameter estimates, pfl.ml, mlpcor.shrink, mlpcor.emp as well as the lasso have the worst performance.

Method	Median						
	PS		MSE		hits	false.pos	df.eff
pfl	468.07	(16.62)	0.10	(0.02)	9	10	10
pfl.ml	**432.27**	(13.97)	0.35	(0.05)	7	6	6
pfl.cor	501.11	(24.53)	0.19	(0.03)	7.5	8	10.5
pcor.shrink	490.96	(26.16)	0.19	(0.03)	7	6.5	10
pcor.emp	484.78	(19.49)	0.16	(0.04)	9	8	10
mlpcor.shrink	450.86	(16.54)	0.29	(0.04)	5.5	3	7
mlpcor.emp	455.47	(16.54)	0.29	(0.05)	5	3	7
enet	481.05	(19.87)	0.11	(0.02)	9	8	15
ridge	484.40	(18.27)	**0.07**	(0.01)	9	11	20
lasso	500.05	(25.03)	0.27	(0.01)	6	5	11

Table 6.15.: Results for the 4th simulation setting based on 50 replications.

Method	Median						
	PS		MSE		hits	false.pos	df.eff
pfl	476.20	(19.53)	0.19	(0.03)	9	9	10.5
pfl.ml	**428.16**	(13.41)	0.59	(0.04)	6	4	6
pfl.cor	483.16	(19.61)	0.26	(0.03)	6.5	7	10.5
pcor.shrink	474.66	(17.01)	0.25	(0.04)	8	7	10
pcor.emp	479.92	(16.59)	0.26	(0.03)	6.5	7	11
mlpcor.shrink	461.58	(23.31)	0.52	(0.06)	5	2	6
mlpcor.emp	457.27	(22.69)	0.53	(0.06)	5	2	6
enet	477.03	(23.03)	0.21	(0.04)	7	7	13
ridge	512.06	(26.45)	**0.13**	(0.01)	9	11	20
lasso	488.66	(21.14)	0.42	(0.05)	5	5	11

Table 6.16.: Results for the 5th simulation setting based on 50 replications.

7. Data Examples

Here, we present two real data examples in order to illustrate the application of the local quadratic approximation approach to generalized linear models. For one example the Gamma distribution is applied, whereas for the other example we choose the Poisson distribution. The tuning parameters for the pairwise fused lasso methods, the elastic net, the lasso as well as the ridge regression are selected via tenfold cross validation. Afterwards we estimate the corresponding models. We only specify the residual deviance of the fitted model because computationally it is too expensive to compute the prediction error on the data set of random splits. Furthermore, we evaluate the effective degrees of freedom.

7.1. Income Data Set

The income data set consists of the following 19 regressors: height, age, working position, working hours a week, married, employee, size of the company (20-200 employees, more than 200 employees), occupation (manager, scientist, engineer, office worker, sales assistant, farmer, craftsman, operator) and graduation (CSE, GCSE, Abitur). The goal is to estimate the income of 3344 women and 3752 men by using these regressors. Due to the non-negativity of the response, we choose the Gamma distribution (Sec. 4.1). Furthermore, for numerical reasons we prefer the log-link to the identity link. The results for the different regularization methods are shown in Table 7.1 (for men) and in Table 7.2 (for women).

	Method				
	pfl	pfl.ml	pfl.cor	pcor.shrink	pcor.emp
res.dev	494.7	494.6	494.5	494.5	494.5
df.eff	18	14	19	19	19

	Method				
	mlpcor.shr	mlpcor.emp	enet	ridge	lasso
res.dev	494.5	494.5	494.5	494.5	494.5
df.eff	19	19	19	19	19

Table 7.1.: Residual deviance and effective degrees of freedom for men.

Considering the residual deviance, the performance does not diverge for these methods. Except for pfl and pfl.ml, the procedures have 19 effective degrees of freedom.

	Method				
	pfl	pfl.ml	pfl.cor	pcor.shrink	pcor.emp
res.dev	528.6	528.5	528.5	528.5	528.5
df.eff	18	18	19	19	19

	Method				
	mlpcor.shr	mlpcor.emp	enet	ridge	lasso
res.dev	528.5	528.5	528.5	528.5	528.5
df.eff	19	19	19	19	19

Table 7.2.: Residual deviance and effective degrees of freedom for women.

7.2. Bones Data Set

This study aims at estimating the age by various measurements of bones for 87 persons. The underlying data set consists of 20 predictors: osteon fragments, osteon population density, type I osteon, type II osteon, Haverssche canals, non Haverssche canals, Volkmannsche canals, resorption lacuna, percentage of resorption lacuna, percentage of general lamellae, percentage of fragmental bones, percentage of osteonal bones, surface of an osteon, surface of a resorption lacuna, quotient of the surface of an resorption lacuna and the surface of an osteon, activation frequency, size of an compact bone, bone formation rate, femur class and gender. Some of the predictors are highly correlated, i.e. $\rho_{ij} \approx 0.9$. Furthermore, we choose the Poisson model and its canonical link (Sec. 4.1). The residual deviance and the effective degrees of freedom for each regularization method are illustrated in Table 7.3.

	Method				
	pfl	pfl.ml	pfl.cor	pcor.shrink	pcor.emp
res.dev	30.89	32.09	36.31	30.89	30.89
df.eff	13	9	8	13	13

	Method				
	kqpcor.shr	kqpcor.emp	enet	ridge	lasso
res.dev	32.09	35.26	30.89	31.03	31.80
df.eff	9	8	13	20	13

Table 7.3.: Residual deviance and effective degrees of freedom.

Except for ridge regression, all methods show the grouping and variable selection property, since the number of effective degrees of freedom is between eight and thirteen. Thereby, pfl, pcor.shrink, pcor.emp and the elastic net have the smallest residual deviance and thirteen effective degrees of freedom. The procedures pfl.cor and kqpcor.emp have only eight effective degrees of freedom but the largest residual deviance.

8. Conclusion

In this thesis, we examined the performance of the *pairwise fused lasso*, a new regularization method. Thereby, we computed the pairwise fused lasso solutions by the approximation procedures LQA [Ulb10b] and LARS [EHJT04] to evaluate the local quadratic approximation (LQA) approach for solving penalized regression problems. Furthermore, we computed ridge regression, the lasso and the elastic net with already established packages as well as the package `lqa` [Ulb10a]. Thus, on the one hand, we compared the pairwise fused lasso with already proposed regularization techniques, and on the other hand, again the LQA approach was evaluated. In the following, the most significant results of this thesis are summarized and approaches for further investigation are proposed.

The simulation studies in Chapters 5 and 6 show that the pairwise fused lasso represents a well-performing alternative regularization method which exhibits both the variable selection and the grouping property. Compared to the elastic net [ZH05], the simulation results illustrate that in most cases the pairwise fused lasso dominates the elastic net with respect to the prediction error on the test data set as well as the mean squared error of the parameter vector $\boldsymbol{\beta}$. Although the elastic net also has the grouping property, it performs worse in grouping highly correlated predictors since it exhibits large medians of the effective degrees of freedom in each simulation setting.
Considering the different modifications of the pairwise fused lasso penalty proposed in Section 3.3, we conclude that our choice of weights neither significantly improves the prediction accuracy nor the accuracy of coefficient estimates. Indeed in some simulation settings the pairwise fused lasso penalties using both partial correlations and ordinary least squares estimates have a smaller prediction error, but with respect to the mean squared error they have the worst performance among all regularization methods (c.f. setting 4 and 5 in Sec. 5.5). However, we encountered another possibility to modify the pairwise fused lasso penalty. In the correlation based pairwise fused lasso penalty (Eq. 3.6), we are weighting the difference of coefficients by $\frac{1}{1-|\rho_{jk}|}$. An alternative weighting would be $|\rho_{jk}|$. This means that if predictors are uncorrelated ($\rho_{jk} = 0$), the pairwise fused lasso penalty simplifies to the lasso penalty. For the fifth setting with highly correlated predictors within the groups (c.f. Sec. 5.1), the solutions for the pairwise fused lasso procedures using $|\rho_{jk}|$ in the penalty term are already computed. Thereby, we choose for ρ_{jk} the marginal correlation, the regularized partial correlation as well as the empirical partial correlation. Furthermore, we again considered the combination of partial correlations and additional weights $w_j = |\beta_j^{ML}|^{-1}$ for the lasso term in the penalty. The first results show that these modifications of the pairwise fused lasso penalty term achieve a considerable improvement of the mean squared error if the used correlations ρ_{jk} corresponds to regularized partial correlations.

Comparing the pairwise fused lasso solutions based on the LQA approach and the solutions based on the LARS algorithm, our simulations show that these two approaches lead to similar but not identical solutions since they work with different approximations. For the simulations based on the linear model in Chapter 5, we computed ridge regression, the lasso and the elastic net additionally with already established functions and packages, whereas the solutions show nearly the same performance as those based on the LQA approach. By contrast, the computations for the Poisson and logit model in Chapter 6 are solely based on the LQA approach. Indeed the algorithm implemented in package `lasso2` [LVTM09] provides the computation of penalized generalized linear models, but for the elastic net no package is established for fitting penalized Poisson models. However, for the computation of elastic net estimates for logistic regression models the package `glmnet` [FHT09] can be used. According to this, the main advantage of the local quadratic approximation approach is that it provides the computation of penalized generalized linear models and thereby comprises a large class of penalties and exponential families. Note that the LQA approach is computationally expensive. With respect to penalized generalized regression models, further simulation studies could address the comparison of solutions based on the LQA approach and those based on algorithms implemented in already established packages, such as `glmnet` [FHT09] and `glmpath` [PH07].

A. Simulations: Normal Distribution

A.1. Setting 1

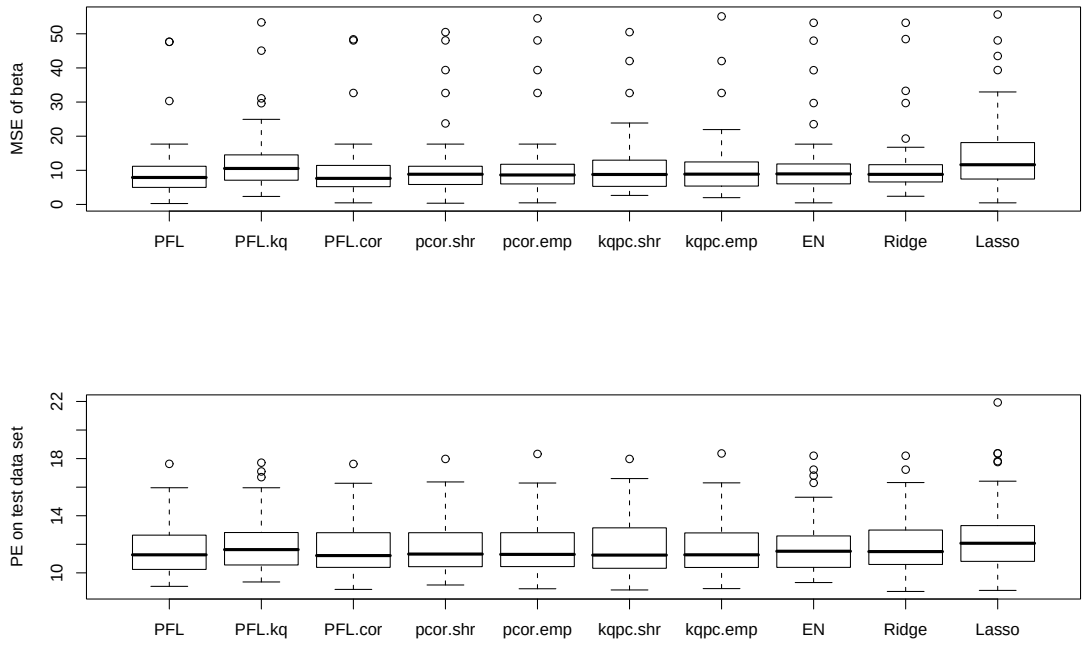

Figure A.1.: Boxplots of the prediction error on the test data set and MSE of β for the first simulation setting and correlation $\rho = 0.9$

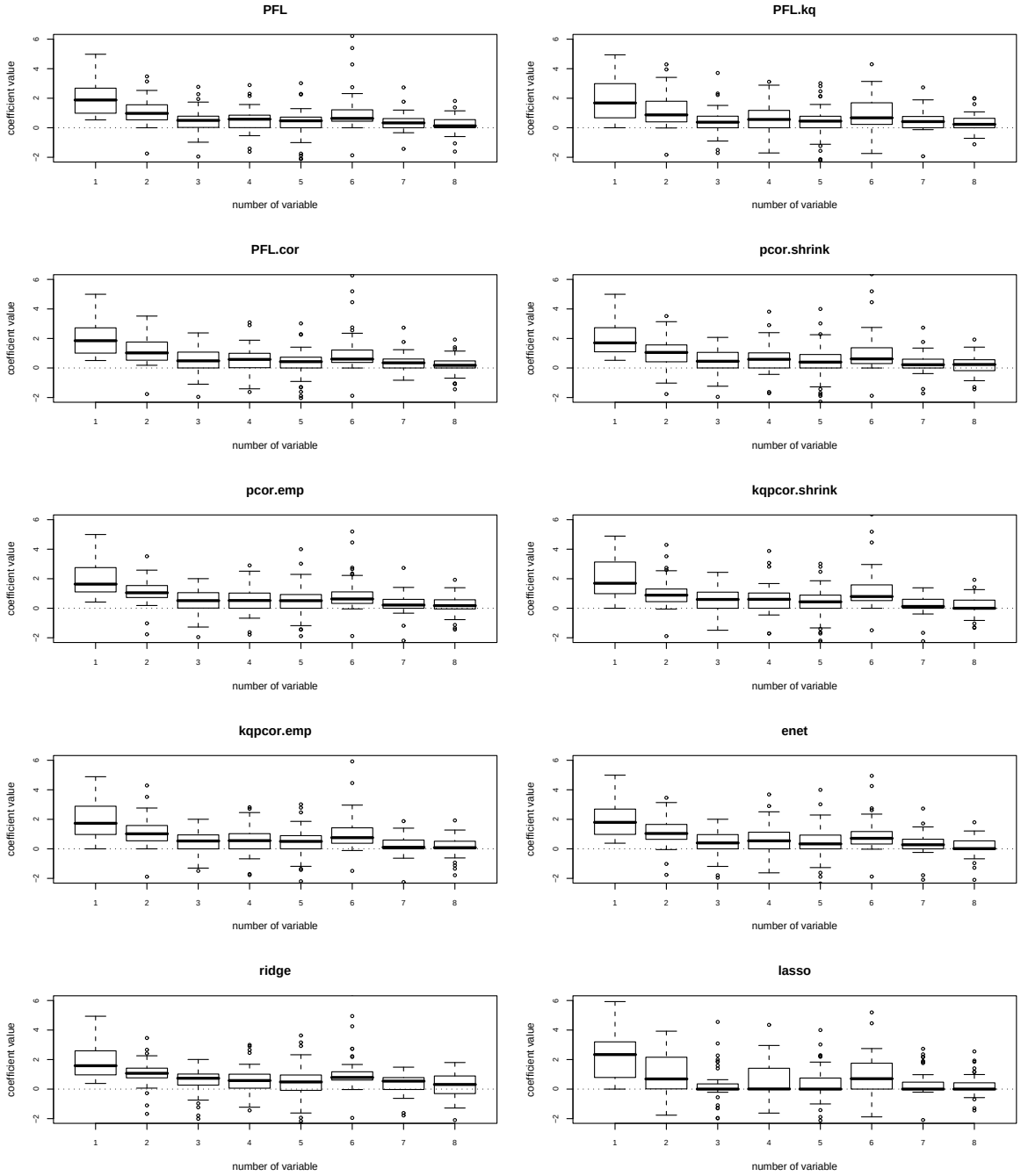

Figure A.2.: Boxplots of the predictors for the first simulation setting and correlation $\rho = 0.9$

Predictor	Method									
	pfl		pfl.kq		pfl.cor		pcor.shrink		pcor.emp	
β_1	1.88	(0.17)	1.68	(0.26)	1.85	(0.23)	1.70	(0.22)	1.63	(0.23)
β_2	0.97	(0.12)	0.87	(0.18)	1.02	(0.11)	1.05	(0.11)	1.05	(0.07)
β_3	0.50	(0.08)	0.38	(0.18)	0.48	(0.10)	0.46	(0.12)	0.52	(0.14)
β_4	0.58	(0.08)	0.56	(0.09)	0.58	(0.09)	0.59	(0.10)	0.53	(0.11)
β_5	0.47	(0.09)	0.45	(0.13)	0.43	(0.11)	0.40	(0.11)	0.51	(0.10)
β_6	0.63	(0.08)	0.67	(0.17)	0.60	(0.11)	0.62	(0.11)	0.63	(0.11)
β_7	0.33	(0.13)	0.42	(0.12)	0.35	(0.08)	0.22	(0.10)	0.22	(0.08)
β_8	0.13	(0.13)	0.23	(0.15)	0.18	(0.10)	0.24	(0.10)	0.18	(0.10)

Predictor	Method									
	kqpcor.shr		kqpcor.emp		enet		ridge		lasso	
β_1	1.69	(0.34)	1.73	(0.28)	1.79	(0.21)	1.58	(0.19)	2.34	(0.26)
β_2	0.89	(0.12)	1.02	(0.11)	1.04	(0.09)	1.07	(0.07)	0.69	(0.28)
β_3	0.60	(0.12)	0.53	(0.14)	0.40	(0.14)	0.74	(0.07)	0.00	(0.02)
β_4	0.60	(0.13)	0.56	(0.16)	0.54	(0.14)	0.58	(0.11)	0.01	(0.16)
β_5	0.44	(0.19)	0.51	(0.16)	0.34	(0.17)	0.48	(0.07)	0.00	(0.03)
β_6	0.80	(0.11)	0.76	(0.11)	0.71	(0.07)	0.79	(0.07)	0.70	(0.27)
β_7	0.12	(0.14)	0.11	(0.13)	0.28	(0.13)	0.54	(0.10)	0.00	(0.02)
β_8	0.00	(0.09)	0.08	(0.10)	0.02	(0.08)	0.31	(0.14)	0.00	(0.00)

Table A.1.: Medians of the predictors for the first simulation setting and correlation $\rho = 0.9$ based on 50 replications.

A.2. Setting 2

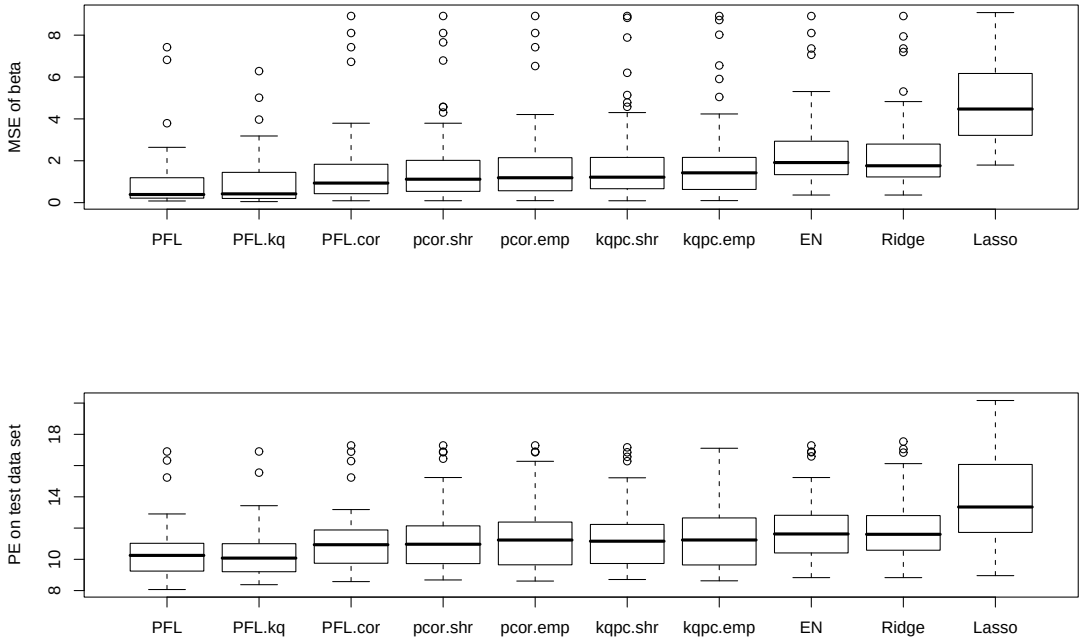

Figure A.3.: Boxplots of the prediction error on the test data set and MSE of β for the second simulation setting and correlation $\rho = 0.5$

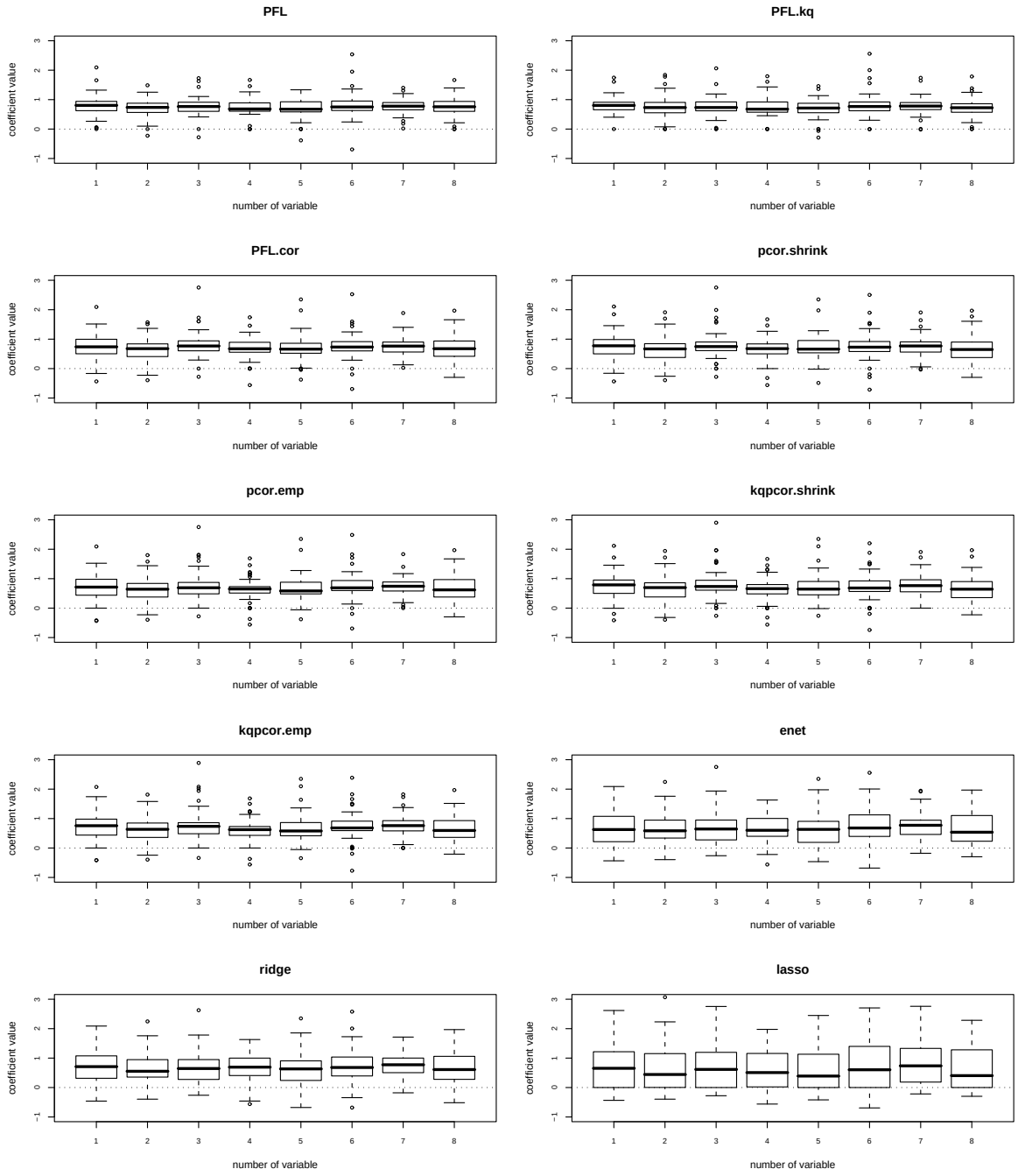

Figure A.4.: Boxplots of the predictors for the second simulation setting and correlation $\rho = 0.5$

Predictor	Method									
	pfl		pfl.kq		pfl.cor		pcor.shrink		pcor.emp	
β_1	0.81	(0.04)	0.80	(0.04)	0.74	(0.06)	0.77	(0.08)	0.71	(0.09)
β_2	0.74	(0.03)	0.73	(0.04)	0.68	(0.03)	0.67	(0.05)	0.64	(0.05)
β_3	0.77	(0.03)	0.73	(0.04)	0.77	(0.05)	0.75	(0.04)	0.69	(0.05)
β_4	0.68	(0.02)	0.68	(0.05)	0.67	(0.03)	0.68	(0.03)	0.65	(0.03)
β_5	0.67	(0.04)	0.71	(0.04)	0.66	(0.05)	0.66	(0.03)	0.58	(0.04)
β_6	0.75	(0.06)	0.77	(0.04)	0.73	(0.05)	0.73	(0.06)	0.69	(0.05)
β_7	0.78	(0.03)	0.78	(0.03)	0.76	(0.04)	0.77	(0.05)	0.74	(0.04)
β_8	0.76	(0.04)	0.72	(0.03)	0.68	(0.07)	0.65	(0.08)	0.62	(0.09)

Predictor	Method									
	kqpcor.shr		kqpcor.emp		enet		ridge		lasso	
β_1	0.79	(0.06)	0.76	(0.08)	0.63	(0.12)	0.71	(0.12)	0.65	(0.26)
β_2	0.70	(0.05)	0.63	(0.06)	0.58	(0.09)	0.55	(0.08)	0.44	(0.15)
β_3	0.73	(0.04)	0.74	(0.04)	0.65	(0.07)	0.65	(0.08)	0.62	(0.29)
β_4	0.66	(0.03)	0.62	(0.04)	0.60	(0.09)	0.69	(0.09)	0.51	(0.14)
β_5	0.65	(0.04)	0.58	(0.04)	0.63	(0.09)	0.63	(0.08)	0.39	(0.16)
β_6	0.68	(0.06)	0.68	(0.05)	0.68	(0.08)	0.68	(0.08)	0.60	(0.22)
β_7	0.76	(0.05)	0.76	(0.04)	0.77	(0.07)	0.77	(0.06)	0.74	(0.14)
β_8	0.64	(0.09)	0.60	(0.09)	0.54	(0.12)	0.61	(0.08)	0.40	(0.18)

Table A.2.: Medians of the predictors for the second simulation setting and correlation $\rho = 0.5$ based on 50 replications.

