

# Deselection of base-learners for statistical boosting—with an application to distributional regression

Statistical Methods in Medical Research  
2022, Vol. 31(2) 207–224  
© The Author(s) 2021  
Article reuse guidelines:  
sagepub.com/journals-permissions  
DOI: 10.1177/09622802211051088  
journals.sagepub.com/home/smm



Annika Strömer<sup>1</sup> , Christian Staerk<sup>1</sup>, Nadja Klein<sup>2</sup>,  
Leonie Weinhold<sup>1</sup>, Stephanie Titze<sup>3</sup> and Andreas Mayr<sup>1</sup> 

## Abstract

We present a new procedure for enhanced variable selection for component-wise gradient boosting. Statistical boosting is a computational approach that emerged from machine learning, which allows to fit regression models in the presence of high-dimensional data. Furthermore, the algorithm can lead to data-driven variable selection. In practice, however, the final models typically tend to include too many variables in some situations. This occurs particularly for low-dimensional data ( $p < n$ ), where we observe a slow overfitting behavior of boosting. As a result, more variables get included into the final model without altering the prediction accuracy. Many of these false positives are incorporated with a small coefficient and therefore have a small impact, but lead to a larger model. We try to overcome this issue by giving the algorithm the chance to deselect base-learners with minor importance. We analyze the impact of the new approach on variable selection and prediction performance in comparison to alternative methods including boosting with earlier stopping as well as twin boosting. We illustrate our approach with data of an ongoing cohort study for chronic kidney disease patients, where the most influential predictors for the health-related quality of life measure are selected in a distributional regression approach based on beta regression.

## Keywords

Beta regression, generalized additive models for location, scale, and shape, model-based boosting, variable selection, earlier stopping

## Introduction

In modern biostatistics, model building and variable selection have become increasingly important, particularly in the context of applications in high-dimensional data settings, where the number of potential predictors  $p$  is larger compared to the sample size ( $p \gg n$ ).<sup>1</sup> Important examples include genetic or molecular data (e.g. Chen et al.<sup>2</sup>; Choi et al.<sup>3</sup>), but also in more classical clinical studies one often aims to obtain a relatively sparse model with good prediction accuracy including only the most relevant variables (e.g. Steyerberg and Vergouwe<sup>4</sup>; Sauerbrei et al.<sup>5</sup>).

Component-wise gradient boosting<sup>6</sup> provides a framework to handle this, with the key features of variable selection and the possibility to manage high-dimensional data problems. In combination with regression-type base-learners,<sup>7</sup> it is able to maintain the usual interpretability of statistical regression models—equivalent to the ones that were estimated using classical penalized likelihood or Bayesian inference. Statistical boosting provides a large flexibility due to the modular nature

<sup>1</sup>Department of Medical Biometrics, Informatics and Epidemiology, Faculty of Medicine, University of Bonn, Germany

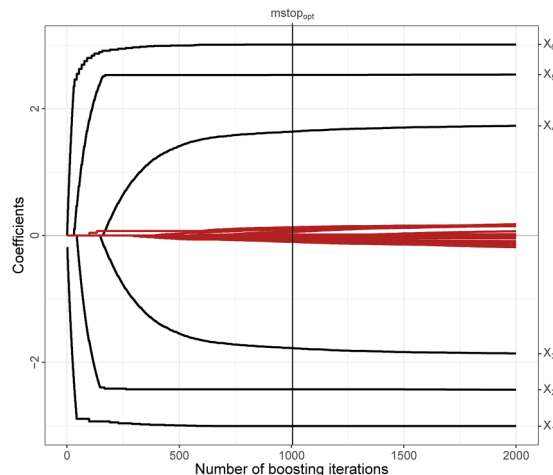
<sup>2</sup>Emmy Noether Research Group in Statistics and Data Science, Humboldt-Universität zu Berlin, Germany

<sup>3</sup>Department of Nephrology and Hypertension, FAU Erlangen-Nuremberg, Germany

## Corresponding author:

Annika Strömer, Department of Medical Biometrics, Informatics and Epidemiology, Faculty of Medicine, University of Bonn, Venusberg-Campus I, 53127 Bonn, Germany.

Email: stroemer@imbie.uni-bonn.de



**Figure 1.** Coefficient paths along the number of boosting iterations for a simulated data set with  $n = 500$  observations and  $p = 50$  variables which were simulated from a multivariate normal distribution with Toeplitz covariance structure and a correlation of 0.9. Only variables  $X_1, \dots, X_6$  are *informative* with true coefficients  $\beta_{\text{inf}} = (-3, -2.5, -2, 2, 2.5, 3)^T$ . The coefficient paths for the *non-informative* variables are colored red. The optimal stopping iteration  $m_{\text{stop}_{\text{opt}}} = 1004$  was determined by 10-fold cross-validation.

of the approach: any type of base-learner (linear models, splines, spatial models) can be combined with any type of convex loss function.<sup>8</sup>

Despite these advantages, in some applications the algorithm tends to select too many variables. This often occurs for rather low-dimensional settings with relatively large sample sizes ( $p < n$ ), where irrelevant base-learners often get selected with a very small effect size. This is associated with slow overfitting and thus with a higher number of boosting iterations  $m_{\text{stop}}$  which lead to a larger set of selected variables. For example, in a recent beta-regression analysis on the health-related quality of life (QoL) in  $n = 3522$  chronic kidney disease patients, statistical boosting selected 33 out of  $p = 54$  potential variables.<sup>9</sup>

As an illustration, Figure 1 displays the coefficient paths of component-wise boosting with the squared error loss in the context of linear regression for a simulated data set in which only the first six variables  $X_1, \dots, X_6$  are informative. One can observe that the estimated coefficients of the six informative variables are the largest in absolute values, while several non-informative variables are incorporated with small coefficient sizes around zero. Therefore, the selected non-relevant variables have only a minor impact on the predictive performance but lead to a larger model with difficult interpretation.