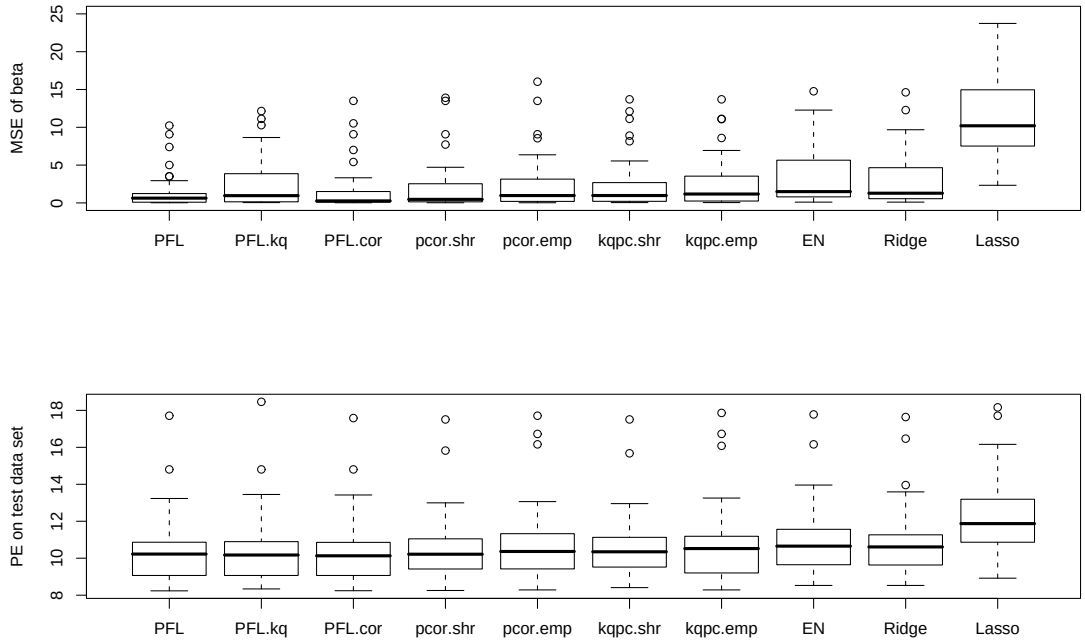

Figure A.5.: Boxplots of the prediction error on the test data set and MSE of $\boldsymbol{\beta}$ for the second simulation setting and correlation $\rho = 0.9$

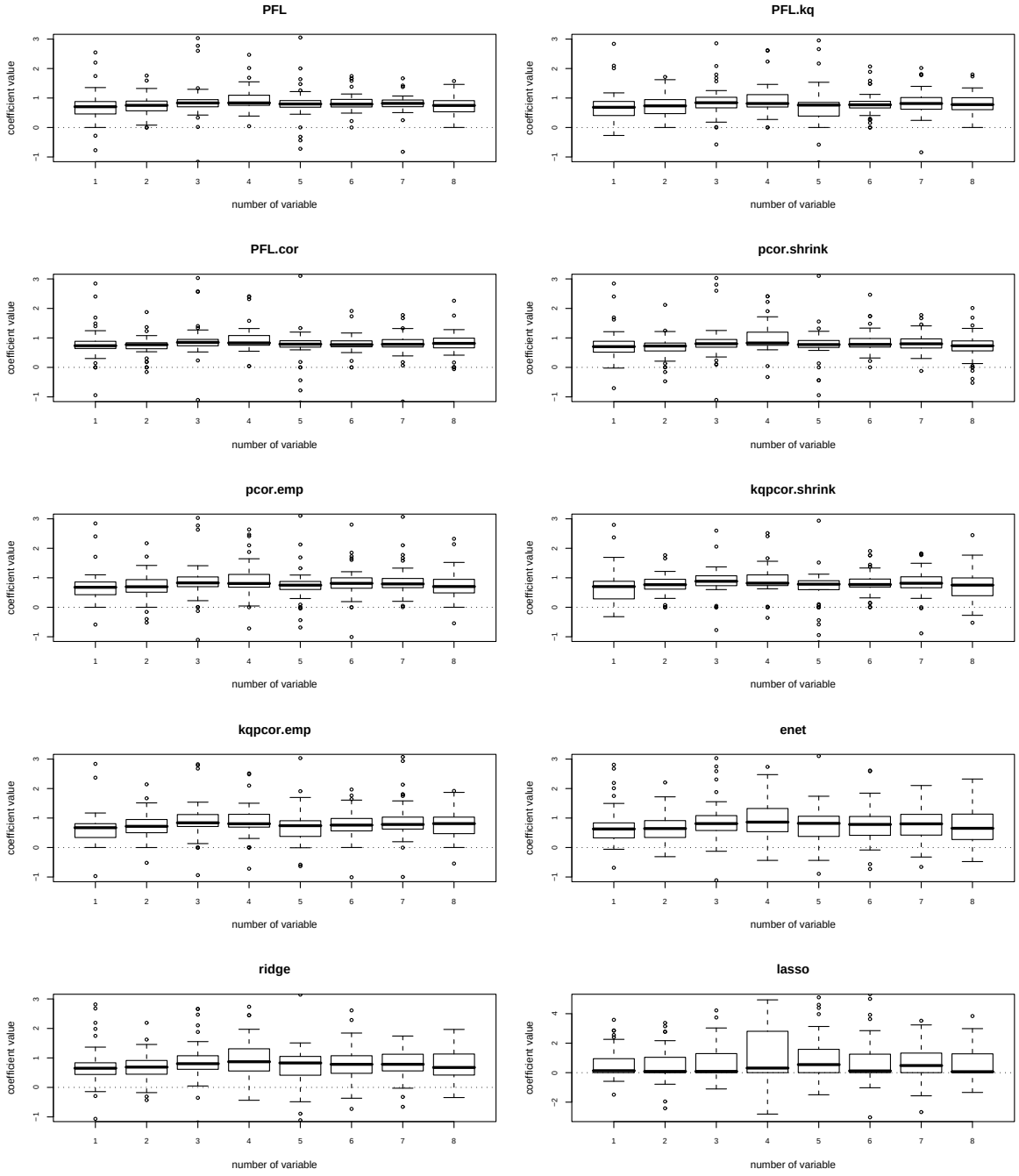

Figure A.6.: Boxplots of the predictors for the second simulation setting and correlation $\rho = 0.9$

Predictor	Method									
	pfl		pfl.kq		pfl.cor		pcor.shrink		pcor.emp	
β_1	0.70	(0.02)	0.69	(0.04)	0.74	(0.03)	0.70	(0.03)	0.68	(0.03)
β_2	0.75	(0.03)	0.73	(0.04)	0.77	(0.02)	0.73	(0.03)	0.70	(0.03)
β_3	0.83	(0.03)	0.84	(0.04)	0.85	(0.03)	0.80	(0.03)	0.83	(0.04)
β_4	0.83	(0.02)	0.81	(0.03)	0.83	(0.03)	0.83	(0.04)	0.81	(0.05)
β_5	0.80	(0.02)	0.76	(0.04)	0.78	(0.02)	0.77	(0.02)	0.75	(0.03)
β_6	0.79	(0.03)	0.77	(0.03)	0.77	(0.03)	0.78	(0.04)	0.81	(0.05)
β_7	0.82	(0.02)	0.81	(0.03)	0.78	(0.03)	0.80	(0.03)	0.79	(0.04)
β_8	0.75	(0.04)	0.78	(0.07)	0.81	(0.05)	0.73	(0.05)	0.70	(0.05)

Predictor	Method									
	kqpcor.shr		kqpcor.emp		enet		ridge		lasso	
β_1	0.70	(0.04)	0.67	(0.04)	0.63	(0.07)	0.65	(0.04)	0.13	(0.15)
β_2	0.77	(0.03)	0.72	(0.04)	0.64	(0.08)	0.69	(0.06)	0.08	(0.14)
β_3	0.88	(0.03)	0.84	(0.04)	0.81	(0.05)	0.80	(0.05)	0.09	(0.22)
β_4	0.82	(0.03)	0.81	(0.04)	0.86	(0.08)	0.87	(0.07)	0.31	(0.38)
β_5	0.78	(0.03)	0.74	(0.05)	0.82	(0.08)	0.83	(0.06)	0.55	(0.29)
β_6	0.77	(0.04)	0.76	(0.06)	0.78	(0.06)	0.78	(0.06)	0.12	(0.16)
β_7	0.81	(0.04)	0.78	(0.05)	0.80	(0.06)	0.78	(0.05)	0.48	(0.29)
β_8	0.75	(0.07)	0.81	(0.07)	0.65	(0.08)	0.68	(0.06)	0.07	(0.20)

Table A.3.: Medians of the predictors for the second simulation setting and correlation $\rho = 0.9$ based on 50 replications.

A.3. Setting 3

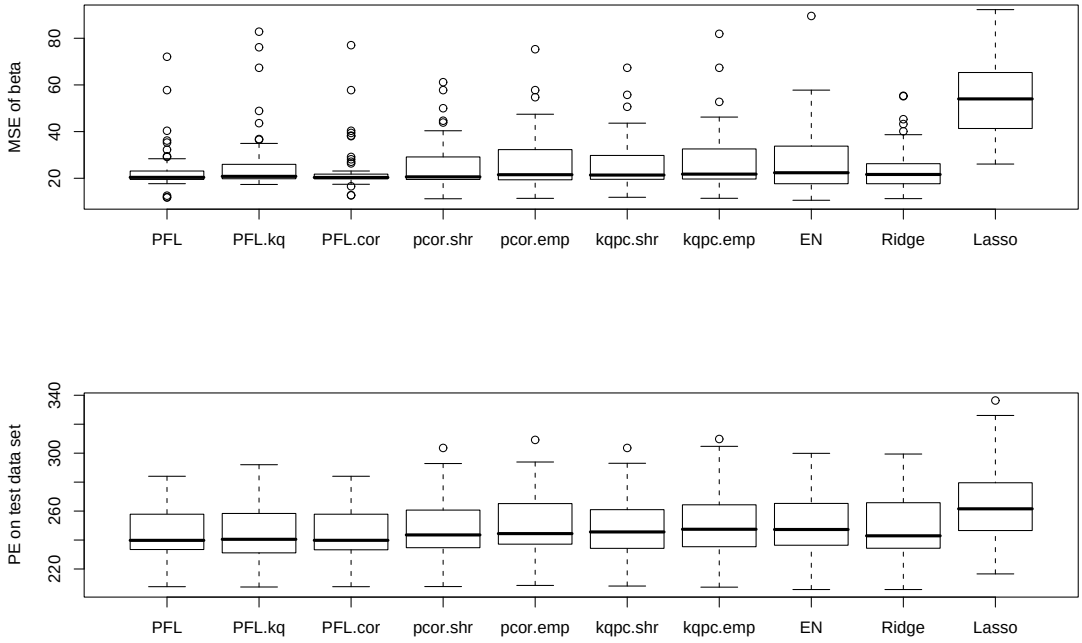

Figure A.7.: Boxplots of the prediction error on the test data set and MSE of $\boldsymbol{\beta}$ for the third simulation setting and correlation $\rho = 0.5$

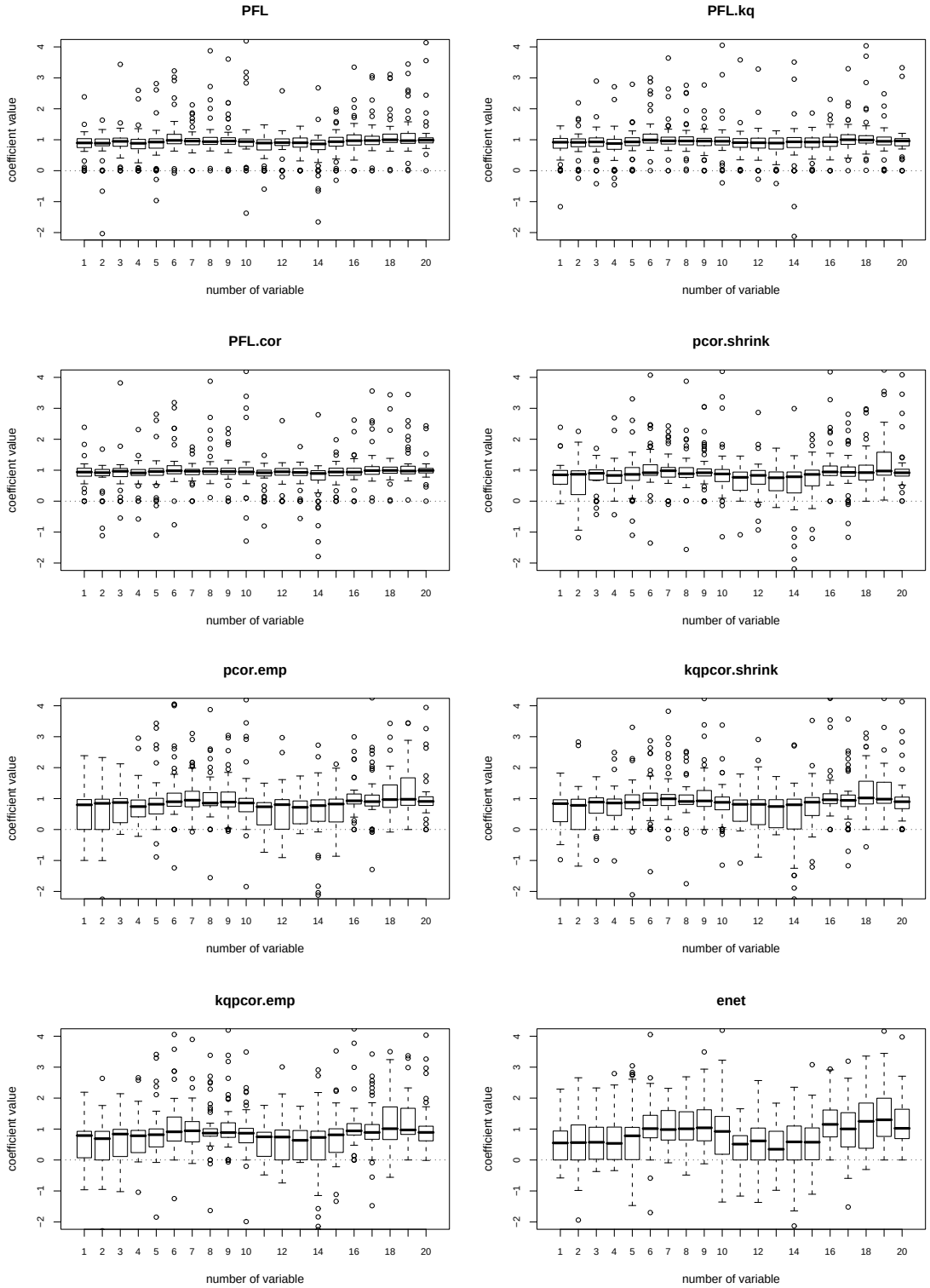

Figure A.8.: Boxplots of the predictors for the third simulation setting and correlation $\rho = 0.5$

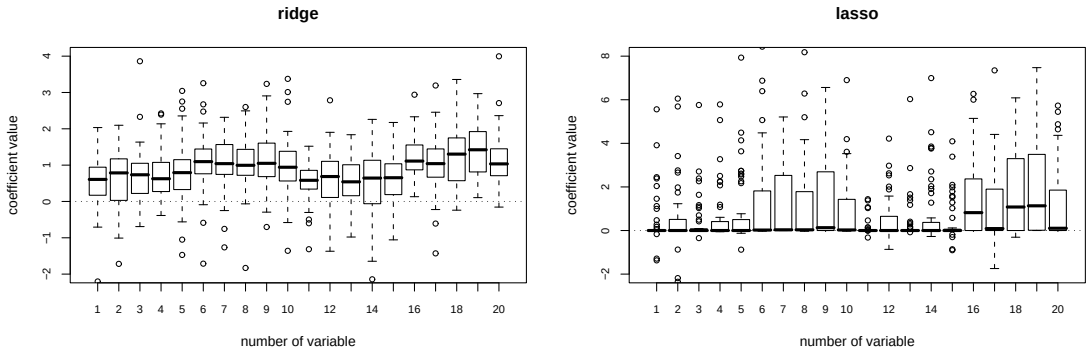

Figure A.9.: Boxplots of the predictors for the third simulation setting and correlation $\rho = 0.5$

Predictor	Method									
	pfl		pfl.kq		pfl.cor		pcor.shrink		pcor.emp	
β_1	0.90	(0.04)	0.92	(0.03)	0.94	(0.04)	0.85	(0.04)	0.80	(0.08)
β_2	0.89	(0.03)	0.92	(0.03)	0.92	(0.02)	0.87	(0.07)	0.85	(0.12)
β_3	0.95	(0.03)	0.93	(0.05)	0.97	(0.03)	0.90	(0.04)	0.87	(0.05)
β_4	0.88	(0.03)	0.88	(0.04)	0.91	(0.03)	0.83	(0.05)	0.74	(0.07)
β_5	0.93	(0.04)	0.93	(0.04)	0.96	(0.03)	0.87	(0.04)	0.82	(0.05)
β_6	0.98	(0.03)	0.99	(0.03)	0.98	(0.02)	0.92	(0.04)	0.90	(0.04)
β_7	0.96	(0.03)	0.96	(0.02)	0.97	(0.03)	0.98	(0.05)	0.95	(0.05)
β_8	0.94	(0.02)	0.96	(0.04)	0.96	(0.02)	0.89	(0.04)	0.85	(0.04)
β_9	0.96	(0.01)	0.96	(0.02)	0.96	(0.02)	0.92	(0.03)	0.89	(0.03)
β_{10}	0.94	(0.02)	0.95	(0.03)	0.95	(0.03)	0.88	(0.04)	0.86	(0.05)
β_{11}	0.89	(0.04)	0.91	(0.04)	0.92	(0.03)	0.77	(0.07)	0.74	(0.09)
β_{12}	0.91	(0.03)	0.90	(0.03)	0.95	(0.03)	0.83	(0.04)	0.80	(0.08)
β_{13}	0.90	(0.05)	0.89	(0.03)	0.93	(0.03)	0.76	(0.07)	0.72	(0.07)
β_{14}	0.86	(0.04)	0.93	(0.03)	0.90	(0.02)	0.79	(0.08)	0.77	(0.08)
β_{15}	0.94	(0.04)	0.93	(0.04)	0.94	(0.04)	0.86	(0.05)	0.82	(0.06)
β_{16}	0.97	(0.04)	0.93	(0.03)	0.94	(0.03)	0.94	(0.03)	0.93	(0.03)
β_{17}	0.97	(0.04)	1.00	(0.04)	0.99	(0.04)	0.93	(0.05)	0.90	(0.03)
β_{18}	1.00	(0.03)	0.99	(0.03)	0.99	(0.03)	0.92	(0.05)	0.97	(0.06)
β_{19}	0.97	(0.05)	0.95	(0.03)	0.97	(0.03)	0.97	(0.06)	0.98	(0.07)
β_{20}	0.99	(0.02)	0.96	(0.03)	0.99	(0.02)	0.92	(0.04)	0.91	(0.03)

Predictor	Method									
	kqpcor.shr		kqpcor.emp		enet		ridge		lasso	
β_1	0.84	(0.04)	0.79	(0.07)	0.55	(0.10)	0.61	(0.08)	0.00	(0.00)
β_2	0.78	(0.12)	0.69	(0.25)	0.56	(0.16)	0.79	(0.16)	0.00	(0.03)
β_3	0.89	(0.06)	0.84	(0.07)	0.57	(0.16)	0.73	(0.14)	0.00	(0.00)
β_4	0.86	(0.08)	0.78	(0.07)	0.54	(0.09)	0.63	(0.08)	0.00	(0.01)
β_5	0.88	(0.05)	0.82	(0.04)	0.78	(0.18)	0.79	(0.10)	0.00	(0.03)
β_6	0.96	(0.03)	0.91	(0.05)	1.01	(0.11)	1.10	(0.08)	0.02	(0.28)
β_7	0.99	(0.06)	0.94	(0.06)	0.98	(0.13)	1.04	(0.11)	0.03	(0.26)
β_8	0.91	(0.04)	0.87	(0.04)	1.01	(0.12)	1.00	(0.10)	0.03	(0.27)
β_9	0.93	(0.03)	0.89	(0.05)	1.04	(0.12)	1.05	(0.09)	0.13	(0.27)
β_{10}	0.88	(0.04)	0.87	(0.04)	0.92	(0.14)	0.94	(0.10)	0.03	(0.08)
β_{11}	0.81	(0.10)	0.75	(0.12)	0.51	(0.07)	0.58	(0.05)	0.00	(0.00)
β_{12}	0.81	(0.08)	0.74	(0.08)	0.62	(0.16)	0.69	(0.09)	0.00	(0.00)
β_{13}	0.74	(0.10)	0.64	(0.11)	0.35	(0.14)	0.54	(0.11)	0.00	(0.00)
β_{14}	0.80	(0.11)	0.73	(0.12)	0.59	(0.12)	0.64	(0.08)	0.00	(0.00)
β_{15}	0.88	(0.06)	0.81	(0.06)	0.57	(0.11)	0.65	(0.07)	0.00	(0.00)
β_{16}	0.96	(0.03)	0.94	(0.04)	1.15	(0.13)	1.11	(0.09)	0.80	(0.53)
β_{17}	0.94	(0.05)	0.89	(0.05)	1.00	(0.12)	1.04	(0.11)	0.09	(0.22)
β_{18}	1.02	(0.07)	1.01	(0.06)	1.25	(0.23)	1.30	(0.18)	1.08	(0.48)
β_{19}	0.98	(0.03)	0.97	(0.04)	1.30	(0.19)	1.42	(0.17)	1.13	(0.55)
β_{20}	0.90	(0.04)	0.89	(0.04)	1.02	(0.07)	1.03	(0.08)	0.10	(0.31)

Table A.4.: Medians of the predictors for the third simulation setting and correlation $\rho = 0.5$ based on 50 replications.

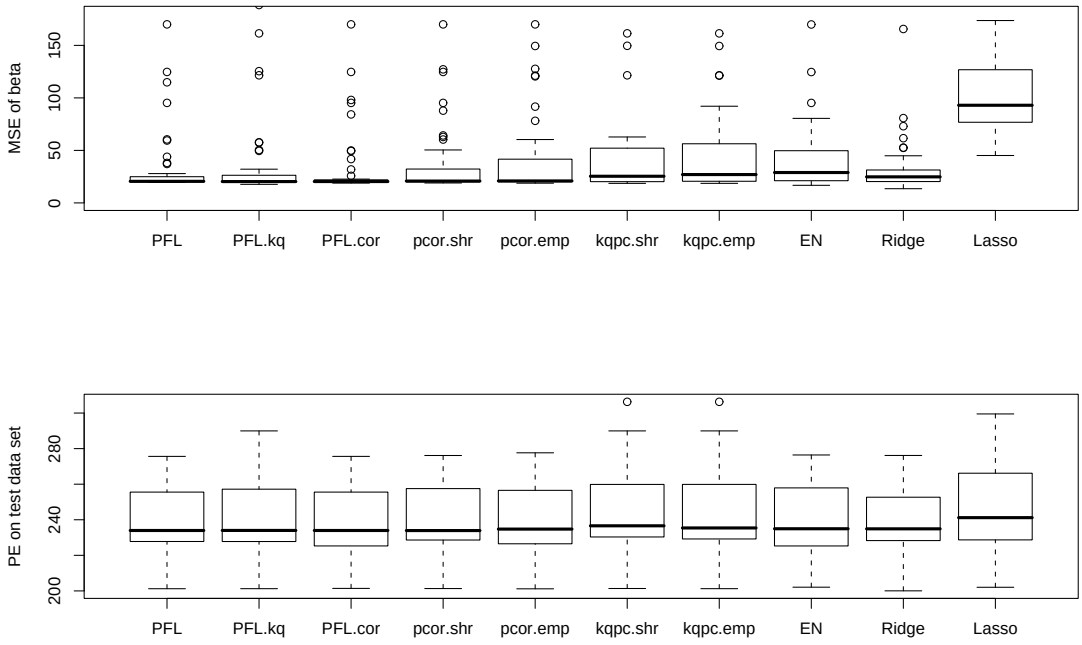

Figure A.10.: Boxplots of the prediction error on the test data set and MSE of β for the third simulation setting and correlation $\rho = 0.9$

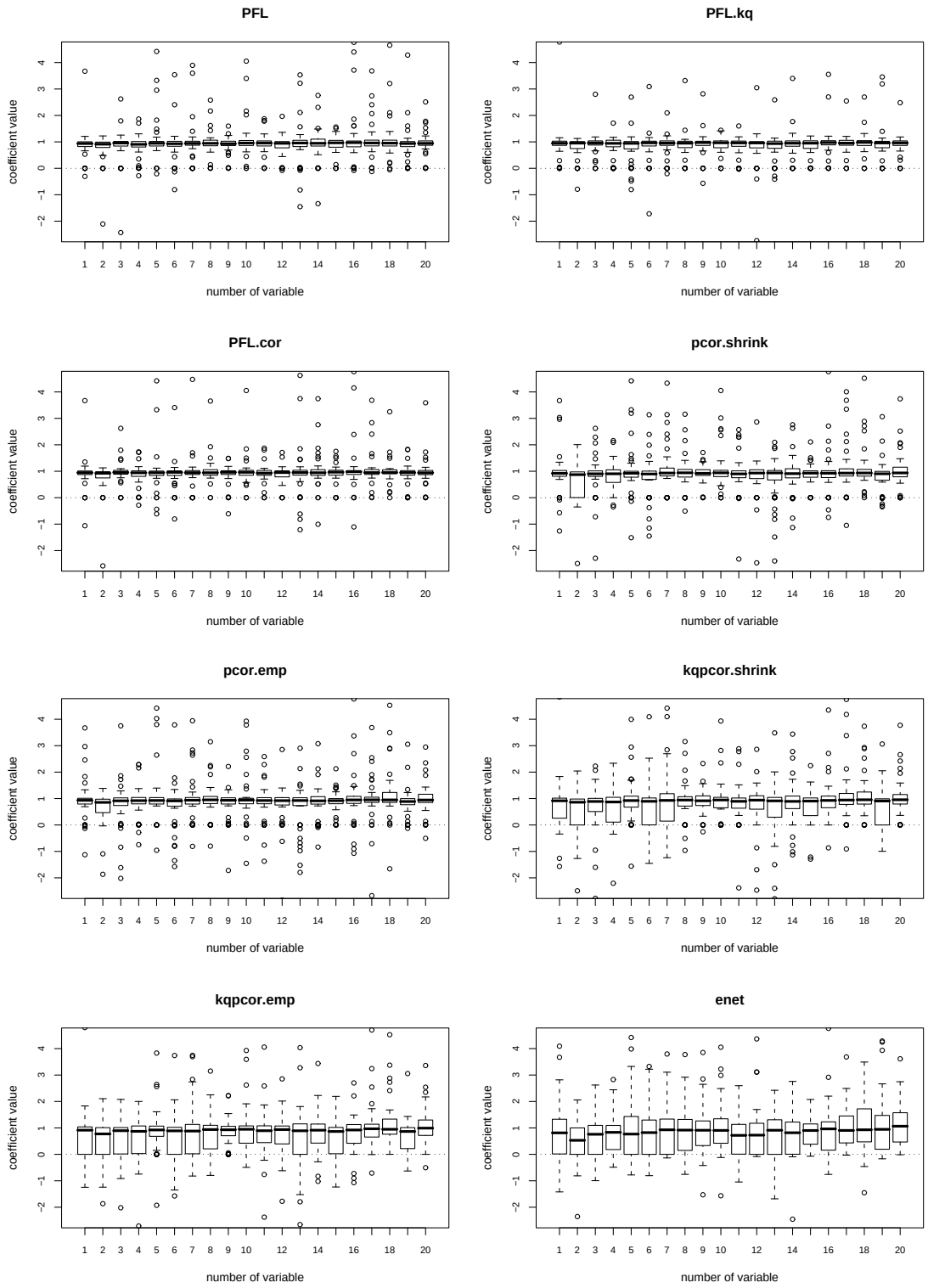

Figure A.11.: Boxplots of the predictors for the third simulation setting and correlation $\rho = 0.9$

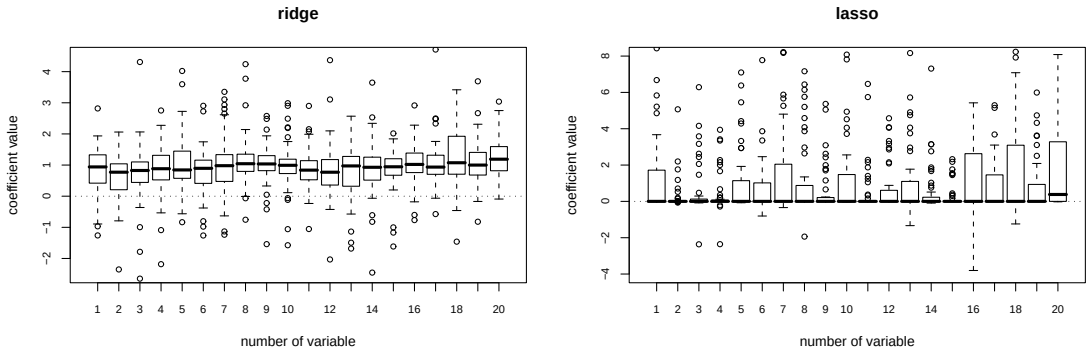

Figure A.12.: Boxplots of the predictors for the third simulation setting and correlation $\rho = 0.9$

Predictor	Method									
	pfl		pfl.kq		pfl.cor		pcor.shrink		pcor.emp	
β_1	0.94	(0.02)	0.95	(0.02)	0.95	(0.02)	0.92	(0.02)	0.93	(0.02)
β_2	0.91	(0.03)	0.95	(0.03)	0.92	(0.03)	0.87	(0.05)	0.85	(0.04)
β_3	0.96	(0.02)	0.96	(0.02)	0.96	(0.02)	0.90	(0.02)	0.91	(0.02)
β_4	0.91	(0.03)	0.94	(0.03)	0.94	(0.03)	0.90	(0.03)	0.92	(0.02)
β_5	0.95	(0.02)	0.94	(0.02)	0.94	(0.02)	0.92	(0.02)	0.93	(0.02)
β_6	0.93	(0.03)	0.96	(0.02)	0.96	(0.02)	0.90	(0.04)	0.91	(0.03)
β_7	0.94	(0.02)	0.95	(0.02)	0.95	(0.02)	0.93	(0.03)	0.94	(0.02)
β_8	0.95	(0.02)	0.95	(0.02)	0.95	(0.02)	0.94	(0.03)	0.95	(0.03)
β_9	0.93	(0.03)	0.97	(0.03)	0.96	(0.03)	0.92	(0.03)	0.93	(0.02)
β_{10}	0.96	(0.02)	0.96	(0.02)	0.96	(0.01)	0.94	(0.03)	0.94	(0.02)
β_{11}	0.95	(0.03)	0.96	(0.03)	0.93	(0.03)	0.90	(0.03)	0.93	(0.03)
β_{12}	0.95	(0.03)	0.96	(0.03)	0.95	(0.02)	0.93	(0.03)	0.92	(0.02)
β_{13}	0.96	(0.03)	0.94	(0.02)	0.95	(0.02)	0.94	(0.04)	0.93	(0.03)
β_{14}	0.95	(0.02)	0.95	(0.03)	0.94	(0.02)	0.91	(0.04)	0.91	(0.03)
β_{15}	0.96	(0.02)	0.95	(0.03)	0.96	(0.03)	0.93	(0.02)	0.92	(0.03)
β_{16}	0.97	(0.02)	0.97	(0.02)	0.98	(0.02)	0.93	(0.02)	0.95	(0.02)
β_{17}	0.96	(0.03)	0.95	(0.03)	0.95	(0.03)	0.93	(0.03)	0.95	(0.02)
β_{18}	0.95	(0.02)	0.98	(0.03)	0.95	(0.02)	0.93	(0.02)	0.95	(0.02)
β_{19}	0.94	(0.03)	0.96	(0.03)	0.95	(0.03)	0.90	(0.04)	0.89	(0.03)
β_{20}	0.94	(0.02)	0.96	(0.02)	0.94	(0.01)	0.94	(0.02)	0.94	(0.02)

Predictor	Method									
	kqpcor.shr		kqpcor.emp		enet		ridge		lasso	
β_1	0.92	(0.05)	0.91	(0.05)	0.81	(0.20)	0.94	(0.12)	0.00	(0.11)
β_2	0.86	(0.08)	0.77	(0.13)	0.53	(0.21)	0.77	(0.09)	0.00	(0.00)
β_3	0.89	(0.04)	0.89	(0.06)	0.76	(0.11)	0.83	(0.08)	0.00	(0.00)
β_4	0.87	(0.07)	0.87	(0.06)	0.83	(0.09)	0.88	(0.04)	0.00	(0.00)
β_5	0.92	(0.03)	0.92	(0.03)	0.77	(0.11)	0.84	(0.12)	0.00	(0.01)
β_6	0.90	(0.05)	0.89	(0.06)	0.82	(0.17)	0.90	(0.11)	0.00	(0.03)
β_7	0.93	(0.05)	0.88	(0.07)	0.93	(0.13)	0.98	(0.08)	0.00	(0.02)
β_8	0.94	(0.03)	0.94	(0.06)	0.92	(0.08)	1.04	(0.08)	0.00	(0.10)
β_9	0.92	(0.03)	0.93	(0.04)	0.91	(0.07)	1.04	(0.09)	0.00	(0.01)
β_{10}	0.95	(0.04)	0.95	(0.04)	0.91	(0.09)	0.99	(0.07)	0.00	(0.00)
β_{11}	0.89	(0.04)	0.89	(0.05)	0.72	(0.12)	0.84	(0.09)	0.00	(0.00)
β_{12}	0.94	(0.04)	0.94	(0.04)	0.73	(0.12)	0.77	(0.10)	0.00	(0.00)
β_{13}	0.91	(0.05)	0.89	(0.08)	0.91	(0.13)	0.97	(0.08)	0.00	(0.01)
β_{14}	0.89	(0.04)	0.91	(0.06)	0.82	(0.12)	0.93	(0.08)	0.00	(0.01)
β_{15}	0.90	(0.05)	0.87	(0.06)	0.90	(0.08)	0.94	(0.05)	0.00	(0.00)
β_{16}	0.93	(0.03)	0.92	(0.04)	0.96	(0.10)	1.02	(0.07)	0.00	(0.03)
β_{17}	0.94	(0.03)	0.97	(0.04)	0.90	(0.08)	0.93	(0.06)	0.00	(0.01)
β_{18}	0.95	(0.04)	0.95	(0.04)	0.93	(0.13)	1.07	(0.11)	0.00	(0.20)
β_{19}	0.91	(0.05)	0.87	(0.07)	0.95	(0.09)	1.00	(0.06)	0.00	(0.01)
β_{20}	0.95	(0.04)	0.99	(0.06)	1.06	(0.13)	1.19	(0.11)	0.38	(0.38)

Table A.5.: Medians of the predictors for the third simulation setting and correlation $\rho = 0.9$ based on 50 replications.

A.4. Setting 4

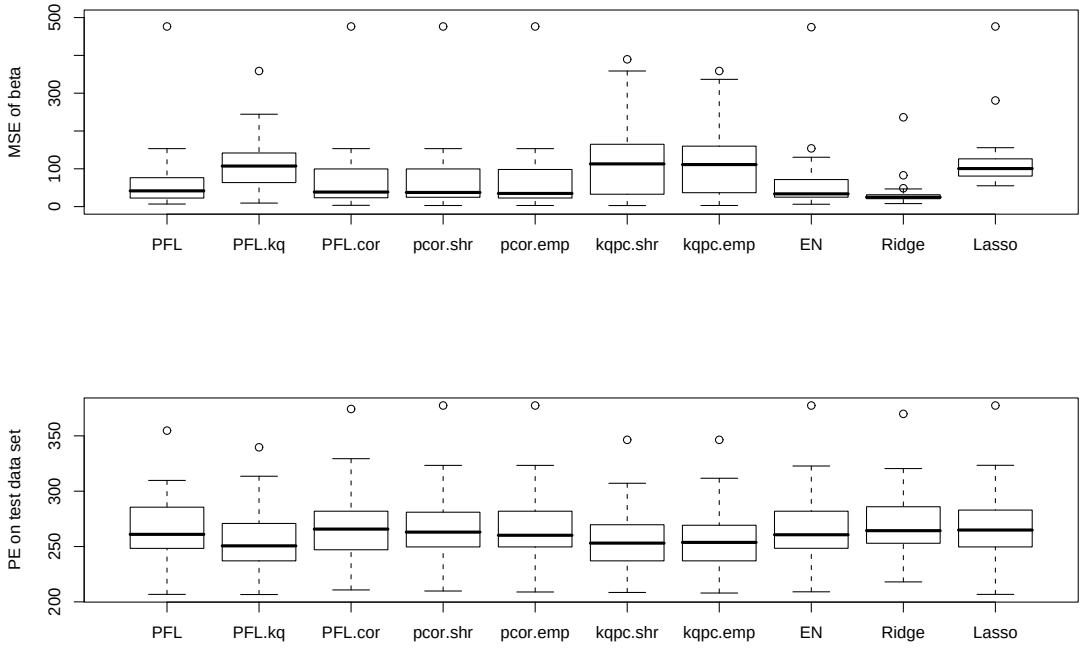

Figure A.13.: Boxplots of the prediction error on the test data set and MSE of β for the fourth simulation setting

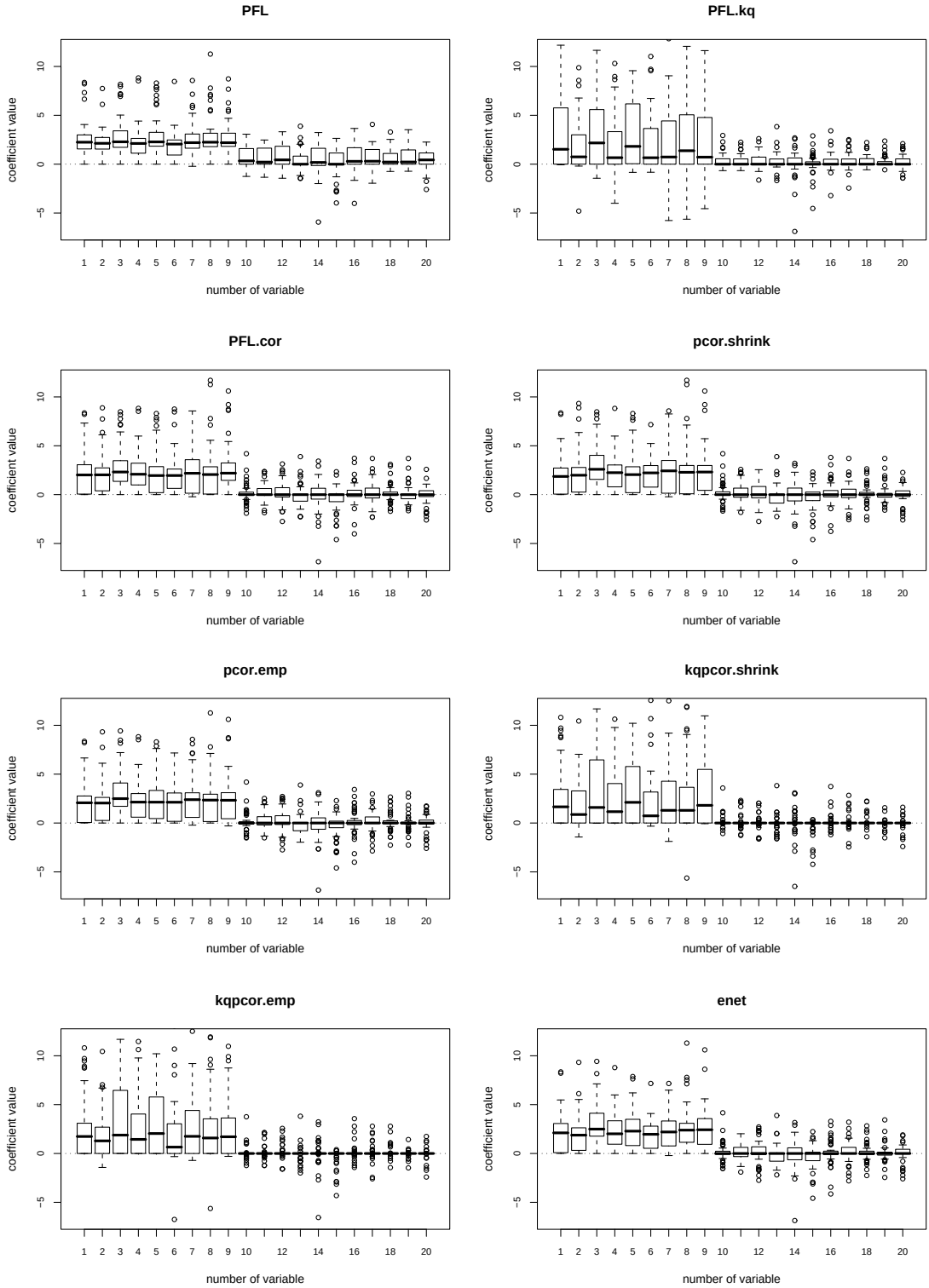

Figure A.14.: Boxplots of the predictors for the fourth simulation setting

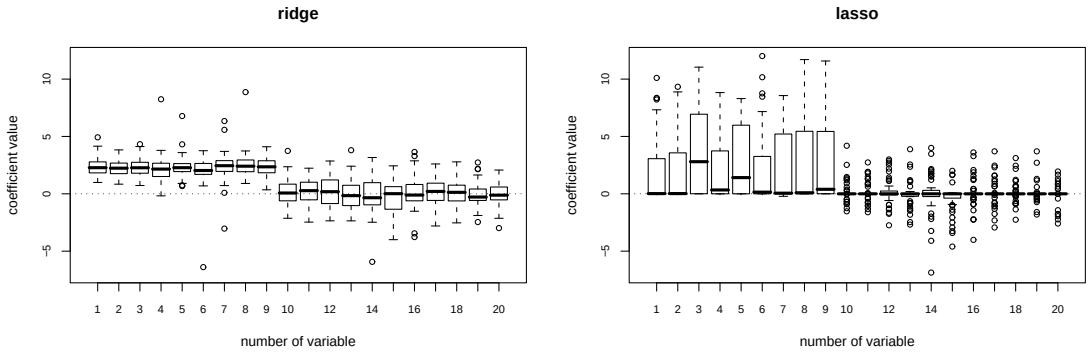

Figure A.15.: Boxplots of the predictors for the fourth simulation setting

Predictor	Method									
	pfl		pfl.kq		pfl.cor		pcor.shrink		pcor.emp	
β_1	2.25	(0.13)	1.52	(0.76)	2.02	(0.23)	1.86	(0.31)	2.07	(0.25)
β_2	2.13	(0.14)	0.75	(0.58)	2.03	(0.19)	1.99	(0.30)	2.04	(0.23)
β_3	2.29	(0.18)	2.17	(0.75)	2.32	(0.23)	2.59	(0.28)	2.50	(0.24)
β_4	2.12	(0.16)	0.66	(0.36)	2.09	(0.26)	2.25	(0.28)	2.14	(0.30)
β_5	2.28	(0.14)	1.82	(0.94)	1.95	(0.28)	2.05	(0.34)	2.13	(0.24)
β_6	2.05	(0.17)	0.66	(0.69)	1.96	(0.28)	2.23	(0.21)	2.13	(0.23)
β_7	2.20	(0.12)	0.73	(0.59)	2.19	(0.23)	2.43	(0.19)	2.39	(0.16)
β_8	2.25	(0.11)	1.38	(0.74)	2.06	(0.28)	2.29	(0.27)	2.34	(0.24)
β_9	2.20	(0.12)	0.72	(0.73)	2.20	(0.23)	2.31	(0.23)	2.33	(0.24)
β_{10}	0.34	(0.23)	0.00	(0.01)	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
β_{11}	0.21	(0.31)	0.00	(0.03)	0.00	(0.03)	0.00	(0.03)	0.00	(0.04)
β_{12}	0.44	(0.24)	0.00	(0.07)	0.00	(0.07)	0.00	(0.07)	0.00	(0.09)
β_{13}	0.00	(0.10)	0.00	(0.01)	0.00	(0.04)	0.00	(0.07)	0.00	(0.11)
β_{14}	0.18	(0.26)	0.00	(0.04)	0.00	(0.04)	0.00	(0.04)	0.00	(0.08)
β_{15}	0.00	(0.14)	0.00	(0.01)	0.00	(0.02)	0.00	(0.03)	0.00	(0.03)
β_{16}	0.29	(0.23)	0.00	(0.01)	0.00	(0.03)	0.00	(0.01)	0.00	(0.01)
β_{17}	0.30	(0.24)	0.00	(0.03)	0.00	(0.04)	0.00	(0.06)	0.00	(0.09)
β_{18}	0.22	(0.16)	0.00	(0.02)	0.00	(0.02)	0.00	(0.01)	0.00	(0.01)
β_{19}	0.22	(0.20)	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)	0.00	(0.01)
β_{20}	0.44	(0.18)	0.00	(0.04)	0.00	(0.01)	0.00	(0.01)	0.00	(0.00)

Predictor	Method									
	kqpcor.shr		kqpcor.emp		enet		ridge		lasso	
β_1	1.66	(0.80)	1.74	(0.78)	2.11	(0.23)	2.28	(0.15)	0.02	(0.10)
β_2	0.87	(0.62)	1.29	(0.63)	1.88	(0.34)	2.24	(0.07)	0.02	(0.29)
β_3	1.59	(0.60)	1.88	(0.54)	2.50	(0.38)	2.27	(0.14)	2.80	(1.12)
β_4	1.16	(0.80)	1.44	(0.89)	2.01	(0.38)	2.16	(0.13)	0.34	(0.47)
β_5	2.12	(0.73)	2.04	(0.67)	2.29	(0.32)	2.28	(0.12)	1.41	(1.30)
β_6	0.74	(0.78)	0.65	(0.85)	1.96	(0.19)	2.03	(0.11)	0.15	(0.41)
β_7	1.30	(0.93)	1.75	(0.84)	2.20	(0.22)	2.45	(0.09)	0.06	(0.85)
β_8	1.29	(0.81)	1.58	(0.79)	2.39	(0.31)	2.41	(0.14)	0.09	(0.20)
β_9	1.81	(0.83)	1.71	(0.85)	2.43	(0.25)	2.35	(0.12)	0.39	(0.93)
β_{10}	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)	0.07	(0.20)	0.00	(0.00)
β_{11}	0.00	(0.00)	0.00	(0.00)	0.00	(0.04)	0.27	(0.20)	0.00	(0.00)
β_{12}	0.00	(0.00)	0.00	(0.00)	0.00	(0.06)	0.19	(0.20)	0.00	(0.00)
β_{13}	0.00	(0.00)	0.00	(0.00)	0.00	(0.07)	-0.16	(0.21)	0.00	(0.00)
β_{14}	0.00	(0.00)	0.00	(0.00)	0.00	(0.03)	-0.34	(0.36)	0.00	(0.00)
β_{15}	0.00	(0.00)	0.00	(0.00)	0.00	(0.05)	0.01	(0.29)	0.00	(0.00)
β_{16}	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)	-0.11	(0.20)	0.00	(0.00)
β_{17}	0.00	(0.00)	0.00	(0.00)	0.00	(0.04)	0.21	(0.22)	0.00	(0.00)
β_{18}	0.00	(0.00)	0.00	(0.00)	0.00	(0.01)	0.13	(0.21)	0.00	(0.00)
β_{19}	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)	-0.28	(0.15)	0.00	(0.00)
β_{20}	0.00	(0.00)	0.00	(0.00)	0.00	(0.01)	-0.12	(0.22)	0.00	(0.00)

Table A.6.: Medians of the predictors for the fourth simulation setting based on 50 replications.

A.5. Setting 5 and 6

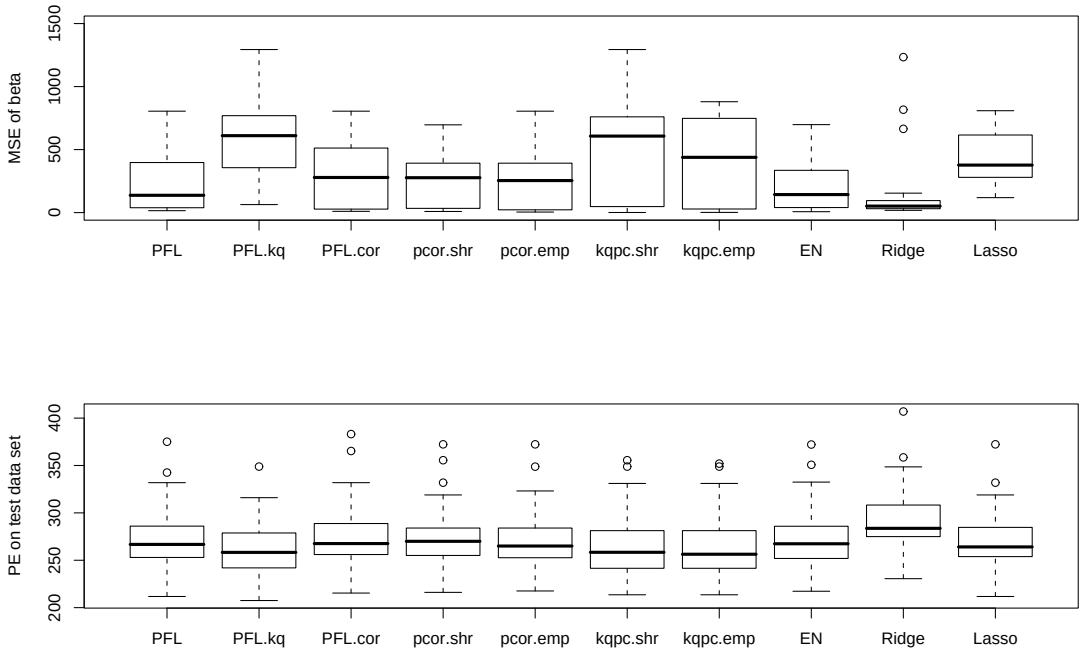

Figure A.16.: Boxplots of the prediction error on the test data set and MSE of β for the sixth simulation setting

Predictor	Method									
	pfl		pfl.kq		pfl.cor		pcor.shrink		pcor.emp	
β_1	3.70	(0.43)	0.12	(1.05)	3.82	(0.61)	3.40	(0.68)	3.78	(0.53)
β_2	1.07	(0.29)	0.03	(0.19)	0.44	(0.28)	0.69	(0.34)	0.80	(0.28)
β_3	0.10	(0.27)	0.00	(0.04)	0.00	(0.01)	0.00	(0.02)	0.00	(0.03)
β_4	0.51	(0.22)	0.00	(0.05)	0.00	(0.05)	0.00	(0.04)	0.00	(0.03)
β_5	0.39	(0.14)	0.00	(0.04)	0.00	(0.07)	0.00	(0.04)	0.00	(0.04)
β_6	8.93	(0.34)	5.56	(5.01)	9.25	(0.56)	9.41	(0.33)	9.36	(0.25)
β_7	0.07	(0.23)	0.00	(0.05)	0.00	(0.00)	0.00	(0.01)	0.00	(0.00)
β_8	9.19	(0.58)	12.43	(5.22)	9.56	(0.99)	9.84	(0.67)	9.84	(0.66)
β_9	1.51	(0.22)	0.63	(0.52)	1.03	(0.48)	1.09	(0.46)	1.03	(0.34)
β_{10}	0.10	(0.26)	0.00	(0.01)	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
β_{11}	0.01	(0.15)	0.00	(0.04)	0.00	(0.00)	0.00	(0.01)	0.00	(0.01)
β_{12}	0.02	(0.24)	0.00	(0.01)	0.00	(0.10)	0.00	(0.09)	0.00	(0.06)
β_{13}	0.01	(0.20)	0.00	(0.01)	0.00	(0.00)	0.00	(0.03)	0.00	(0.02)
β_{14}	3.77	(0.47)	1.27	(1.52)	3.81	(0.74)	4.18	(0.91)	4.13	(0.37)
β_{15}	8.39	(0.85)	0.38	(0.80)	6.73	(1.22)	8.42	(1.69)	8.04	(1.65)
β_{16}	0.96	(0.27)	0.75	(0.39)	0.40	(0.20)	0.61	(0.24)	0.63	(0.17)
β_{17}	0.10	(0.18)	0.00	(0.06)	0.00	(0.05)	0.00	(0.07)	0.00	(0.02)
β_{18}	0.00	(0.16)	0.00	(0.04)	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
β_{19}	4.09	(0.39)	0.90	(1.06)	3.80	(0.81)	3.58	(0.84)	4.12	(0.57)
β_{20}	0.01	(0.24)	0.00	(0.03)	0.00	(0.02)	0.00	(0.00)	0.00	(0.01)

Predictor	Method									
	kqpcor.shr		kqpcor.emp		enet		ridge		lasso	
β_1	2.46	(1.29)	3.71	(1.07)	3.51	(0.46)	4.39	(0.19)	0.19	(0.71)
β_2	0.47	(0.32)	0.62	(0.37)	0.48	(0.35)	1.63	(0.22)	0.00	(0.01)
β_3	0.00	(0.00)	0.00	(0.00)	0.00	(0.01)	0.37	(0.38)	0.00	(0.00)
β_4	0.00	(0.00)	0.00	(0.00)	0.00	(0.02)	0.14	(0.35)	0.00	(0.00)
β_5	0.00	(0.00)	0.00	(0.00)	0.00	(0.02)	0.49	(0.29)	0.00	(0.00)
β_6	9.30	(1.26)	9.29	(0.76)	9.26	(0.48)	9.36	(0.27)	6.62	(3.05)
β_7	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)	0.02	(0.22)	0.00	(0.00)
β_8	9.90	(1.63)	10.04	(1.24)	10.05	(0.76)	9.06	(0.20)	11.30	(3.53)
β_9	0.68	(0.43)	0.90	(0.42)	1.12	(0.53)	1.82	(0.21)	0.02	(0.75)
β_{10}	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)	0.16	(0.27)	0.00	(0.00)
β_{11}	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)	-0.09	(0.47)	0.00	(0.00)
β_{12}	0.00	(0.00)	0.00	(0.00)	0.00	(0.07)	0.37	(0.25)	0.00	(0.01)
β_{13}	0.00	(0.00)	0.00	(0.00)	0.00	(0.01)	-0.12	(0.35)	0.00	(0.00)
β_{14}	3.73	(1.04)	4.12	(0.41)	3.69	(0.82)	4.40	(0.24)	0.68	(2.48)
β_{15}	0.14	(2.65)	3.90	(3.16)	8.83	(0.73)	8.96	(0.19)	3.82	(1.77)
β_{16}	0.69	(0.34)	1.08	(0.31)	0.66	(0.24)	1.74	(0.16)	0.02	(0.16)
β_{17}	0.00	(0.00)	0.00	(0.00)	0.00	(0.02)	0.05	(0.45)	0.00	(0.01)
β_{18}	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)	0.10	(0.28)	0.00	(0.00)
β_{19}	2.85	(0.98)	3.80	(0.90)	3.90	(0.51)	4.31	(0.15)	0.30	(1.01)
β_{20}	0.00	(0.00)	0.00	(0.01)	0.00	(0.00)	-0.09	(0.23)	0.00	(0.00)

Table A.7.: Medians of the predictors for the sixth simulation setting based on 50 replications.

Predictor	Method									
	pfl		pfl.kq		pfl.cor		pcor.shrink		pcor.emp	
β_1	3.74	(0.41)	1.30	(1.64)	4.06	(0.30)	3.66	(0.44)	3.80	(0.29)
β_2	3.81	(0.34)	0.74	(0.55)	4.14	(0.26)	3.74	(0.45)	4.00	(0.25)
β_3	4.29	(0.42)	6.60	(3.11)	4.40	(0.29)	4.30	(0.40)	4.32	(0.20)
β_4	1.33	(0.31)	0.41	(0.24)	1.08	(0.35)	0.91	(0.29)	0.88	(0.29)
β_5	1.50	(0.20)	1.35	(0.60)	1.09	(0.29)	1.12	(0.24)	1.19	(0.22)
β_6	1.03	(0.25)	0.57	(0.27)	1.00	(0.24)	0.85	(0.19)	0.93	(0.25)
β_7	8.93	(0.75)	6.26	(3.68)	9.30	(0.26)	9.23	(0.37)	9.19	(0.19)
β_8	8.89	(0.66)	3.69	(3.94)	9.09	(0.33)	9.08	(0.38)	9.12	(0.26)
β_9	9.11	(0.60)	0.90	(1.38)	9.56	(0.31)	9.51	(0.50)	9.31	(0.34)
β_{10}	0.00	(0.16)	0.00	(0.10)	0.00	(0.00)	0.00	(0.01)	0.00	(0.00)
β_{11}	0.00	(0.13)	0.00	(0.12)	0.00	(0.02)	0.00	(0.02)	0.00	(0.01)
β_{12}	0.00	(0.21)	0.01	(0.19)	0.00	(0.07)	0.00	(0.04)	0.00	(0.02)
β_{13}	0.00	(0.02)	0.00	(0.06)	0.00	(0.03)	0.00	(0.04)	0.00	(0.04)
β_{14}	0.00	(0.16)	0.00	(0.17)	0.00	(0.05)	0.00	(0.05)	0.00	(0.06)
β_{15}	0.00	(0.05)	0.00	(0.04)	0.00	(0.03)	0.00	(0.03)	0.00	(0.03)
β_{16}	0.12	(0.26)	0.00	(0.14)	0.00	(0.02)	0.00	(0.02)	0.00	(0.02)
β_{17}	0.00	(0.16)	0.01	(0.17)	0.00	(0.06)	0.00	(0.03)	0.00	(0.04)
β_{18}	0.03	(0.11)	0.00	(0.17)	0.00	(0.04)	0.00	(0.02)	0.00	(0.02)
β_{19}	0.00	(0.06)	0.00	(0.07)	0.00	(0.01)	0.00	(0.00)	0.00	(0.01)
β_{20}	0.00	(0.16)	0.00	(0.16)	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)

Predictor	Method									
	kqpcor.shr		kqpcor.emp		enet		ridge		lasso	
β_1	4.26	(0.43)	3.96	(0.63)	3.88	(0.45)	4.49	(0.24)	0.37	(0.50)
β_2	3.42	(0.56)	3.36	(0.87)	3.61	(0.56)	4.37	(0.25)	0.40	(1.01)
β_3	4.31	(0.36)	4.40	(0.37)	4.59	(0.47)	4.40	(0.22)	6.22	(2.09)
β_4	0.55	(0.30)	0.60	(0.37)	0.67	(0.41)	1.60	(0.19)	0.01	(0.13)
β_5	1.36	(0.37)	1.05	(0.45)	1.00	(0.34)	1.76	(0.16)	0.81	(0.39)
β_6	0.76	(0.25)	0.60	(0.31)	0.76	(0.22)	1.56	(0.17)	0.03	(0.14)
β_7	9.03	(0.92)	9.28	(1.07)	9.25	(0.50)	9.02	(0.13)	4.98	(3.57)
β_8	8.97	(0.66)	9.06	(1.96)	9.04	(0.48)	8.87	(0.25)	6.56	(2.14)
β_9	8.89	(0.67)	9.13	(1.39)	9.47	(0.81)	8.99	(0.18)	7.25	(3.11)
β_{10}	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)	-0.01	(0.24)	0.00	(0.00)
β_{11}	0.00	(0.00)	0.00	(0.00)	0.00	(0.02)	0.28	(0.27)	0.00	(0.00)
β_{12}	0.00	(0.00)	0.00	(0.00)	0.00	(0.01)	0.34	(0.30)	0.00	(0.00)
β_{13}	0.00	(0.00)	0.00	(0.00)	0.00	(0.02)	-0.34	(0.32)	0.00	(0.00)
β_{14}	0.00	(0.00)	0.00	(0.00)	0.00	(0.01)	-0.48	(0.53)	0.00	(0.00)
β_{15}	0.00	(0.00)	0.00	(0.00)	0.00	(0.04)	0.01	(0.36)	0.00	(0.01)
β_{16}	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)	0.03	(0.29)	0.00	(0.00)
β_{17}	0.00	(0.00)	0.00	(0.00)	0.00	(0.01)	0.40	(0.39)	0.00	(0.00)
β_{18}	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)	0.39	(0.28)	0.00	(0.00)
β_{19}	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)	-0.42	(0.15)	0.00	(0.00)
β_{20}	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)	-0.03	(0.25)	0.00	(0.00)

Table A.8.: Medians of the predictors for the fifth simulation setting based on 50 replications.

A.6. Setting 7

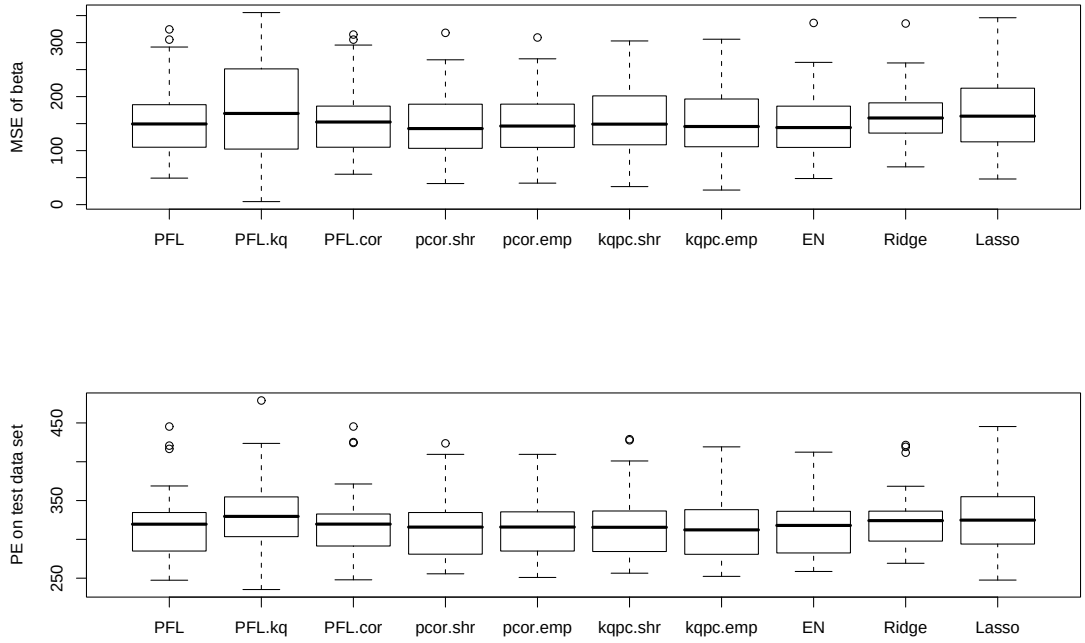

Figure A.17.: Boxplots of the prediction error on the test data set and MSE of β for the seventh simulation setting and correlation $\rho = 0.5$

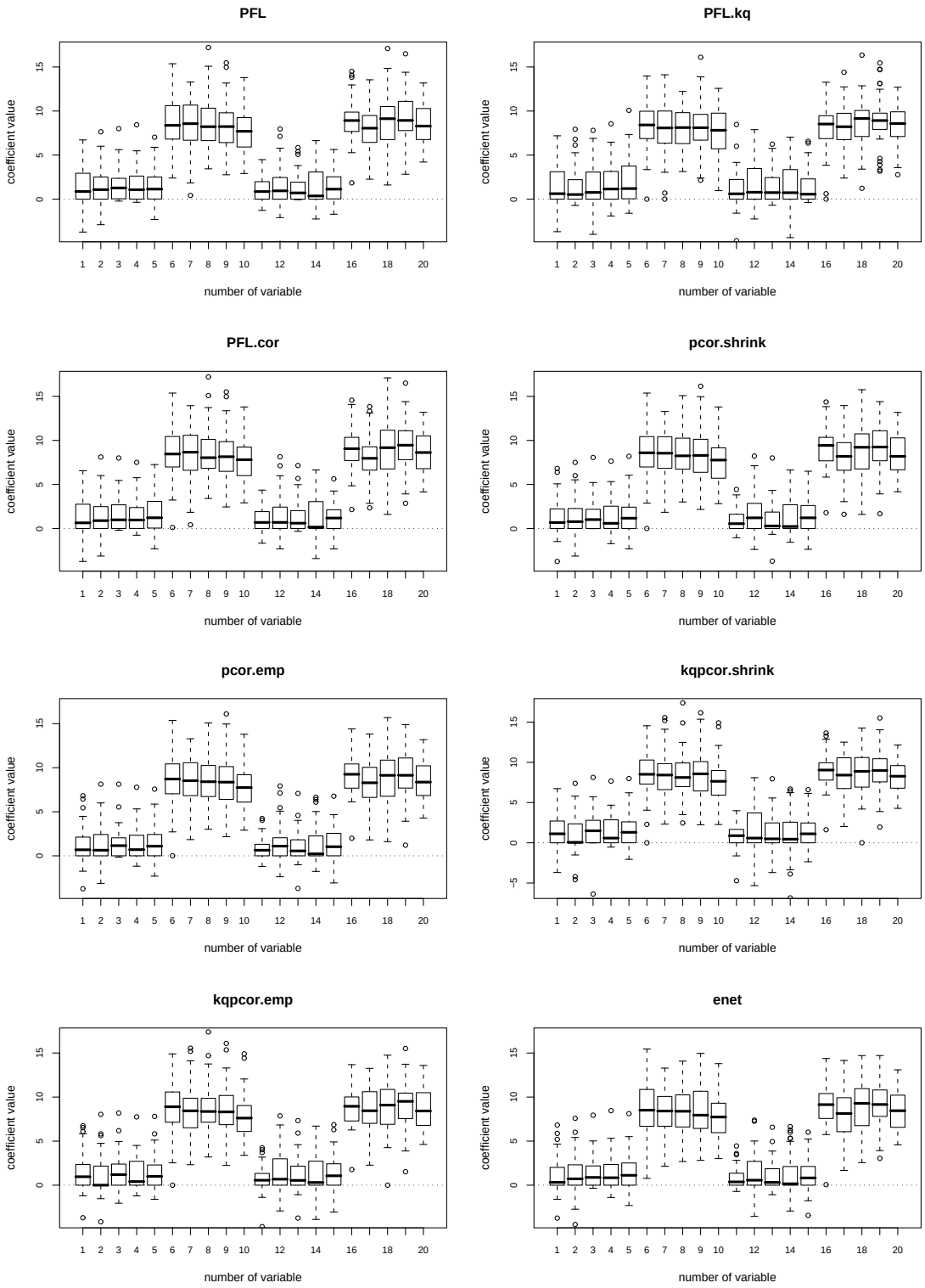

Figure A.18.: Boxplots of the predictors for the seventh simulation setting and correlation $\rho = 0.5$

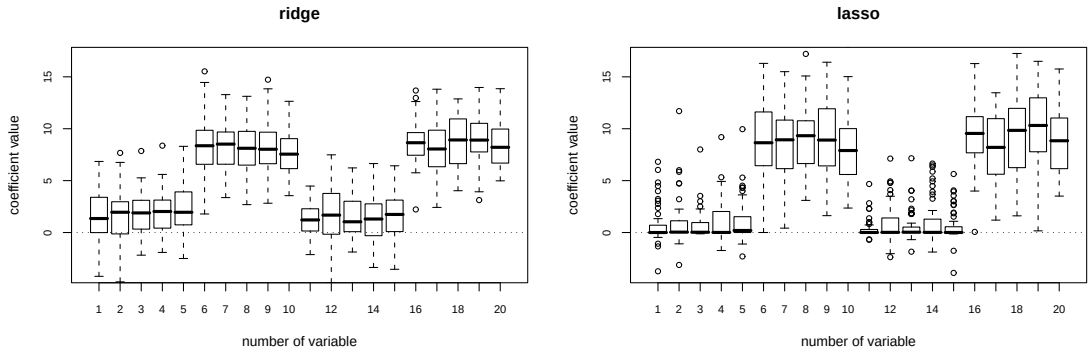

Figure A.19.: Boxplots of the predictors for the seventh simulation setting and correlation $\rho = 0.5$

Predictor	Method									
	pfl		pfl.kq		pfl.cor		pcor.shrink		pcor.emp	
β_1	0.87	(0.41)	0.62	(0.36)	0.65	(0.48)	0.68	(0.44)	0.68	(0.36)
β_2	1.08	(0.69)	0.52	(0.34)	0.89	(0.49)	0.78	(0.57)	0.64	(0.52)
β_3	1.27	(0.45)	0.76	(0.37)	0.99	(0.50)	1.01	(0.47)	1.16	(0.25)
β_4	1.07	(0.42)	1.15	(0.44)	0.97	(0.31)	0.59	(0.44)	0.70	(0.33)
β_5	1.14	(0.30)	1.20	(0.50)	1.22	(0.39)	1.17	(0.36)	1.09	(0.25)
β_6	8.37	(0.26)	8.42	(0.40)	8.46	(0.28)	8.59	(0.33)	8.72	(0.31)
β_7	8.57	(0.48)	8.08	(0.37)	8.66	(0.44)	8.55	(0.40)	8.52	(0.42)
β_8	8.22	(0.42)	8.12	(0.46)	8.04	(0.46)	8.25	(0.37)	8.41	(0.44)
β_9	8.23	(0.46)	8.10	(0.35)	8.15	(0.48)	8.30	(0.39)	8.36	(0.47)
β_{10}	7.70	(0.56)	7.82	(0.62)	7.81	(0.50)	7.77	(0.31)	7.75	(0.31)
β_{11}	0.86	(0.46)	0.61	(0.29)	0.69	(0.33)	0.55	(0.35)	0.63	(0.32)
β_{12}	0.94	(0.34)	0.78	(0.58)	0.69	(0.38)	1.21	(0.42)	1.10	(0.38)
β_{13}	0.71	(0.34)	0.75	(0.30)	0.61	(0.34)	0.29	(0.25)	0.56	(0.27)
β_{14}	0.37	(0.47)	0.73	(0.55)	0.16	(0.46)	0.23	(0.33)	0.23	(0.31)
β_{15}	1.14	(0.46)	0.56	(0.43)	1.19	(0.31)	1.21	(0.29)	1.03	(0.27)
β_{16}	8.92	(0.27)	8.52	(0.25)	9.06	(0.27)	9.43	(0.36)	9.25	(0.44)
β_{17}	8.05	(0.33)	8.22	(0.53)	7.97	(0.38)	8.19	(0.35)	8.29	(0.38)
β_{18}	9.13	(0.61)	9.15	(0.25)	9.17	(0.59)	9.22	(0.52)	9.13	(0.59)
β_{19}	8.93	(0.36)	8.92	(0.21)	9.46	(0.44)	9.25	(0.50)	9.14	(0.47)
β_{20}	8.29	(0.35)	8.58	(0.47)	8.62	(0.34)	8.19	(0.38)	8.36	(0.35)

Predictor	Method									
	kqpcor.shr		kqpcor.emp		enet		ridge		lasso	
β_1	1.12	(0.43)	0.96	(0.35)	0.33	(0.35)	1.35	(0.36)	0.00	(0.05)
β_2	0.06	(0.21)	0.00	(0.20)	0.72	(0.52)	1.95	(0.38)	0.04	(0.07)
β_3	1.49	(0.31)	1.20	(0.34)	0.88	(0.35)	1.89	(0.43)	0.03	(0.08)
β_4	0.58	(0.53)	0.41	(0.48)	0.85	(0.31)	2.03	(0.40)	0.01	(0.13)
β_5	1.30	(0.54)	1.01	(0.36)	1.12	(0.39)	1.95	(0.61)	0.20	(0.15)
β_6	8.51	(0.36)	8.90	(0.35)	8.51	(0.40)	8.36	(0.45)	8.65	(0.56)
β_7	8.42	(0.29)	8.44	(0.31)	8.42	(0.53)	8.52	(0.51)	8.93	(0.62)
β_8	8.12	(0.47)	8.36	(0.42)	8.39	(0.54)	8.12	(0.53)	9.33	(0.61)
β_9	8.57	(0.31)	8.32	(0.35)	7.95	(0.56)	8.02	(0.28)	8.91	(0.65)
β_{10}	7.64	(0.43)	7.61	(0.41)	7.74	(0.33)	7.55	(0.37)	7.90	(0.52)
β_{11}	0.88	(0.37)	0.56	(0.30)	0.37	(0.17)	1.21	(0.32)	0.00	(0.01)
β_{12}	0.58	(0.54)	0.68	(0.45)	0.57	(0.46)	1.68	(0.39)	0.01	(0.07)
β_{13}	0.49	(0.37)	0.54	(0.33)	0.32	(0.26)	1.04	(0.36)	0.03	(0.06)
β_{14}	0.45	(0.39)	0.31	(0.43)	0.16	(0.19)	1.30	(0.31)	0.01	(0.06)
β_{15}	1.12	(0.61)	1.07	(0.52)	0.83	(0.33)	1.74	(0.34)	0.00	(0.04)
β_{16}	9.04	(0.25)	8.95	(0.29)	9.14	(0.37)	8.65	(0.29)	9.55	(0.46)
β_{17}	8.41	(0.24)	8.45	(0.29)	8.14	(0.33)	8.05	(0.47)	8.20	(0.49)
β_{18}	8.88	(0.46)	9.09	(0.54)	9.28	(0.71)	8.92	(0.58)	9.84	(0.72)
β_{19}	8.99	(0.35)	9.51	(0.37)	9.16	(0.49)	8.91	(0.30)	10.32	(0.60)
β_{20}	8.27	(0.40)	8.42	(0.46)	8.45	(0.49)	8.21	(0.41)	8.84	(0.53)

Table A.9.: Medians of the predictors for the seventh simulation setting and correlation $\rho = 0.5$ based on 50 replications.

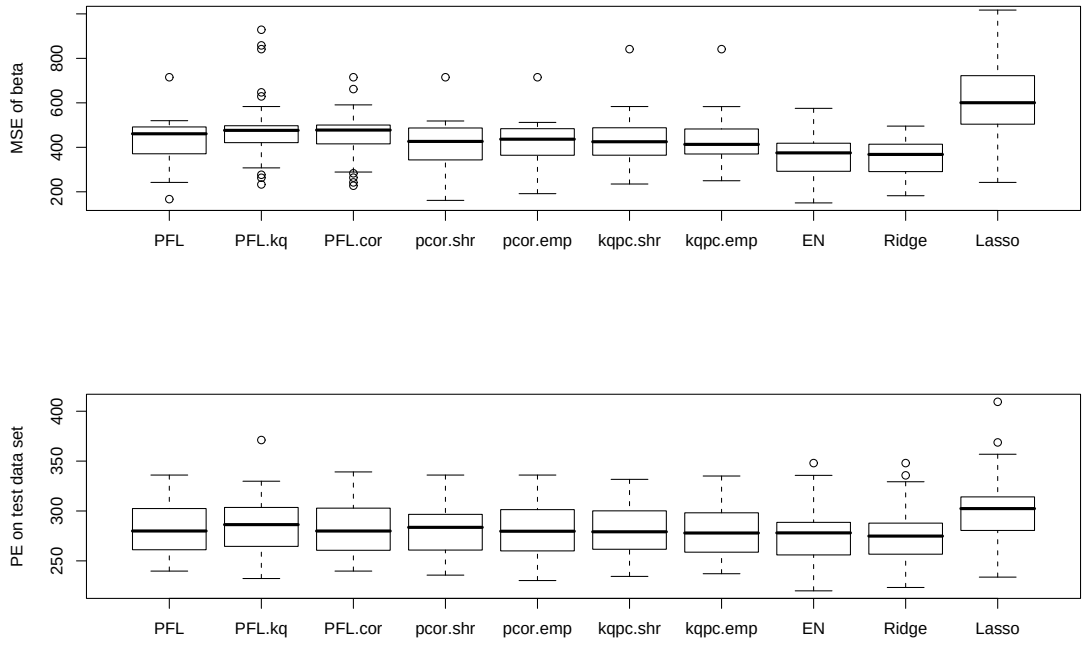

Figure A.20.: Boxplots of the prediction error on the test data set and MSE of β for the seventh simulation setting and correlation $\rho = 0.9$

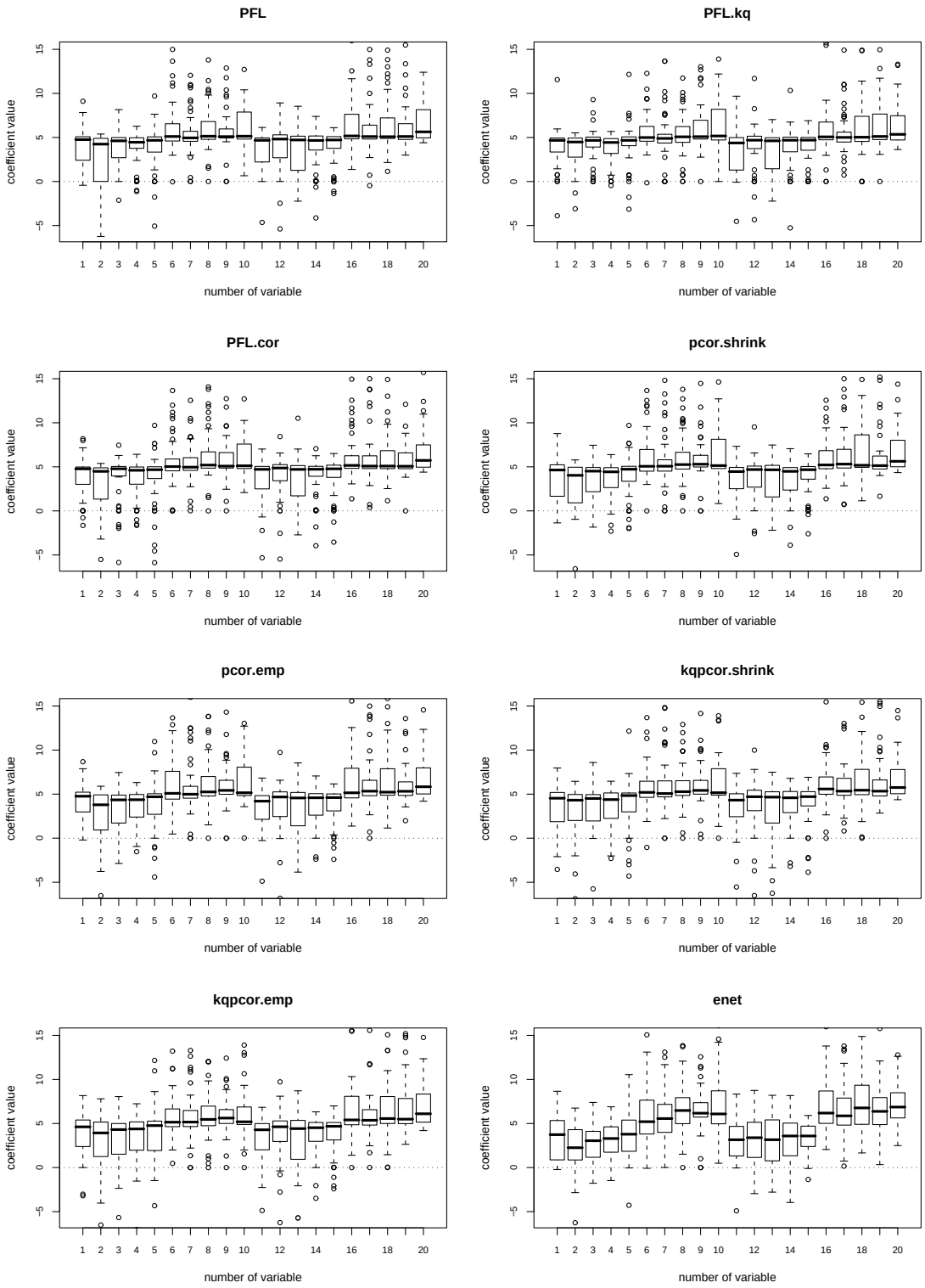

Figure A.21.: Boxplots of the predictors for the seventh simulation setting and correlation $\rho = 0.9$

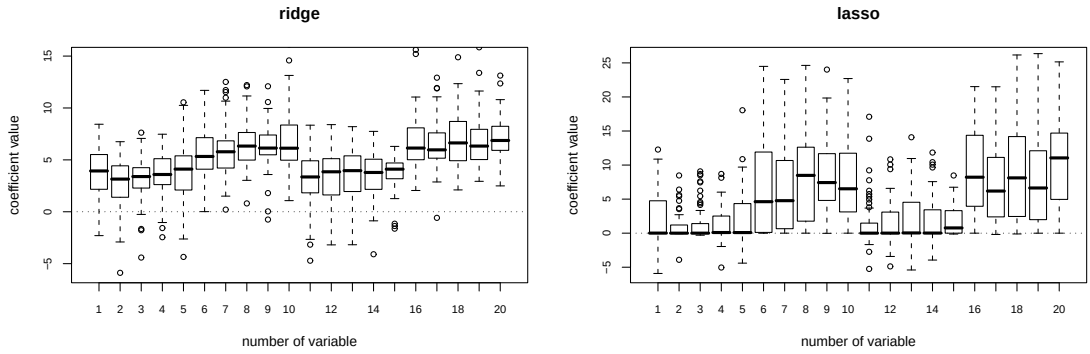

Figure A.22.: Boxplots of the predictors for the seventh simulation setting and correlation $\rho = 0.9$

Predictor	Method									
	pfl		pfl.kq		pfl.cor		pcor.shrink		pcor.emp	
β_1	4.76	(0.12)	4.67	(0.11)	4.77	(0.19)	4.64	(0.28)	4.76	(0.26)
β_2	4.24	(0.45)	4.48	(0.22)	4.49	(0.45)	4.04	(0.44)	3.79	(0.51)
β_3	4.62	(0.17)	4.67	(0.12)	4.78	(0.10)	4.51	(0.24)	4.35	(0.38)
β_4	4.46	(0.16)	4.44	(0.15)	4.61	(0.15)	4.43	(0.18)	4.37	(0.24)
β_5	4.67	(0.08)	4.67	(0.08)	4.68	(0.10)	4.73	(0.15)	4.70	(0.13)
β_6	5.11	(0.08)	4.99	(0.15)	5.03	(0.07)	5.06	(0.15)	5.09	(0.18)
β_7	4.95	(0.15)	4.89	(0.10)	4.96	(0.12)	5.07	(0.15)	4.99	(0.16)
β_8	5.14	(0.07)	5.07	(0.13)	5.20	(0.17)	5.26	(0.15)	5.25	(0.29)
β_9	5.10	(0.09)	5.09	(0.12)	5.08	(0.17)	5.30	(0.20)	5.43	(0.27)
β_{10}	5.14	(0.26)	5.17	(0.26)	5.12	(0.21)	5.14	(0.24)	5.16	(0.26)
β_{11}	4.67	(0.25)	4.38	(0.23)	4.72	(0.12)	4.46	(0.29)	4.20	(0.34)
β_{12}	4.82	(0.17)	4.69	(0.19)	4.87	(0.11)	4.69	(0.25)	4.68	(0.27)
β_{13}	4.72	(0.26)	4.62	(0.26)	4.71	(0.17)	4.66	(0.41)	4.57	(0.51)
β_{14}	4.66	(0.14)	4.69	(0.12)	4.73	(0.11)	4.49	(0.30)	4.59	(0.24)
β_{15}	4.72	(0.12)	4.70	(0.16)	4.77	(0.11)	4.68	(0.24)	4.61	(0.29)
β_{16}	5.17	(0.12)	5.07	(0.18)	5.16	(0.15)	5.21	(0.13)	5.15	(0.18)
β_{17}	5.10	(0.16)	5.01	(0.09)	5.08	(0.13)	5.31	(0.18)	5.34	(0.18)
β_{18}	5.09	(0.18)	5.03	(0.15)	5.09	(0.17)	5.18	(0.31)	5.23	(0.25)
β_{19}	5.11	(0.11)	5.11	(0.25)	5.07	(0.12)	5.14	(0.16)	5.32	(0.14)
β_{20}	5.64	(0.39)	5.35	(0.21)	5.73	(0.35)	5.63	(0.35)	5.84	(0.34)

Predictor	Method									
	kqpcor.shr		kqpcor.emp		enet		ridge		lasso	
β_1	4.54	(0.40)	4.62	(0.44)	3.74	(0.57)	3.93	(0.55)	0.01	(0.47)
β_2	4.32	(0.36)	3.93	(0.49)	2.25	(0.50)	3.15	(0.50)	0.00	(0.04)
β_3	4.51	(0.20)	4.31	(0.32)	3.05	(0.31)	3.39	(0.24)	0.00	(0.06)
β_4	4.38	(0.32)	4.38	(0.36)	3.30	(0.33)	3.59	(0.37)	0.10	(0.25)
β_5	4.84	(0.14)	4.76	(0.23)	3.79	(0.39)	4.11	(0.44)	0.09	(0.62)
β_6	5.20	(0.18)	5.15	(0.19)	5.21	(0.61)	5.33	(0.43)	4.62	(1.95)
β_7	5.07	(0.20)	5.16	(0.23)	5.56	(0.45)	5.78	(0.37)	4.77	(1.58)
β_8	5.27	(0.16)	5.47	(0.29)	6.48	(0.47)	6.33	(0.27)	8.48	(1.27)
β_9	5.42	(0.24)	5.62	(0.29)	6.17	(0.19)	6.14	(0.25)	7.43	(0.98)
β_{10}	5.16	(0.28)	5.19	(0.16)	6.08	(0.29)	6.13	(0.29)	6.52	(1.40)
β_{11}	4.32	(0.31)	4.29	(0.35)	3.16	(0.50)	3.35	(0.47)	0.00	(0.01)
β_{12}	4.70	(0.19)	4.63	(0.27)	3.39	(0.47)	3.85	(0.38)	0.00	(0.25)
β_{13}	4.66	(0.30)	4.42	(0.52)	3.16	(0.80)	3.96	(0.65)	0.03	(0.37)
β_{14}	4.59	(0.22)	4.52	(0.30)	3.59	(0.45)	3.79	(0.34)	0.02	(0.12)
β_{15}	4.71	(0.16)	4.68	(0.30)	3.59	(0.29)	4.11	(0.23)	0.77	(0.48)
β_{16}	5.59	(0.22)	5.41	(0.35)	6.18	(0.40)	6.14	(0.39)	8.22	(1.22)
β_{17}	5.34	(0.28)	5.37	(0.16)	5.87	(0.25)	5.97	(0.22)	6.19	(1.34)
β_{18}	5.45	(0.31)	5.56	(0.30)	6.77	(0.50)	6.63	(0.43)	8.12	(2.02)
β_{19}	5.34	(0.27)	5.49	(0.28)	6.38	(0.32)	6.33	(0.30)	6.63	(1.04)
β_{20}	5.75	(0.41)	6.10	(0.38)	6.87	(0.36)	6.87	(0.26)	11.05	(1.23)

Table A.10.: Medians of the predictors for the seventh simulation setting and correlation $\rho = 0.9$ based on 50 replications.

B. Simulations: Binomial Distribution

B.1. Setting 1

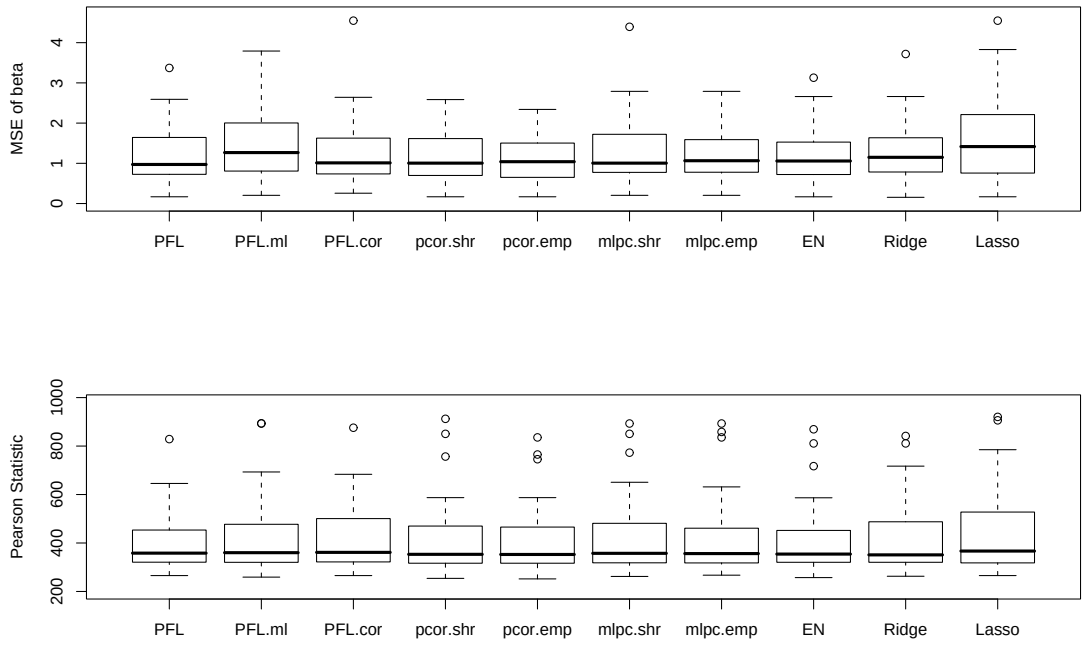

Figure B.1.: Boxplots of the pearson statistic on the test data set and MSE of β for the first simulation setting and correlation $\rho = 0.9$

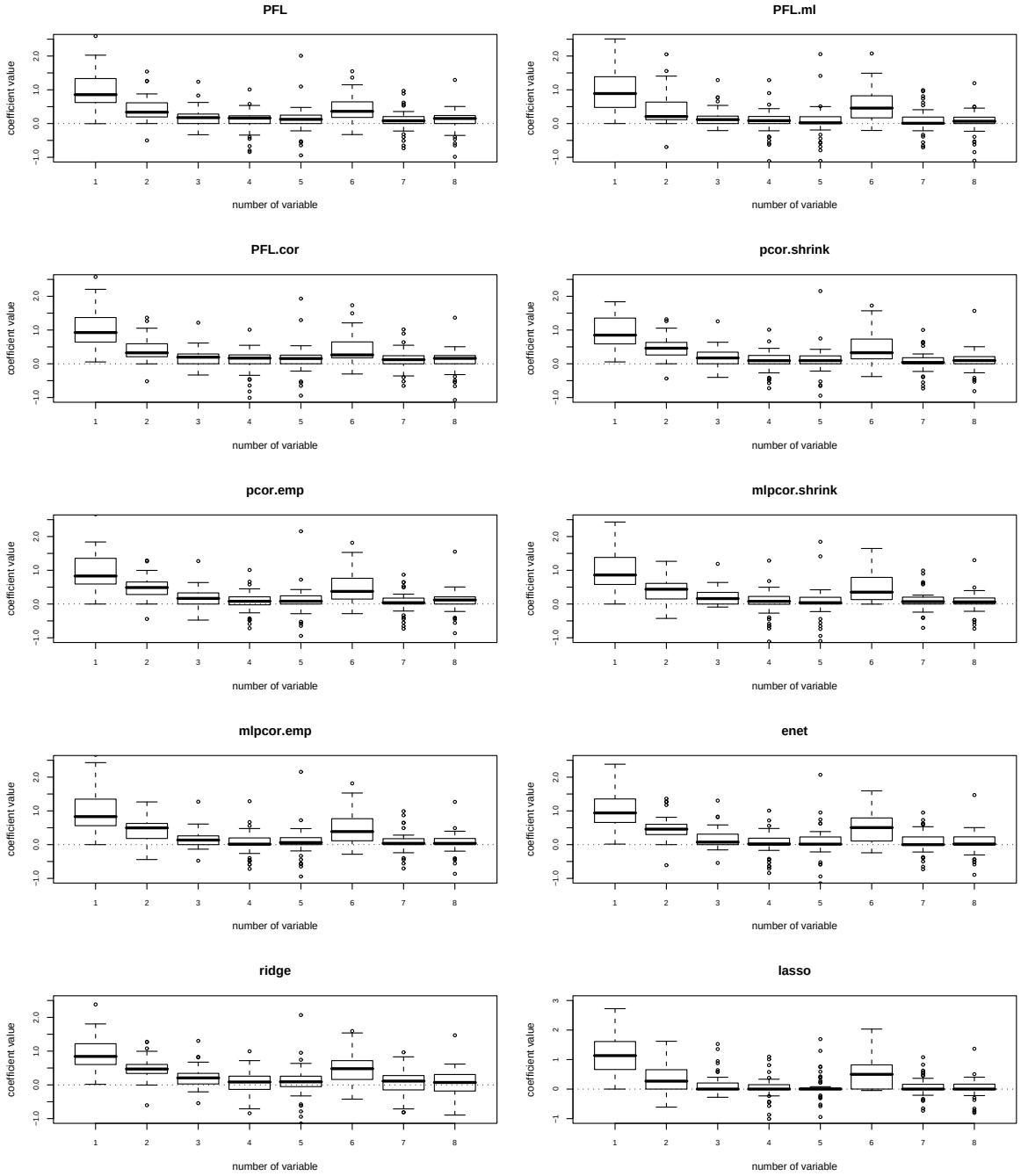

Figure B.2.: Boxplots of the predictors for the first simulation setting and correlation $\rho = 0.9$

B.2. Setting 2

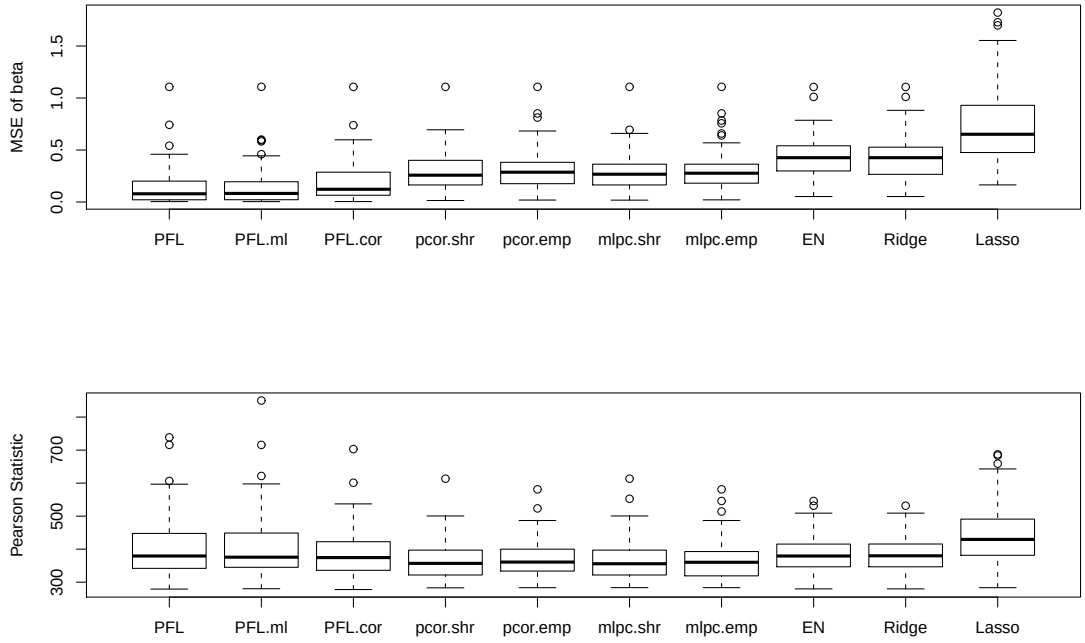

Figure B.3.: Boxplots of the pearson statistic on the test data set and MSE of $\boldsymbol{\beta}$ for the second simulation setting and correlation $\rho = 0.5$

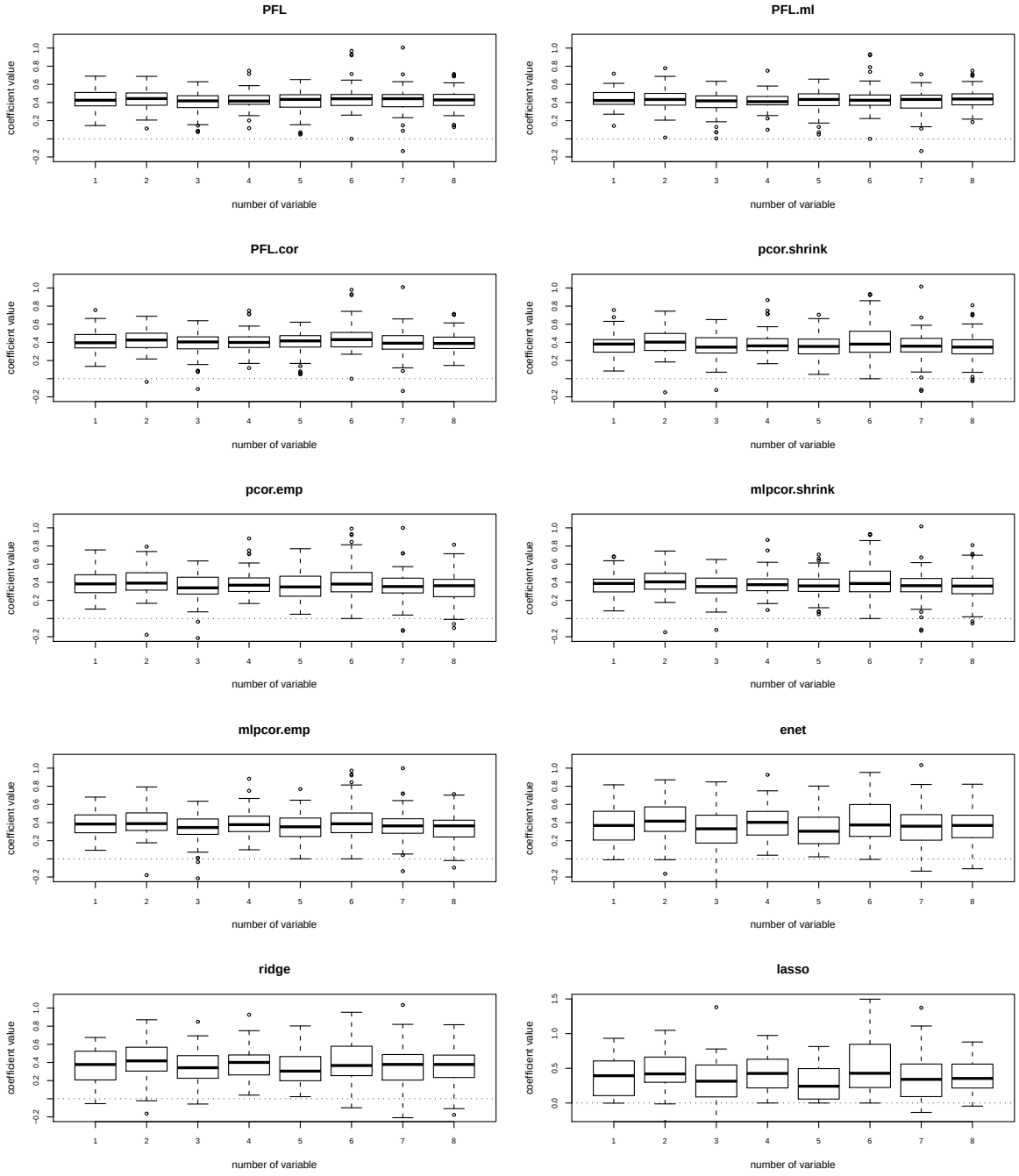

Figure B.4.: Boxplots of the predictors for the second simulation setting and correlation $\rho = 0.5$

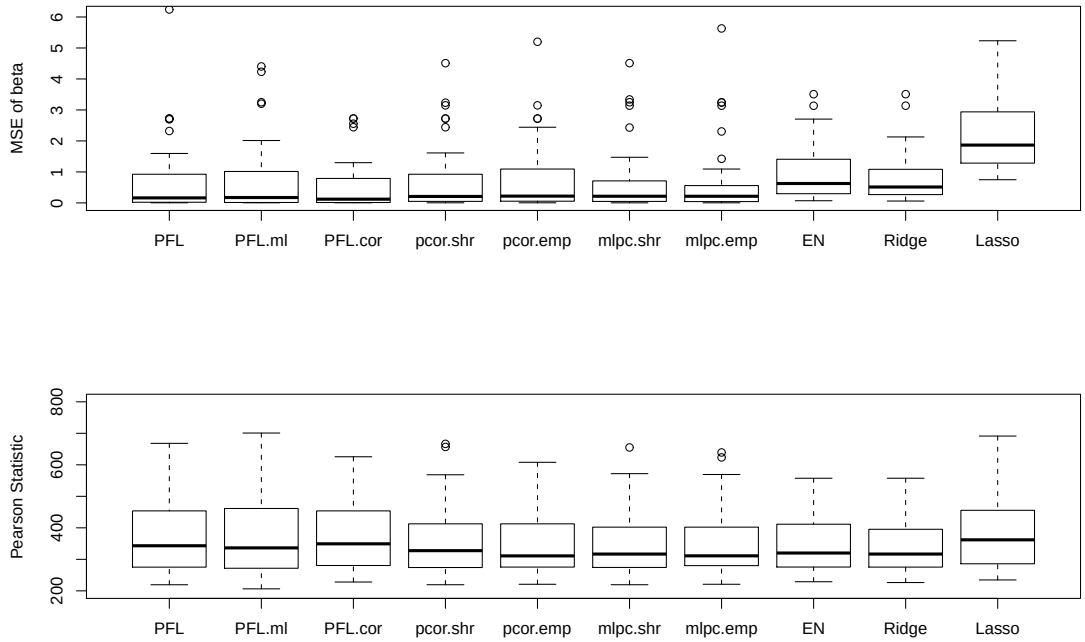

Figure B.5.: Boxplots of the pearson statistic on the test data set and MSE of $\boldsymbol{\beta}$ for the second simulation setting and correlation $\rho = 0.9$

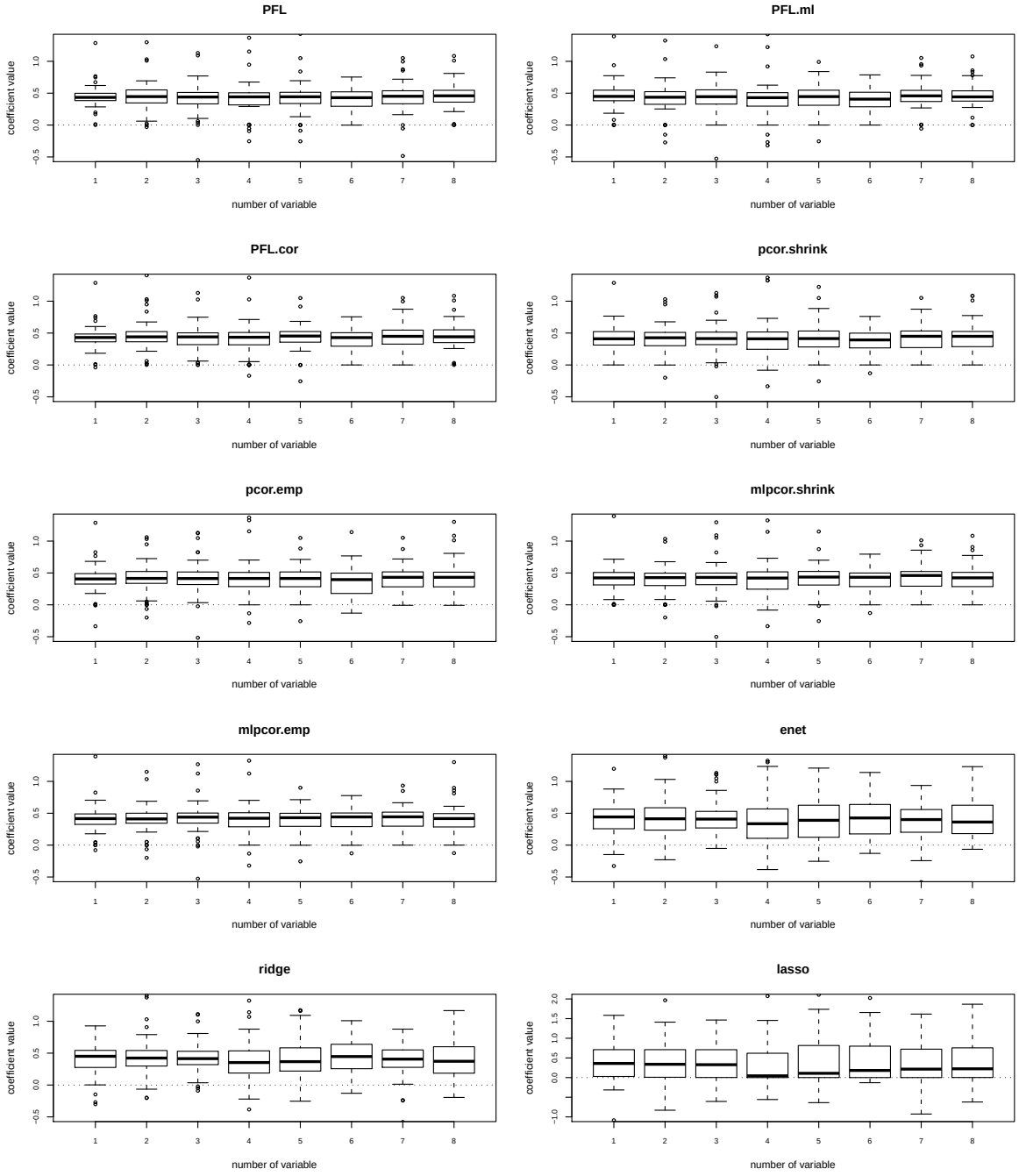

Figure B.6.: Boxplots of the predictors for the second simulation setting and correlation $\rho = 0.9$

B.3. Setting 3

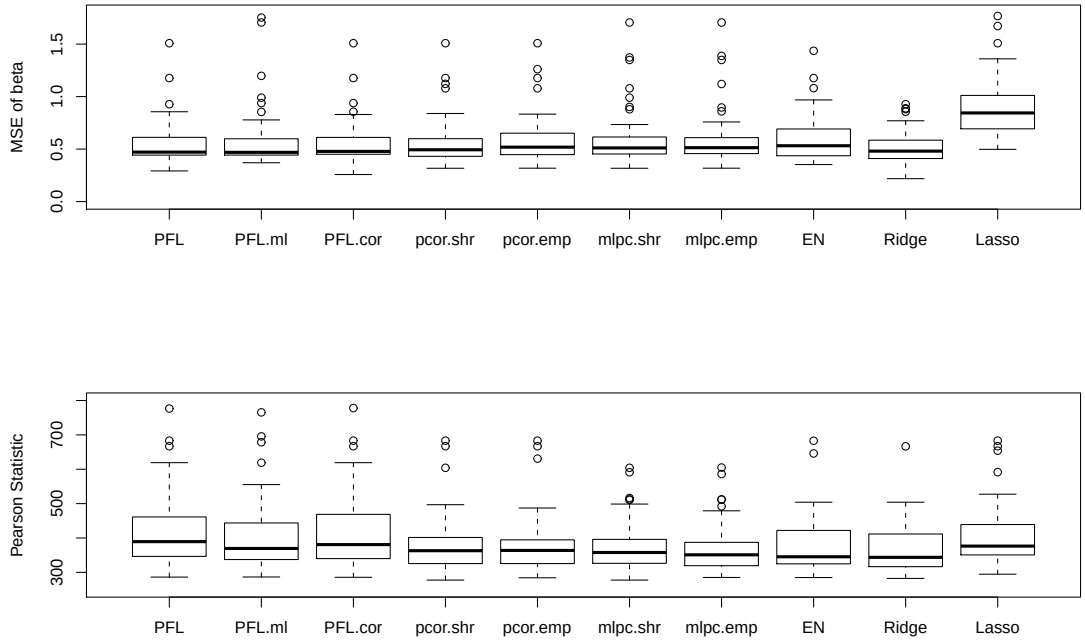

Figure B.7.: Boxplots of the pearson statistic on the test data set and MSE of $\boldsymbol{\beta}$ for the third simulation setting and correlation $\rho = 0.5$

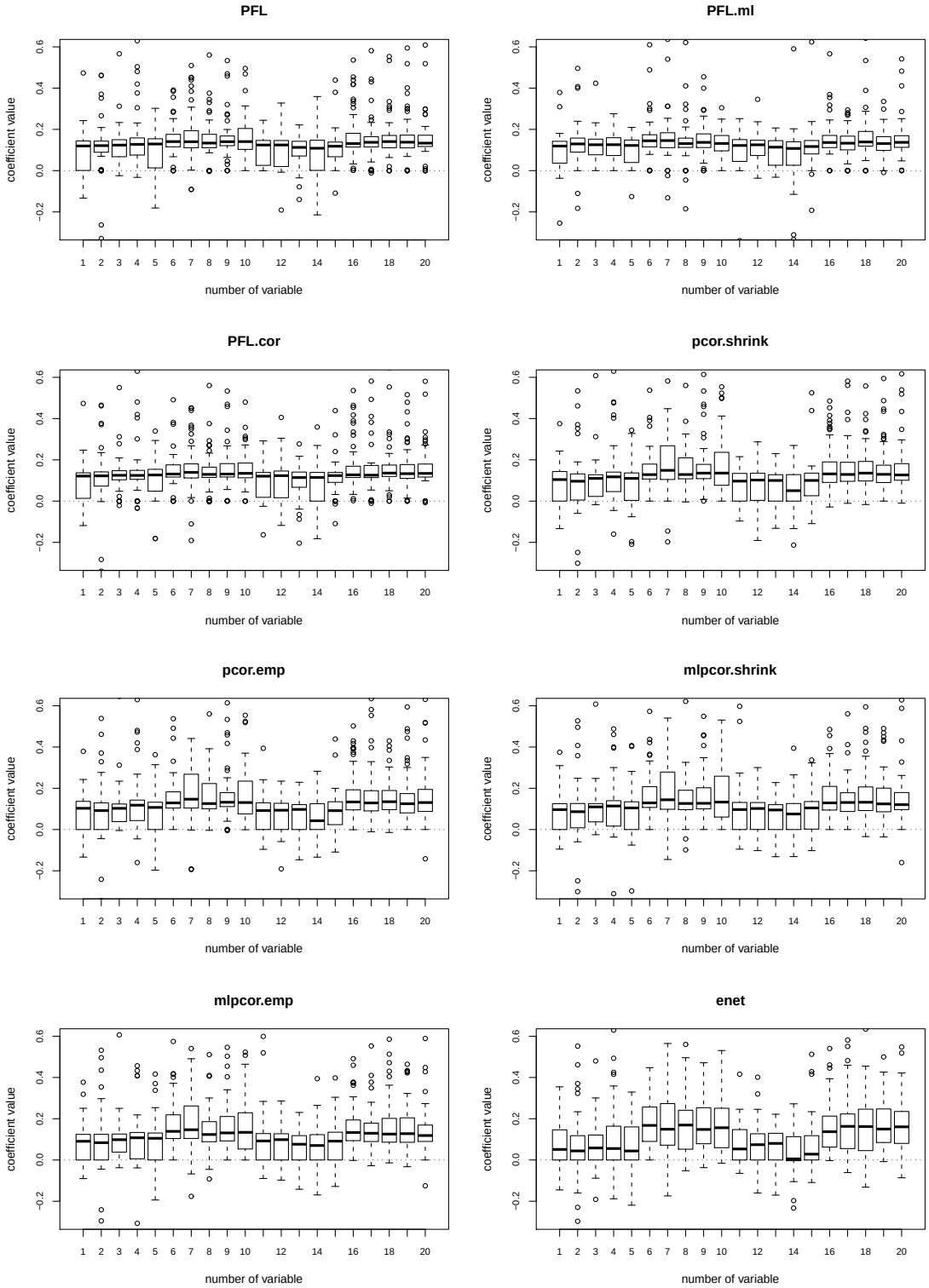

Figure B.8.: Boxplots of the predictors for the third simulation setting and correlation $\rho = 0.5$

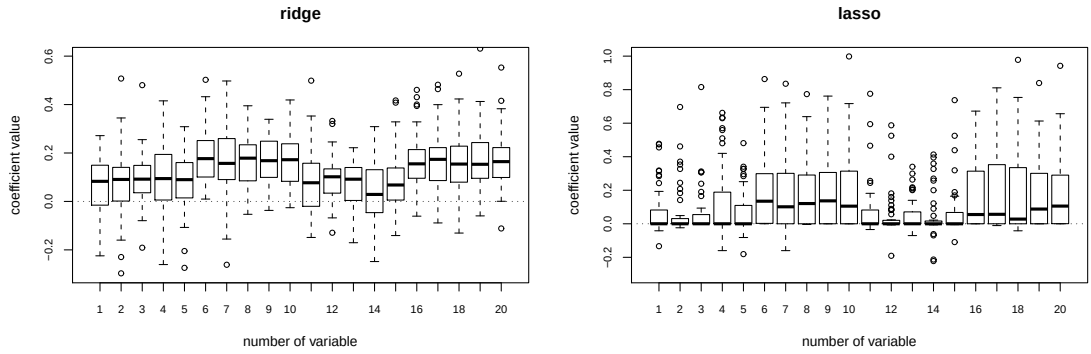

Figure B.9.: Boxplots of the predictors for the third simulation setting and correlation $\rho = 0.5$

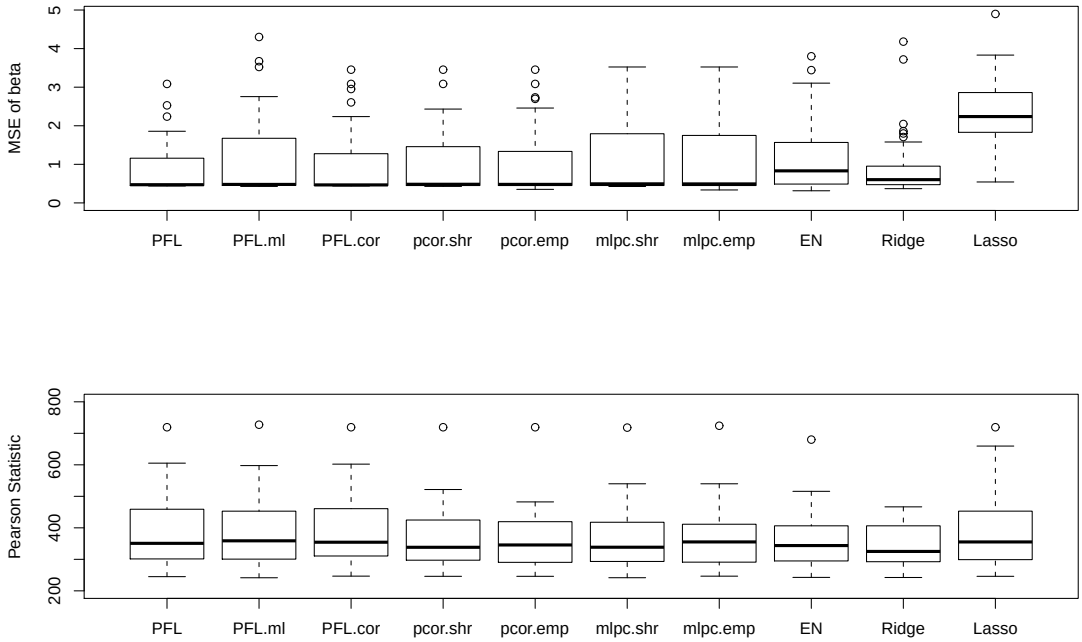

Figure B.10.: Boxplots of the pearson statistic on the test data set and MSE of β for the third simulation setting and correlation $\rho = 0.9$

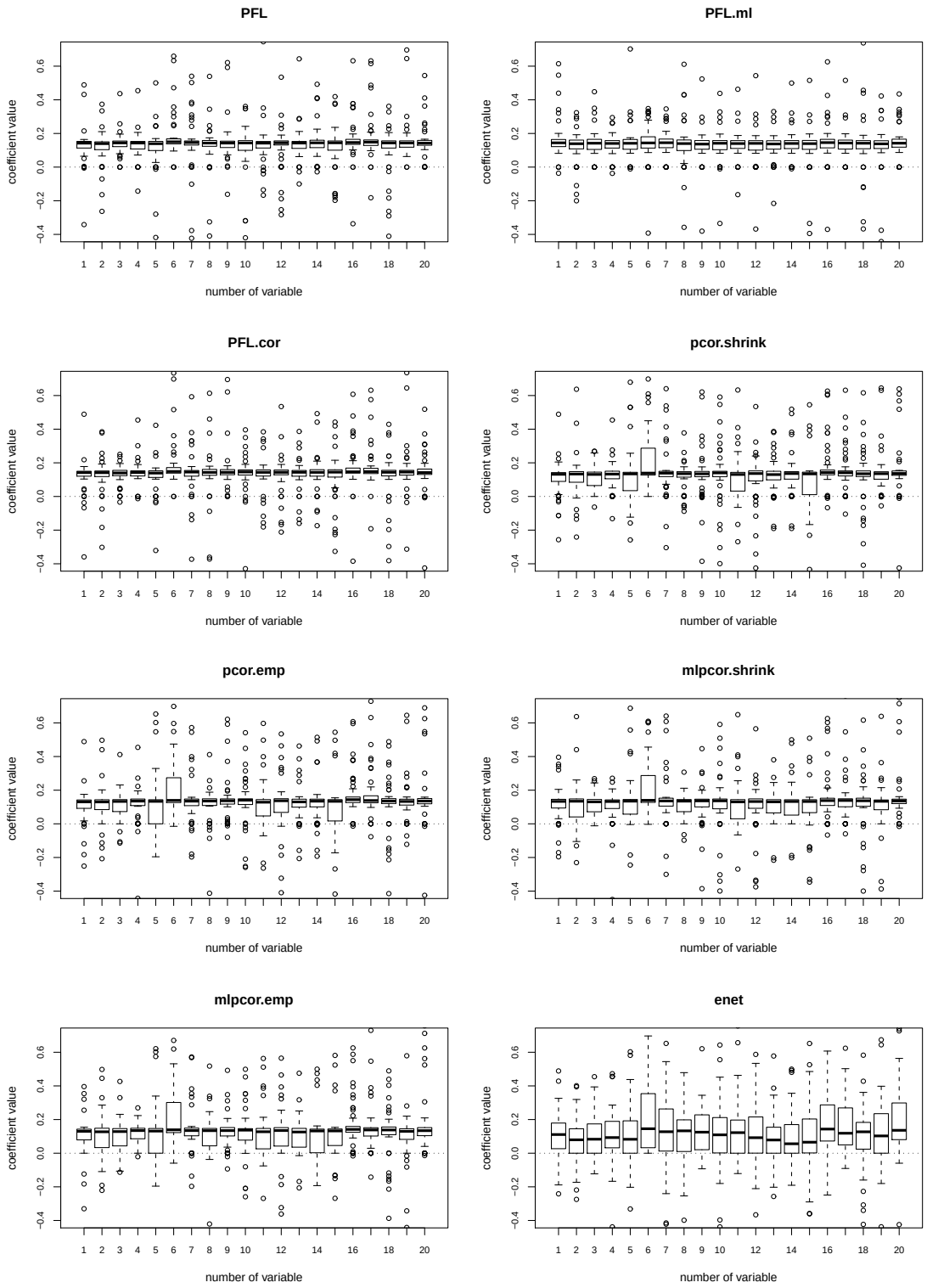

Figure B.11.: Boxplots of the predictors for the third simulation setting and correlation $\rho = 0.9$

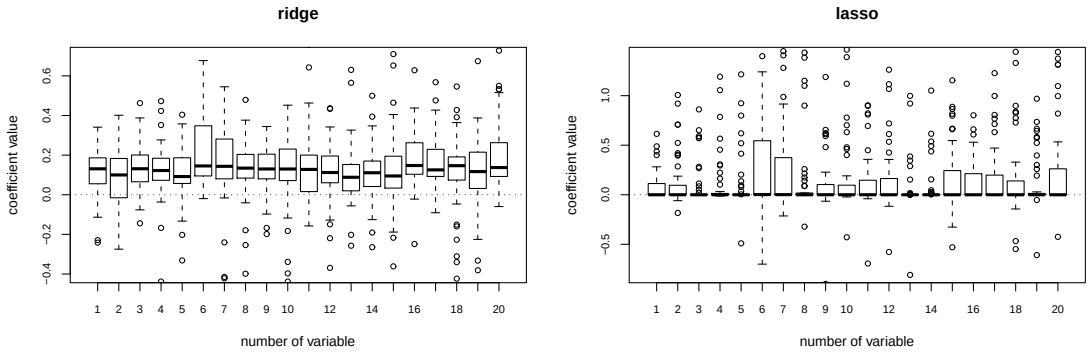

Figure B.12.: Boxplots of the predictors for the third simulation setting and correlation $\rho = 0.9$

B.4. Setting 4

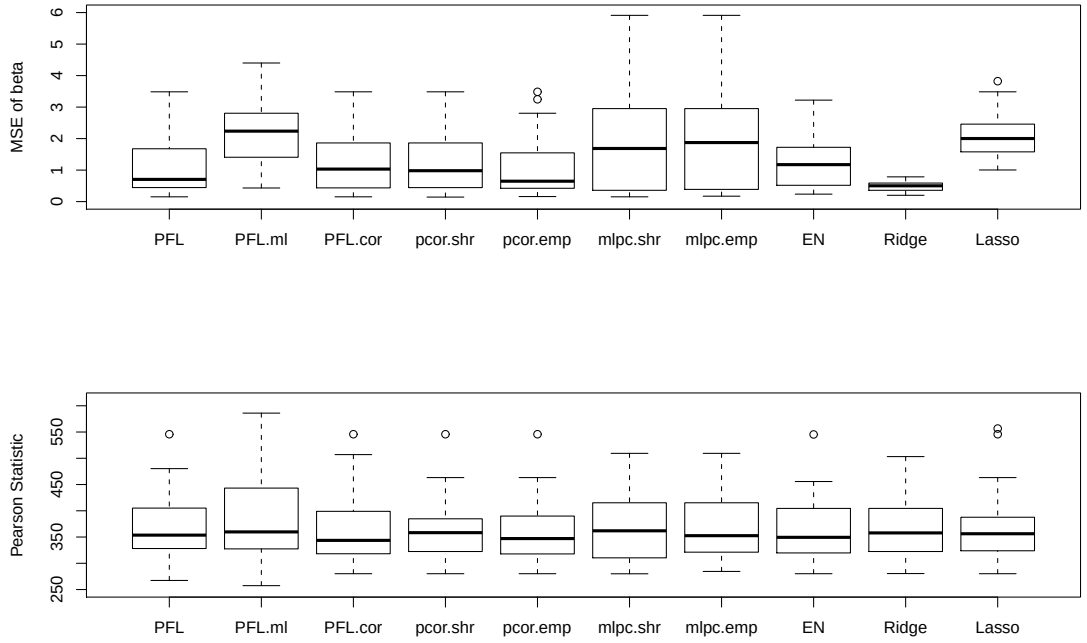

Figure B.13.: Boxplots of the pearson statistic on the test data set and MSE of β for the fourth simulation setting

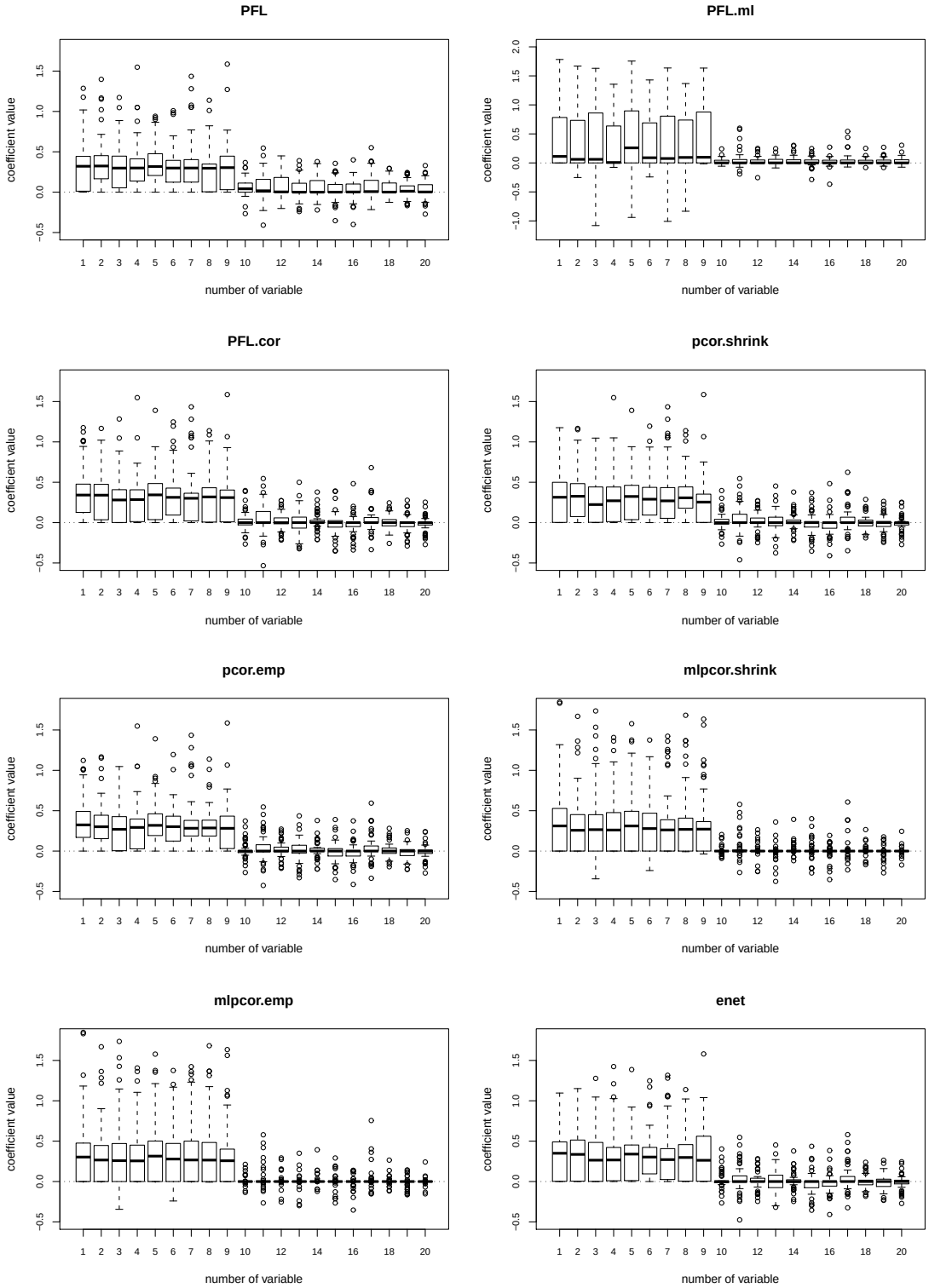

Figure B.14.: Boxplots of the predictors for the fourth simulation setting

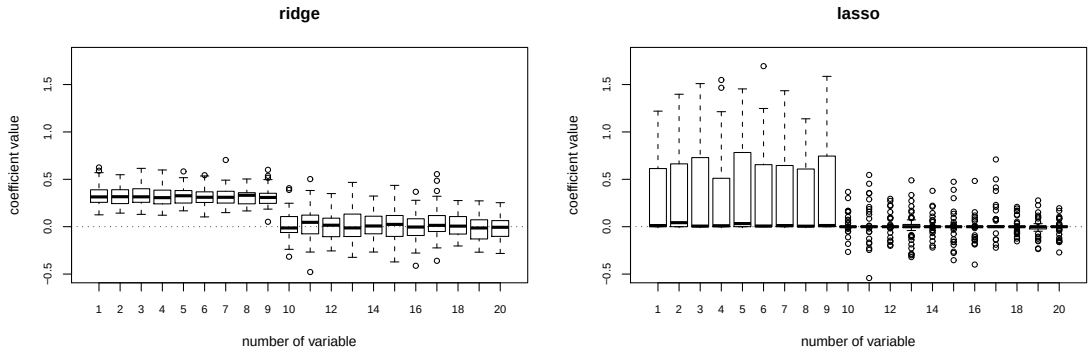

Figure B.15.: Boxplots of the predictors for the fourth simulation setting

C. Simulations: Poisson Distribution

C.1. Setting 1

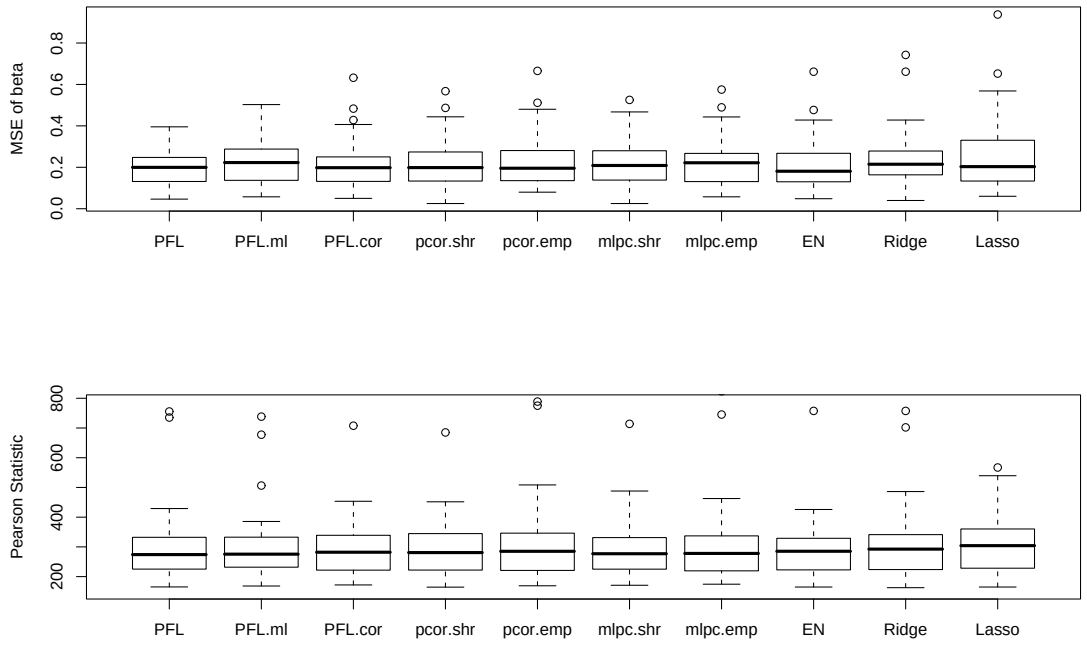

Figure C.1.: Boxplots of the pearson statistic on the test data set and MSE of β for the first simulation setting and correlation $\rho = 0.5$

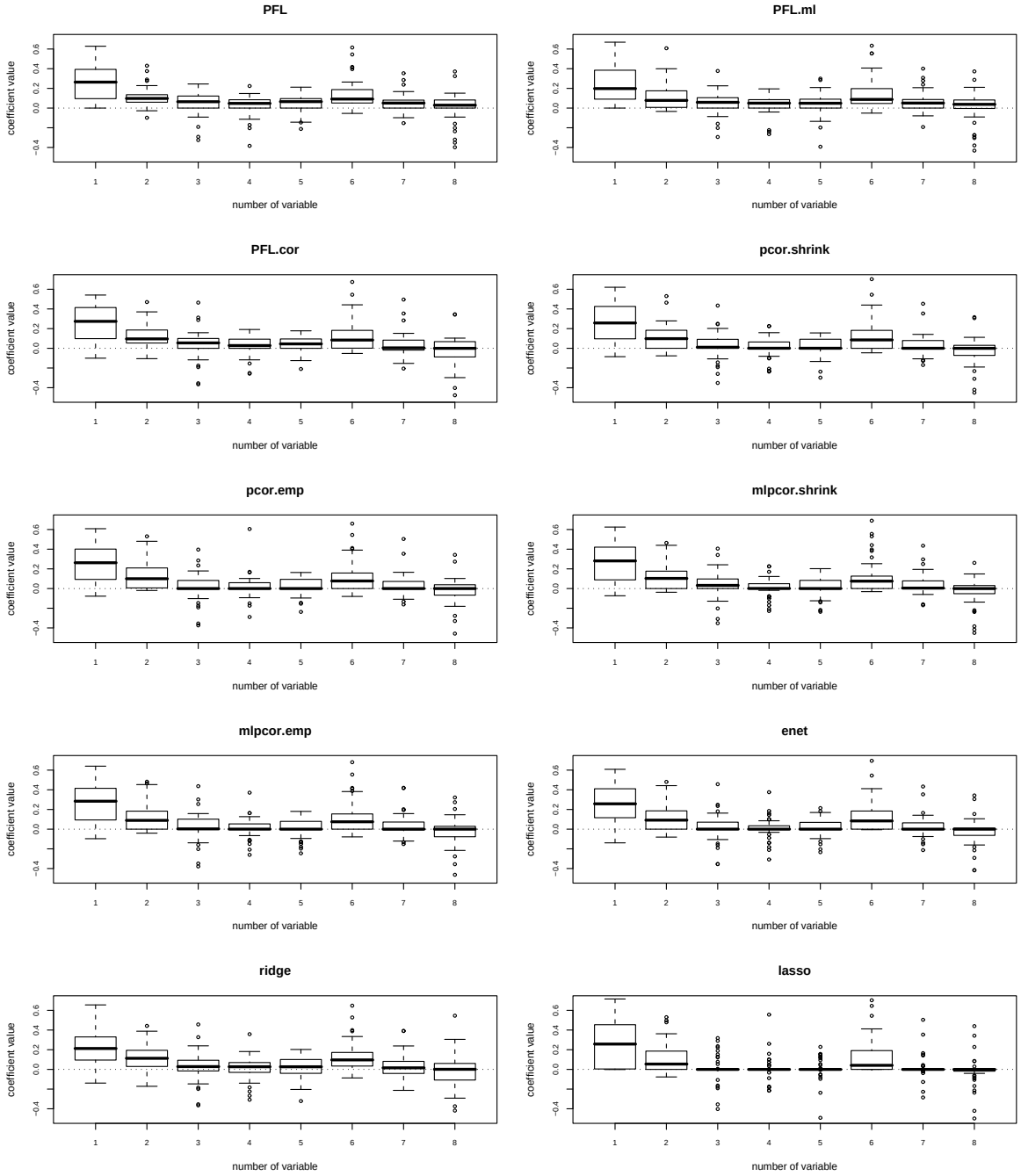

Figure C.2.: Boxplots of the predictors for the first simulation setting and correlation $\rho = 0.5$

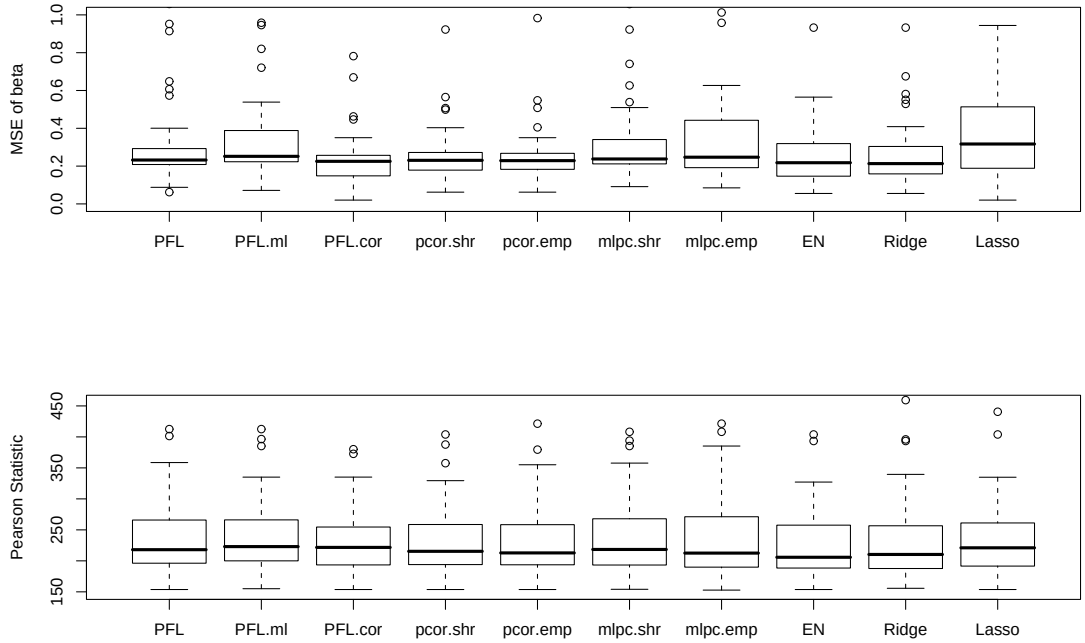

Figure C.3.: Boxplots of the pearson statistic on the test data set and MSE of $\boldsymbol{\beta}$ for the first simulation setting and correlation $\rho = 0.9$

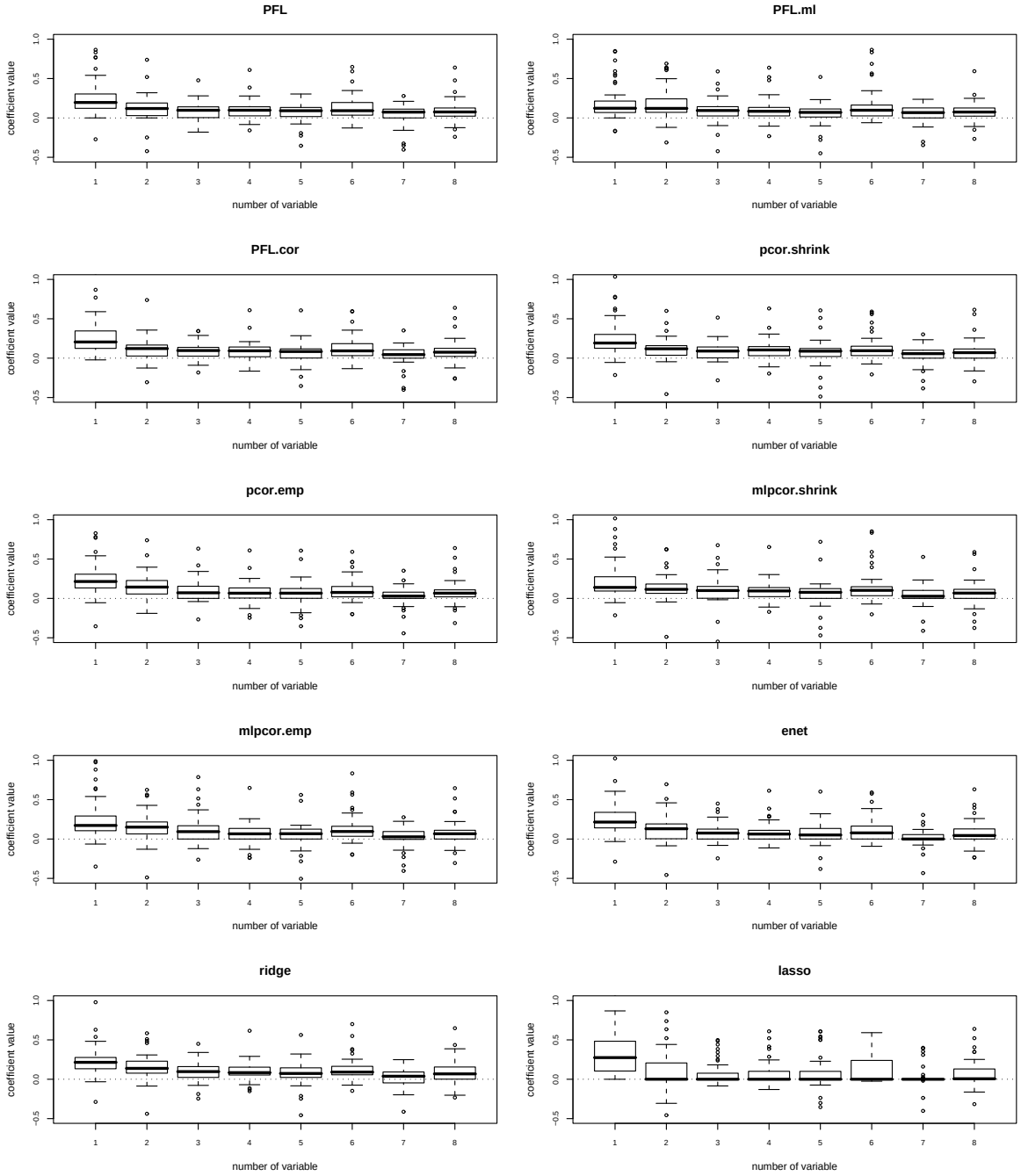

Figure C.4.: Boxplots of the predictors for the first simulation setting and correlation $\rho = 0.9$

C.2. Setting 2

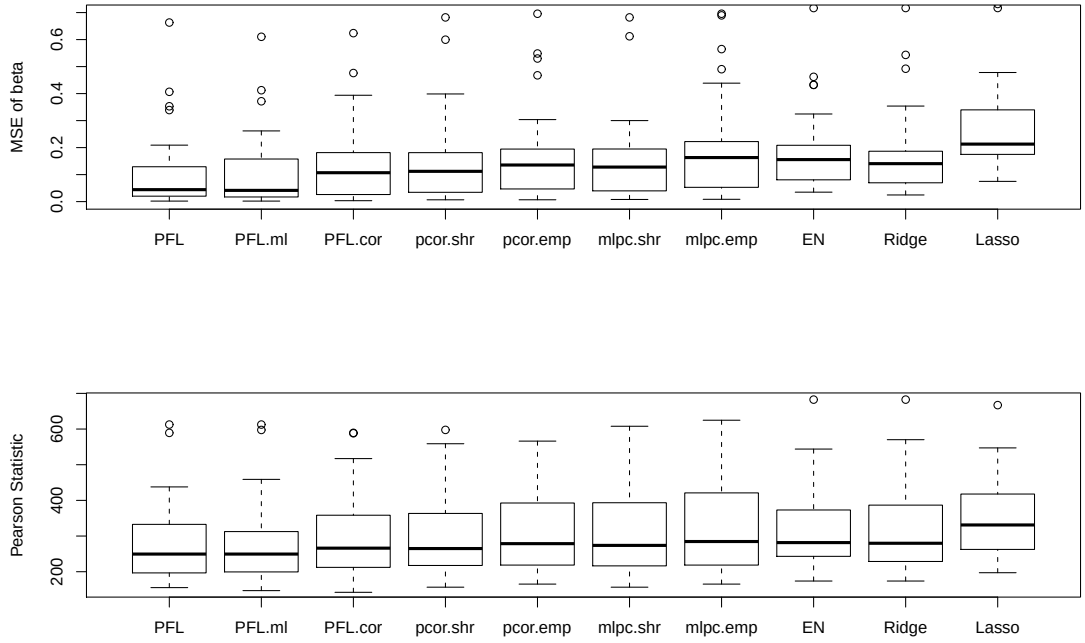

Figure C.5.: Boxplots of the pearson statistic on the test data set and MSE of β for the second simulation setting and correlation $\rho = 0.5$

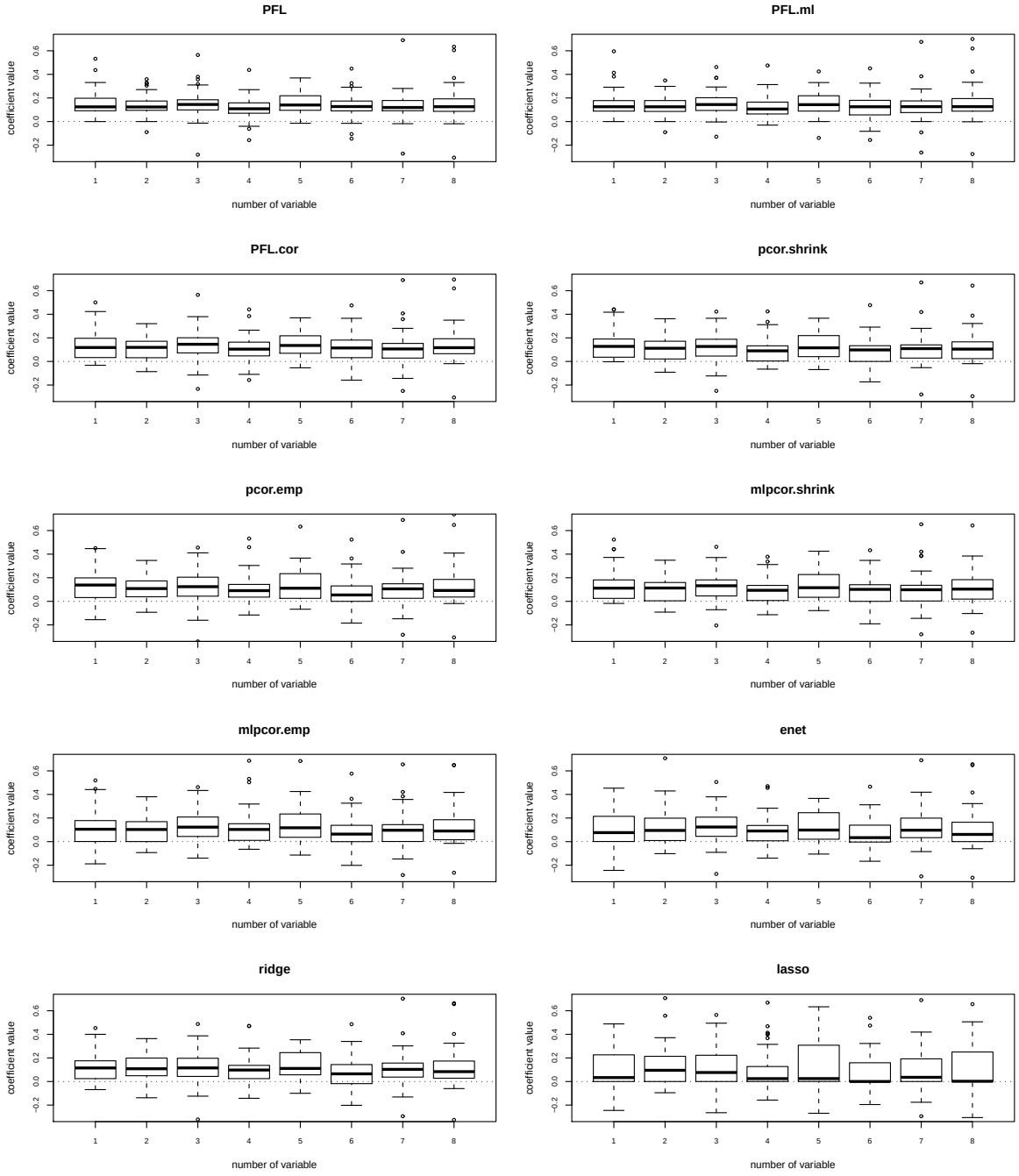

Figure C.6.: Boxplots of the predictors for the second simulation setting and correlation $\rho = 0.5$

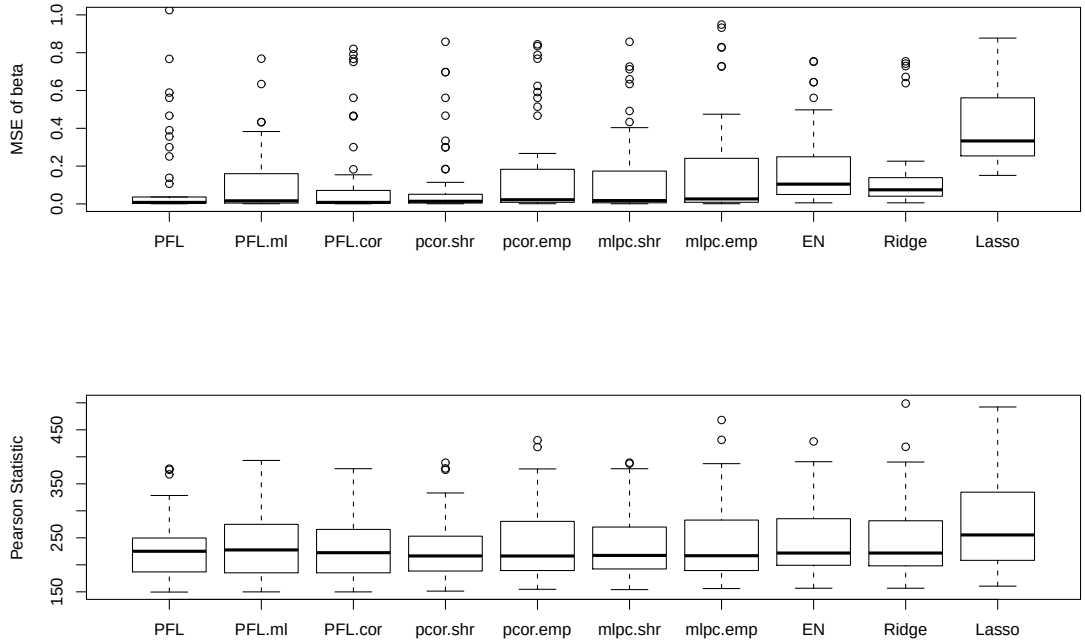

Figure C.7.: Boxplots of the pearson statistic on the test data set and MSE of $\boldsymbol{\beta}$ for the second simulation setting and correlation $\rho = 0.9$

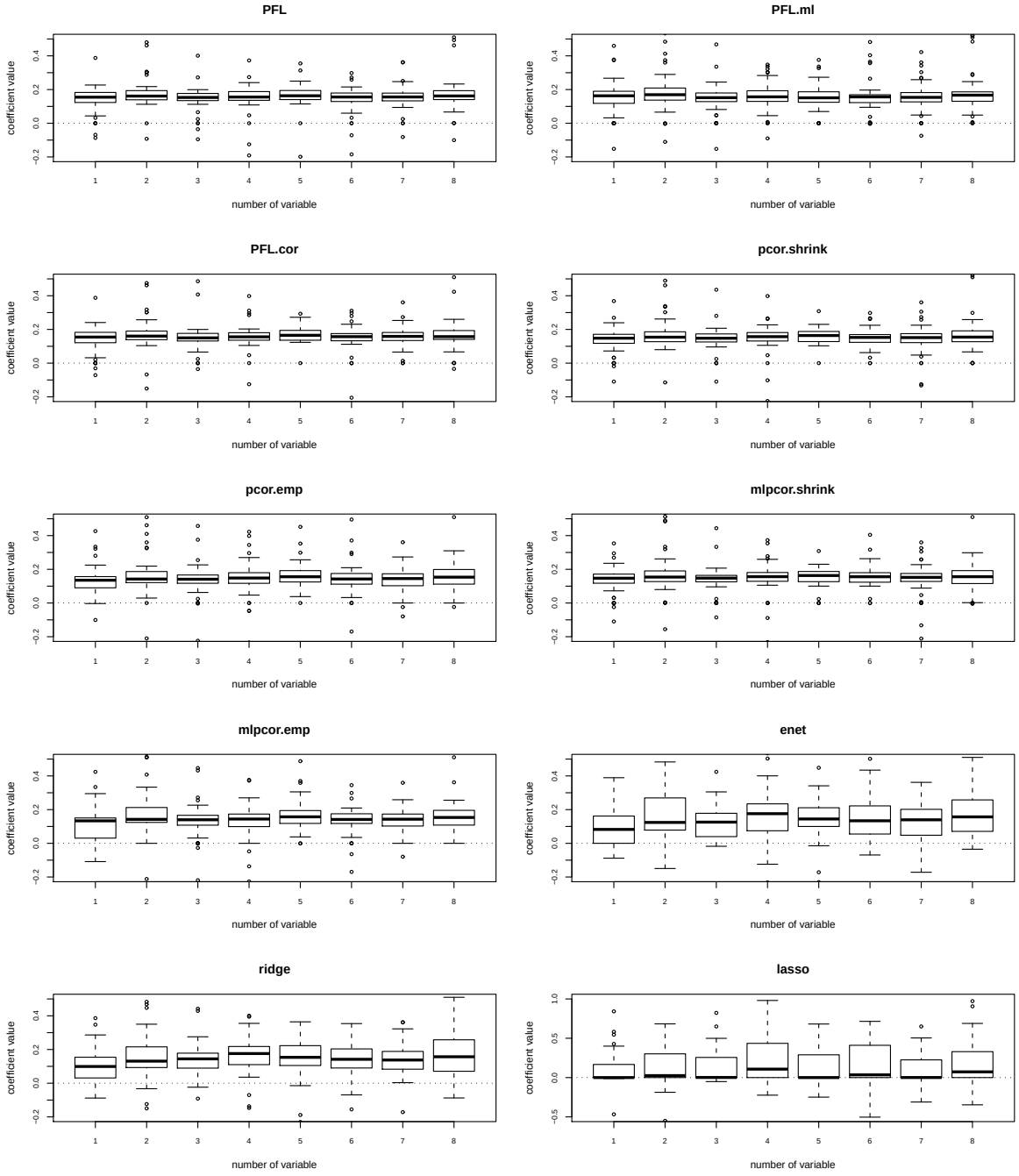

Figure C.8.: Boxplots of the predictors for the second simulation setting and correlation $\rho = 0.9$

C.3. Setting 3

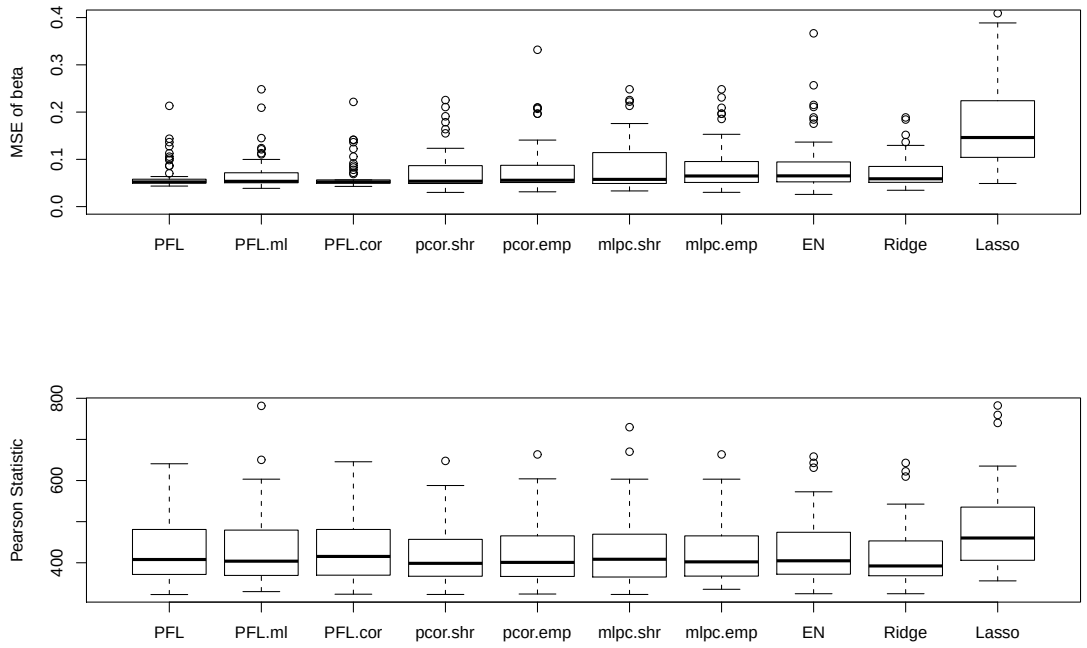

Figure C.9.: Boxplots of the pearson statistic on the test data set and MSE of $\boldsymbol{\beta}$ for the third simulation setting and correlation $\rho = 0.5$

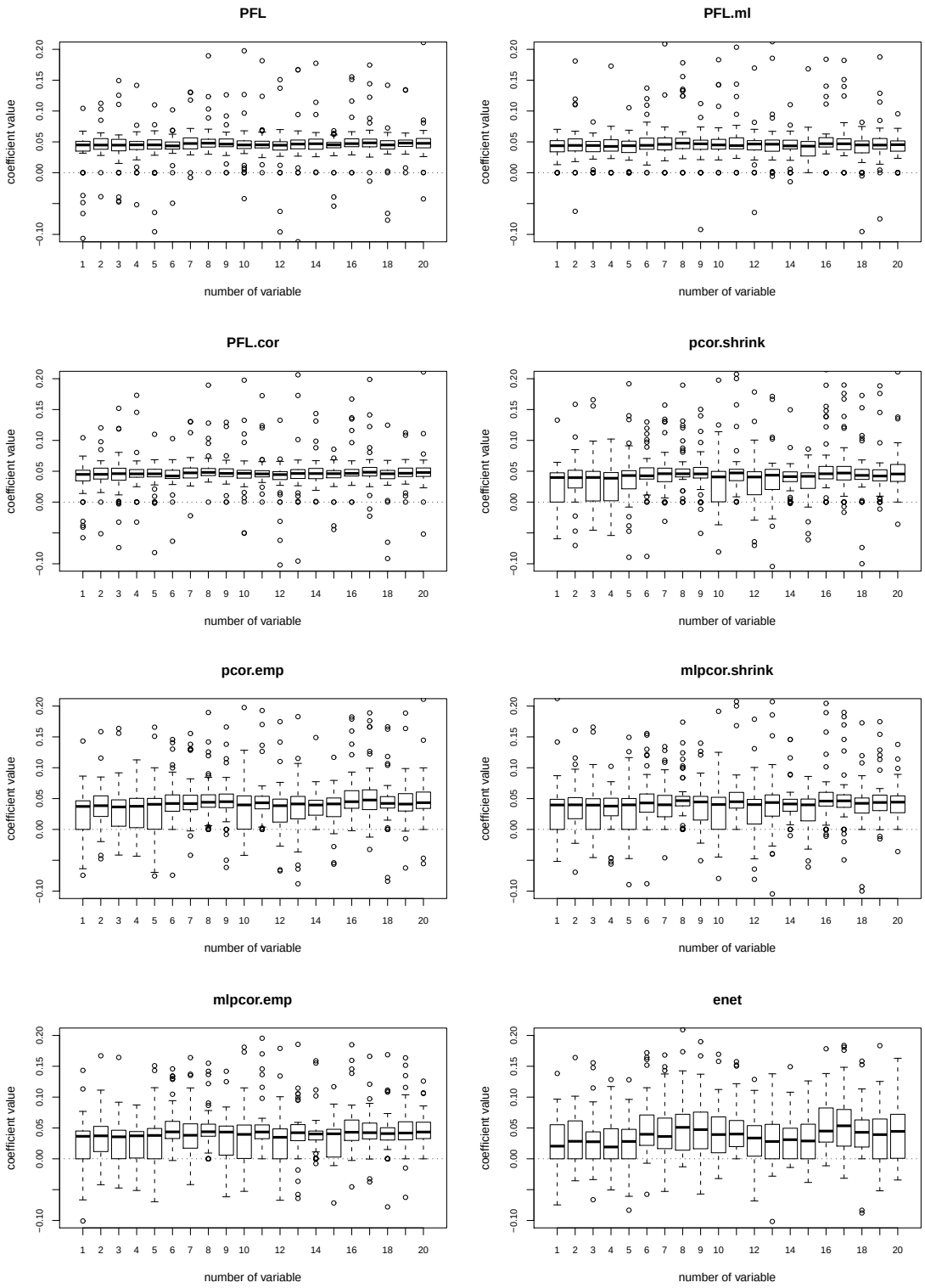

Figure C.10.: Boxplots of the predictors for the third simulation setting and correlation $\rho = 0.5$

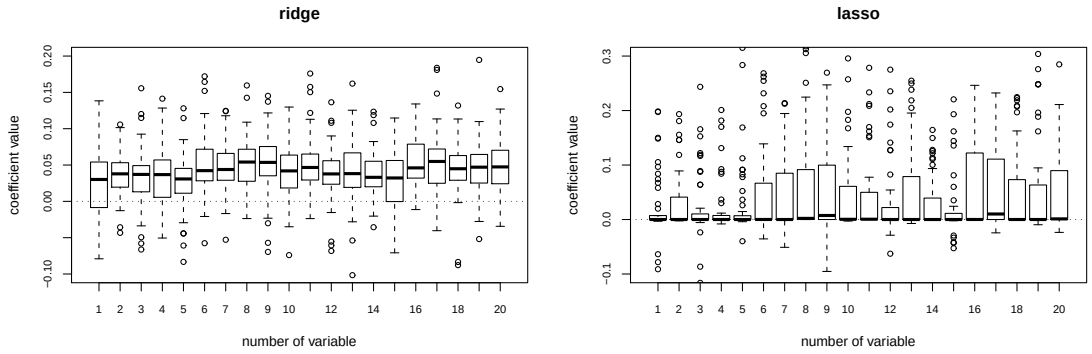

Figure C.11.: Boxplots of the predictors for the third simulation setting and correlation $\rho = 0.5$

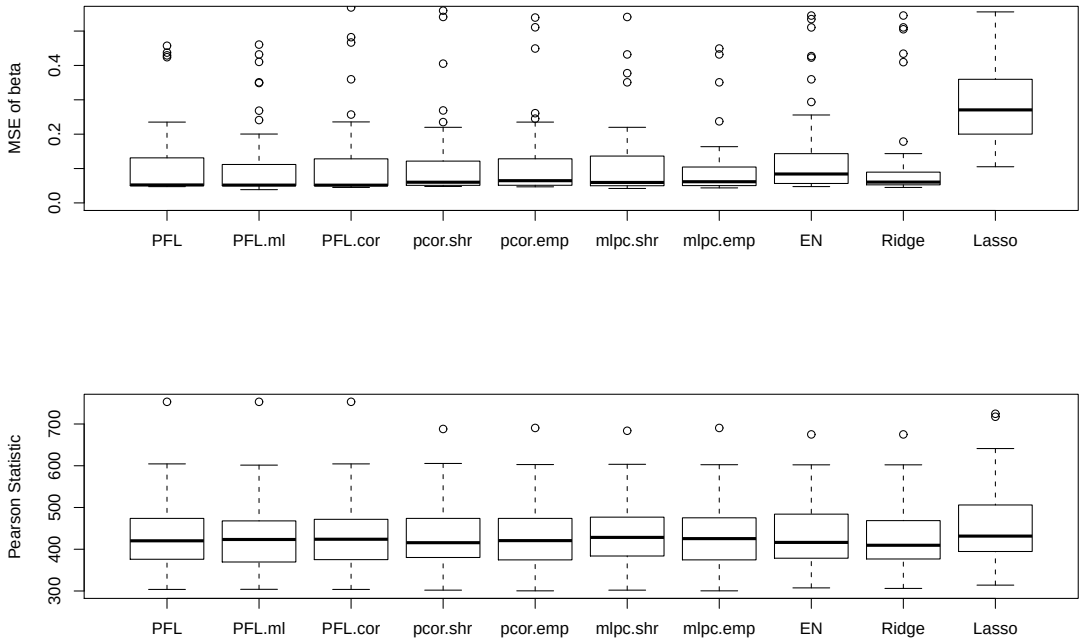

Figure C.12.: Boxplots of the pearson statistic on the test data set and MSE of β for the third simulation setting and correlation $\rho = 0.9$

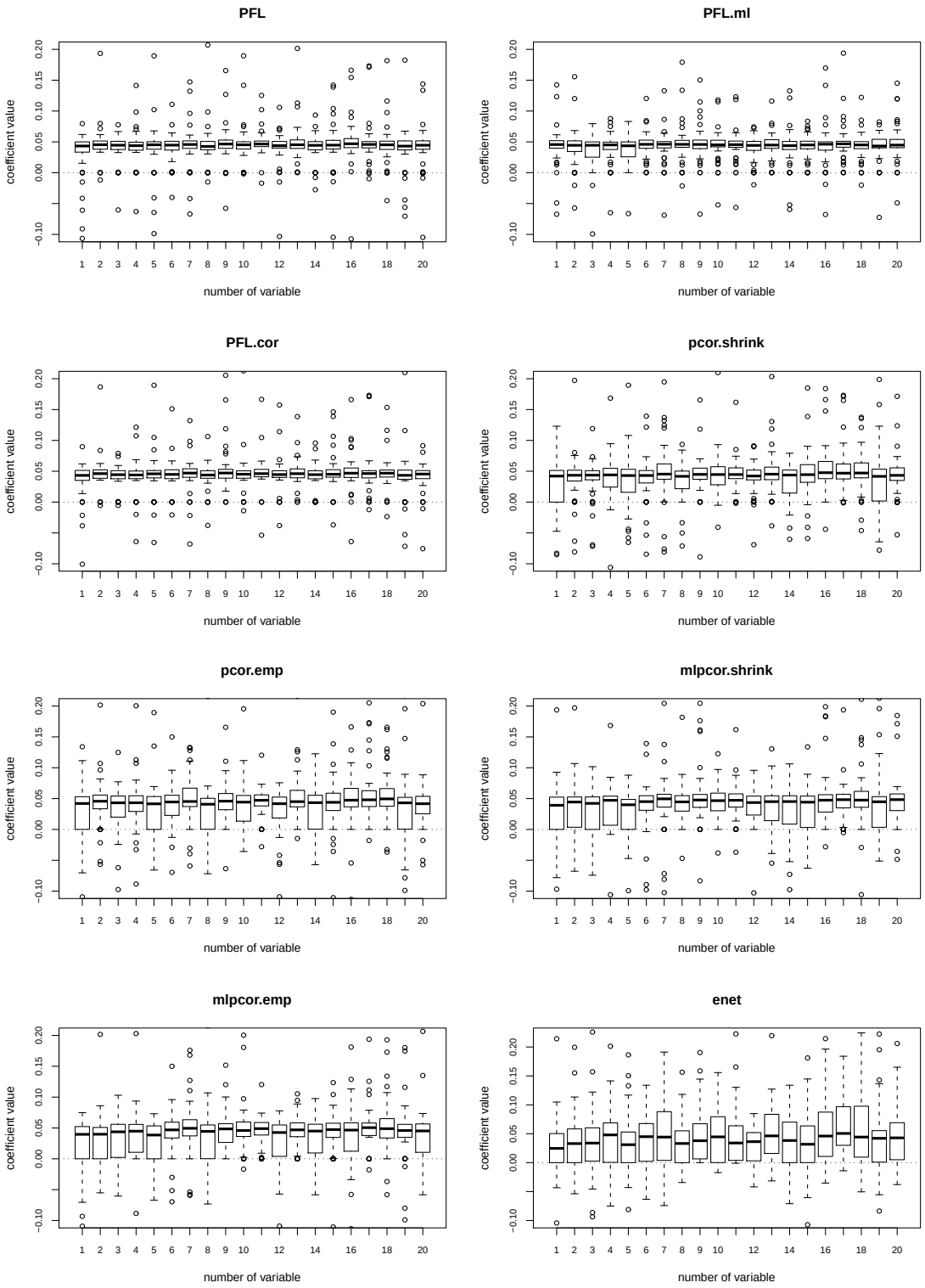

Figure C.13.: Boxplots of the predictors for the third simulation setting and correlation $\rho = 0.9$

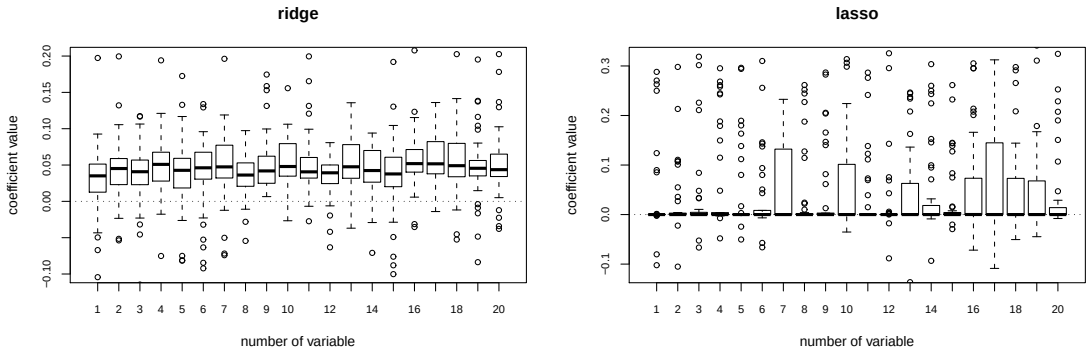

Figure C.14.: Boxplots of the predictors for the third simulation setting and correlation $\rho = 0.9$

C.4. Setting 4

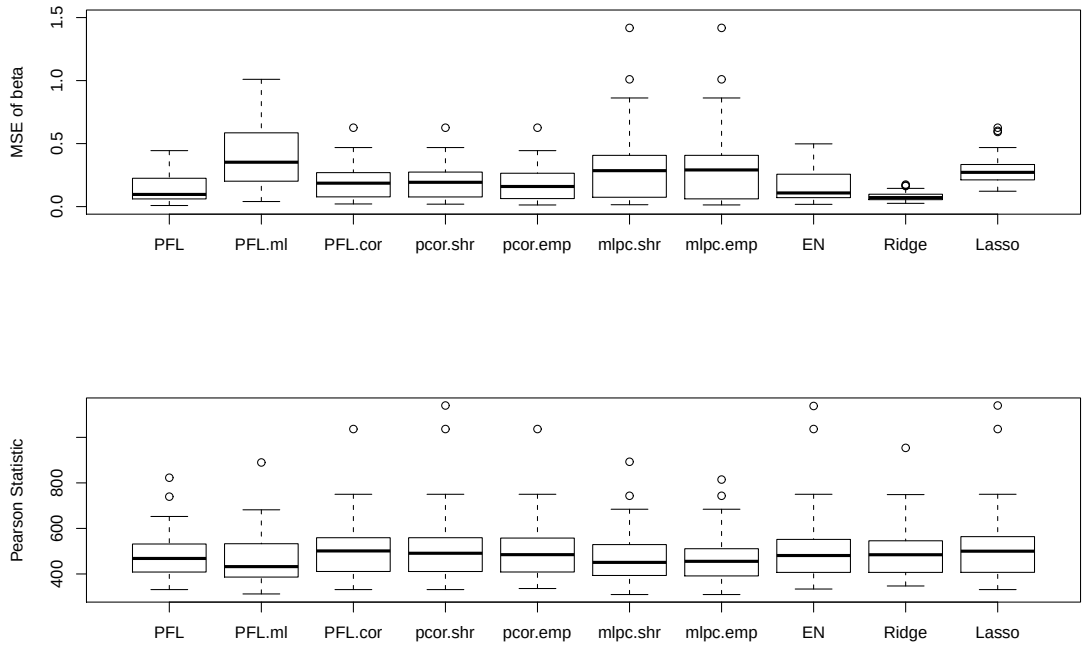

Figure C.15.: Boxplots of the pearson statistic on the test data set and MSE of β for the fourth simulation setting

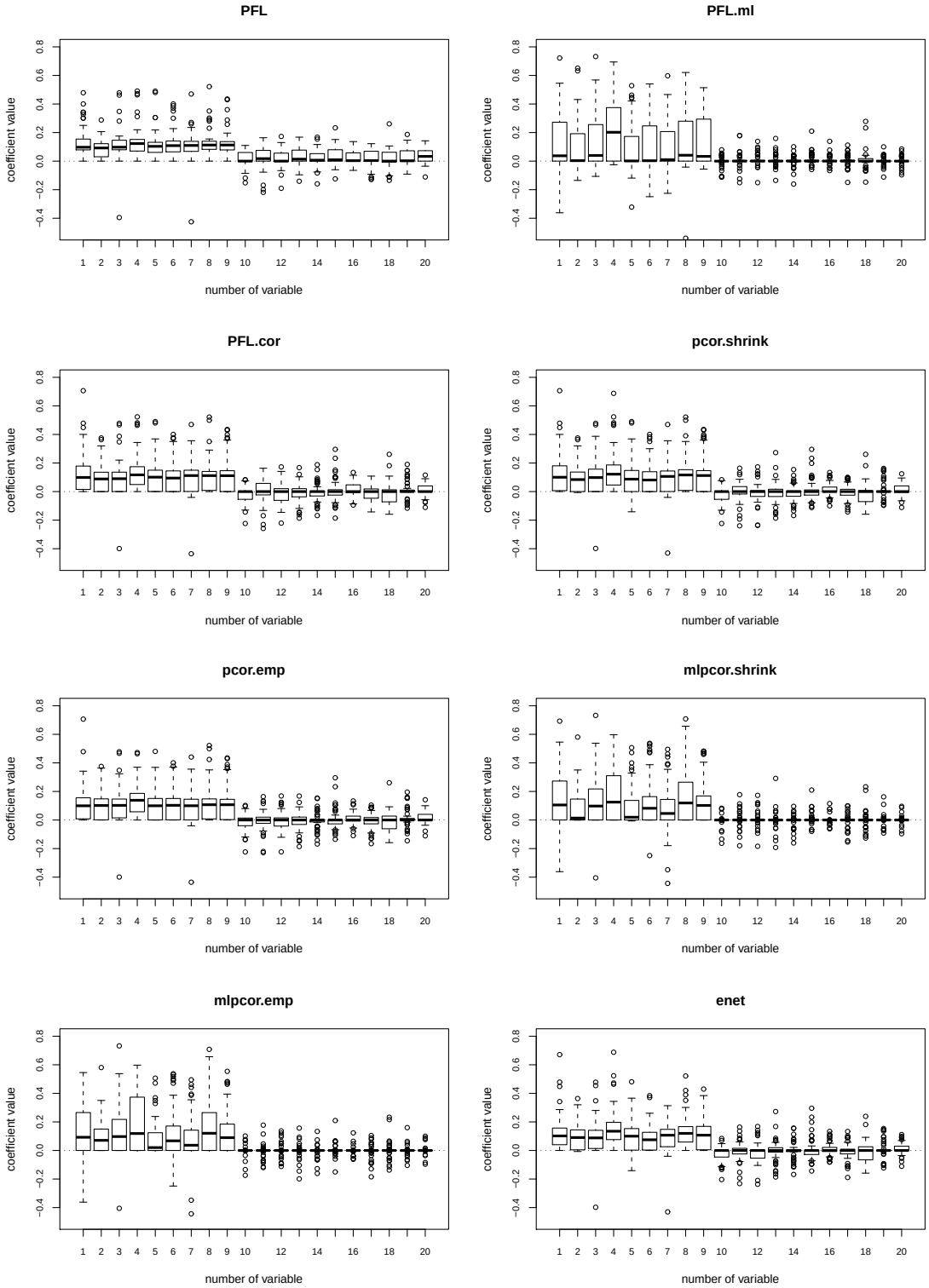

Figure C.16.: Boxplots of the predictors for the fourth simulation setting

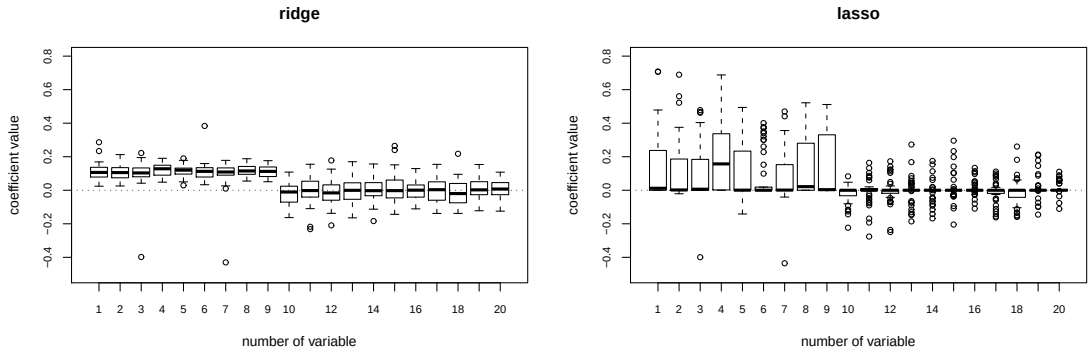

Figure C.17.: Boxplots of the predictors for the fourth simulation setting

C.5. Setting 5

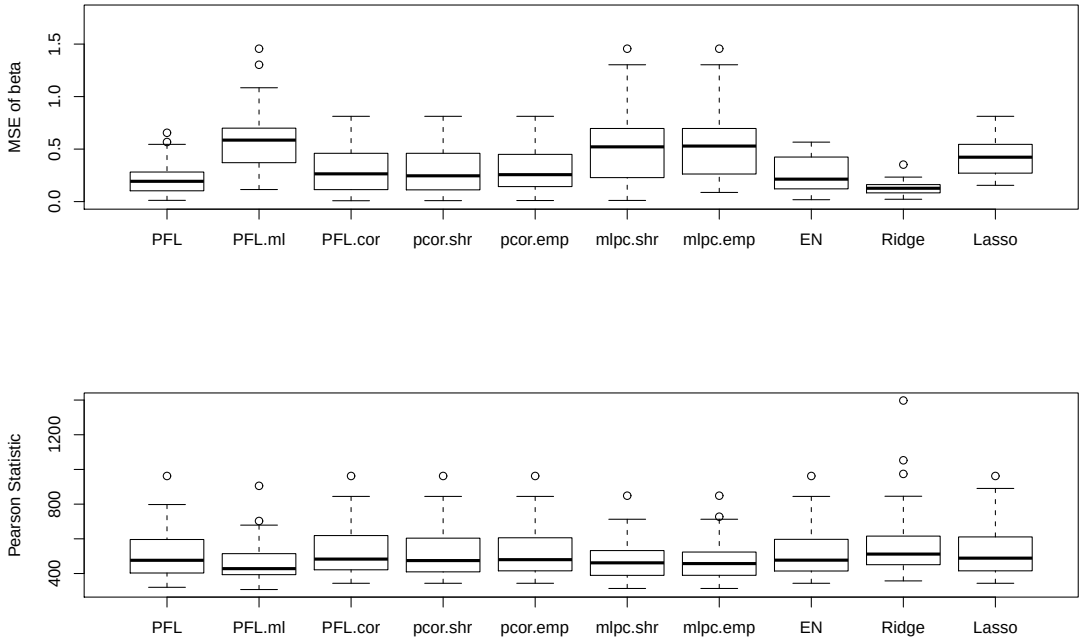

Figure C.18.: Boxplots of the pearson statistic on the test data set and MSE of $\boldsymbol{\beta}$ for the fifth simulation setting

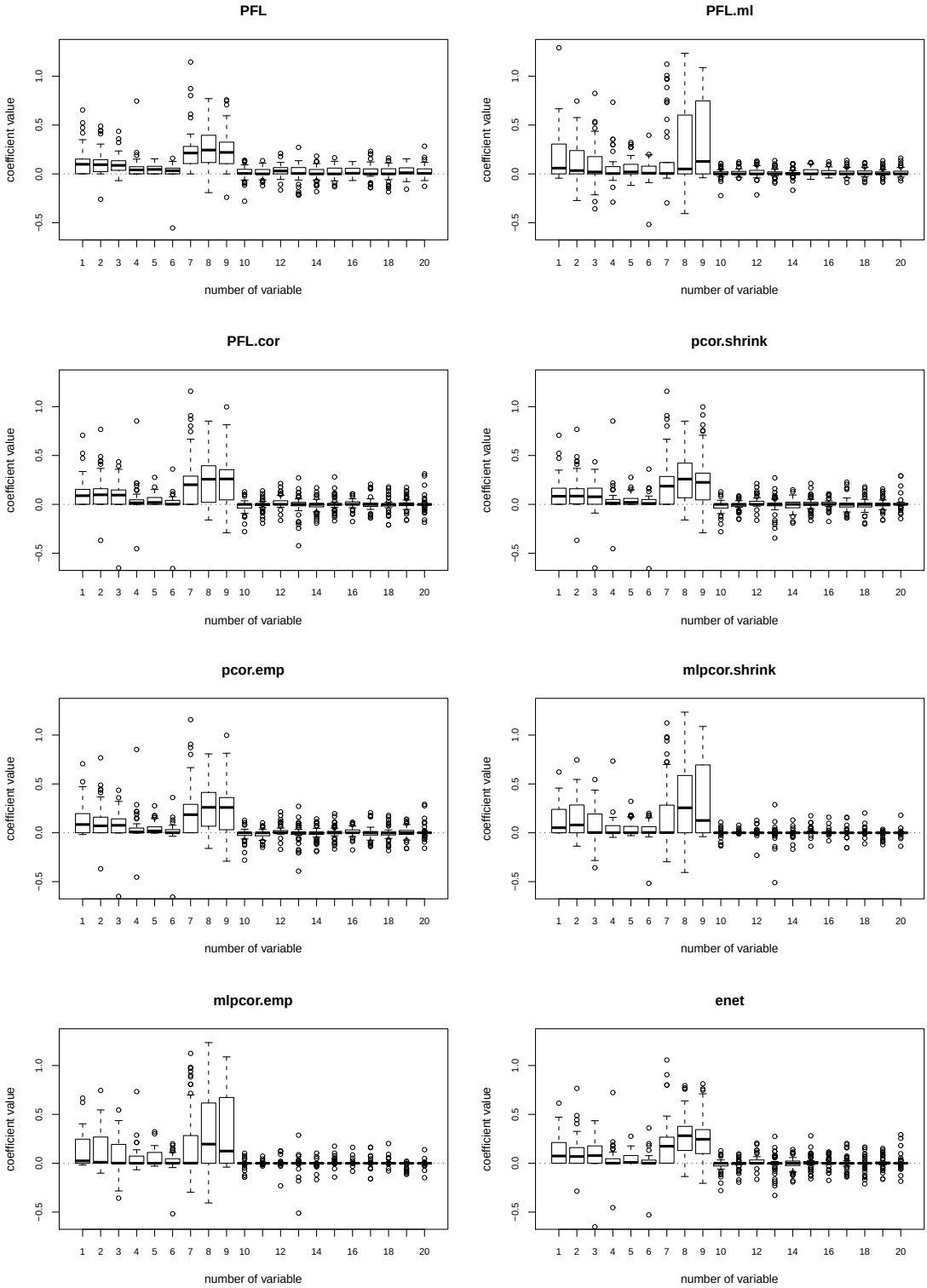

Figure C.19.: Boxplots of the predictors for the fifth simulation setting

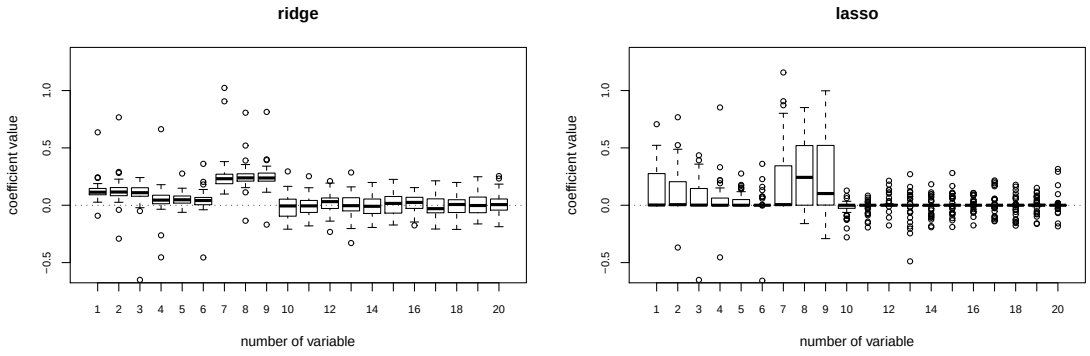

Figure C.20.: Boxplots of the predictors for the fifth simulation setting

Bibliography

[Bre96] Leo Breiman. Heuristics of instability and stabilization in model selection. *The Annals of Statistics*, 24(6), 1996. 8

[EHJT04] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32:407–499, 2004. 5, 12, 13, 15, 73

[ET98] Bradley Efron and Robert Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, London, 1998. 37

[FHT09] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. *glmnet: Lasso and elastic-net regularized generalized linear models*, 2009. R package version 1.1-4, `http://CRAN.R-project.org/package=glmnet`. 74

[FKL07] Ludwig Fahrmeir, Thomas Kneib, and Stefan Lang. *Regression: Modelle, Methoden und Anwendungen*. Springer, Berlin, 1. edition, 2007. 8

[FL01] Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96:1348–1360, 2001. 29

[FT01] Ludwig Fahrmeir and Gerhard Tutz. *Multivariate Statistical Modelling Based on Generalized Linear Models*. Springer, New York, 2. edition, 2001. 25, 26, 28, 56

[HK70] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970. 8, 9

[HTF09] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2. edition, 2009. 13, 14

[LVTM09] Justin Lokhorst, Bill Venables, Berwin Turlach, and Martin Maechler. *lasso2: L1 constrained estimation aka 'lasso'*, 2009. R package version 1.2-10, `http://www.maths.uwa.edu.au/~berwin/software/lasso.html`. 39, 74

[LW04] Olivier Ledoit and Michael Wolf. Honey, I shrunk the sample covariance matrix. *Journal of Portfolio Management*, 30(4):110–119, 2004. 19

[ORS07] Rainer Opgen-Rhein and Korbinian Strimmer. Accurate ranking of differentially expressed genes by a distribution-free shrinkage approach. *Statistical Applications in Genetics and Molecular Biology*, 6(9), 2007. 19

[Pet09] Sebastian Petry. *Shrinkage regression with polytopes*, 2009. `http://www.statistik.lmu.de/~petry/Hoehenried.pdf`. 5, 16, 20, 22

[PH07] Mee Young Park and Trevor Hastie. *glmpath: L1 Regularization path for generalized linear models and cox proportional hazards model*, 2007. R package version 0.94, `http://CRAN.R-project.org/package=glmpath`. 74

[R D09] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2009. ISBN 3-900051-07-0, `http://www.R-project.org`. 36

[Rei06] Florian Reithinger. *Zusammenhangsstrukturen*, 2006. Vorlesungsskript, Multivariate Verfahren SS06, `http://www.statistik.lmu.de/~flo/ss06/strukturen.pdf`. 18

[SORS09] Juliane Schäfer, Rainer Opgen-Rhein, and Korbinian Strimmer. *Efficient estimation of covariance and (partial) correlation*, 2009. R package version 1.5.3, `http://CRAN.R-project.org/package=corpcor`. 20

[SS05] Juliane Schäfer and Korbinian Strimmer. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4(32), 2005. 19

[Tib96] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, 58:267–288, 1996. 5, 9, 10

[Tib09] Robert Tibshirani. *Functions for the book "An Introduction to the Bootstrap"*, 2009. R package version 1.0-22, `http://CRAN.R-project.org/package=bootstrap`. 37

[Tou03] Helge Toutenburg. *Lineare Modelle: Theorie und Anwendungen*. Physica-Verlag, Heidelberg, 2. edition, 2003. 8

[TSR+05] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society B*, 67:91–108, 2005. 15, 16

[TU09] Gerhard Tutz and Jan Ulbricht. Penalized regression with correlation based penalty. *Statistics and Computing*, 19:239–253, 2009. 17, 37

[Ulb10a] Jan Ulbricht. *lqa: Local quadratic approximation*, 2010. R package version 1.0-2, `http://CRAN.R-project.org`. 36, 73

[Ulb10b] Jan Ulbricht. *Variable selection in generalized linear models*. Dissertation, Ludwig-Maximilians-Universität, München, 2010. 5, 29, 30, 32, 33, 73

[VR02] William N. Venables and Brian D. Ripley. *Modern Applied Statistics with S*. Springer, New York, 4. edition, 2002. 39

[Wei85] Sanford Weisberg. *Applied Linear Regression*. Wiley, New York, 2. edition, 1985. 13, 16

[Whi90] Joe Whittaker. *Graphical Models in Applied Multivariate Statistics*. John Wiley, Chichester, 1990. 18

[ZH05] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society B*, 67:301–320, 2005. 5, 11, 20, 35, 73

[ZH08] Hui Zou and Trevor Hastie. *elasticnet: Elastic-Net for Sparse Estimation and Sparse PCA*, 2008. R package version 1.0-5, `http://www.stat.umn.edu/~hzou`. 39

[Zou06] Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006. 17

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 25. Februar 2010 Claudia Flexeder