Bühlmann and Hothorn<sup>10</sup> tried to overcome this issue and extended the classical boosting approach to a two-stage design called twin boosting, which was inspired by the adaptive lasso (see Zou<sup>11</sup>). The first stage consists of a classical boosting algorithm. The second stage is similar to the first, with the difference that variables that have not been selected are excluded; furthermore, variables that have been selected in the first stage receive weights based on the size of their coefficients, making it more likely that the important variables will be selected again in the second stage. Other approaches aiming to increase the sparsity of statistical boosting focus on reducing the number of iterations  $m_{\text{stop}}$ : for example, the one standard-error rule was originally considered by Breiman et al.<sup>12</sup> in the context of random forests and does not select the optimal tuning parameter regarding prediction accuracy, but in case of boosting the smallest  $m_{\text{stop}}$  that is still in the margin of one standard error from the minimum risk. Ellenbach et al.<sup>13</sup> further extended this approach (*RobustC*) to obtain a less complex prediction rule that is less affected by the characteristics of the resampling scheme compared to the one standard-error rule. A potential disadvantage of approaches that lead to earlier stopping is that they suffer from the side-effect of inducing also a higher amount of shrinkage. This additional shrinkage of selected effect estimates might not necessarily lead to a better performance (cf. Van Calster et al.<sup>14</sup>).

Here we propose a general procedure to enhance the sparsity of statistical boosting models, where the final selection of variables is based on the risk reduction resulting from the individual updates of the corresponding base-learners. With this approach, we exclude those base-learners (and their corresponding variables) from the prediction model which attributed only slightly to the total risk reduction. As an alternative to earlier stopping of the boosting algorithm—that is moving “horizontally” on the regularization paths—we consider the individual contributions of different variables after a particular number of boosting iterations. The benefits of this “vertical” view on regularization paths have also recently been discussed in the context of other regularization methods such as the thresholded Lasso<sup>15; 16</sup> including the possibility of deselecting noise variables which are included “early” on the regularization paths. Furthermore, we directly enforce the sparsity of the

final models without unnecessarily increasing the amount of shrinkage on effect estimates. We illustrate the proposed method with the selection of predictors for the health-related QoL data of the German Chronic Kidney Disease Study (GCKD). We compare our results to a previous analysis of these data<sup>9</sup> which partly motivated the new methodological development. With the new deselection approach, we are able to select much sparser models while still yielding a similar prediction performance.

The remainder of the paper is structured as follows. In Section ‘Methods’, we introduce the new approach for an improved variable selection and consider alternative methods for achieving sparser models. In Section ‘Simulation study’, we compare these methods via simulated data under various conditions for different models. Finally, we apply our new approach to the QoL data and present the results in Section ‘Quality of life of chronic kidney disease patients’. Conclusively, Section ‘Discussion and conclusion’ summarizes our findings and discusses future research questions.

## Methods

### Model-based boosting

Boosting was first established in the context of machine learning<sup>17; 18</sup> and was later extended to fit statistical models.<sup>19; 20</sup> Statistical boosting algorithms<sup>21; 22</sup> can be used to analyze high-dimensional data problems, in which classical inferential methods are no longer applicable (e.g. least squares method for linear regression models). Furthermore, boosting yields data-driven variable selection and shrinkage of effect estimates.<sup>6</sup>

The model fitting is carried out by iteratively minimizing the empirical risk of an appropriate loss function. This loss defines the regression problem and needs to be specified in advance. In generalized linear models (GLMs) and generalized additive models (GAMs), the loss function corresponds to the negative log-likelihood of the outcome distribution. For classical linear regression models, for example, we minimize the squared error ( $L_2$  loss), which corresponds to maximizing the likelihood of a Gaussian distribution. Different effect types can be determined for each covariate (e.g. linear or smooth effects), which reflect the type of influence the variable has in the model. These underlying functions are called base-learners; in the simplest case they are univariate linear models representing linear effects. In each iteration, the negative gradient of the loss function is determined and every base-learner is separately fitted to the negative gradient. Afterwards, only the best performing base-learner is selected (i.e. the base-learner that best fits the negative gradient) and the corresponding estimated effect is multiplied by a small fixed step size (default is  $\nu = 0.1$ ) before it is included in the model. Due to the selection of single base-learners in each iteration, the algorithm carries out variable selection. This process is repeated until the number of boosting iterations  $m_{\text{stop}}$  is reached, whereby every base-learner can be selected several times. In the classical boosting algorithm, every base-learner that was once included in the model cannot be deselected.<sup>23</sup>

The number of boosting iterations is the main tuning parameter and can be selected, for example, by cross-validation or other resampling techniques. The optimization of the stopping iteration—also referred to as *early stopping*—is crucial to prevent overfitting and to favor the sparsity of the resulting model. The smaller  $m_{\text{stop}}$ , the fewer variables are included in the final model as only one base-learner is updated in each iteration. Additionally, early stopping typically improves the prediction accuracy and leads to shrinkage of effect estimates.<sup>24</sup>

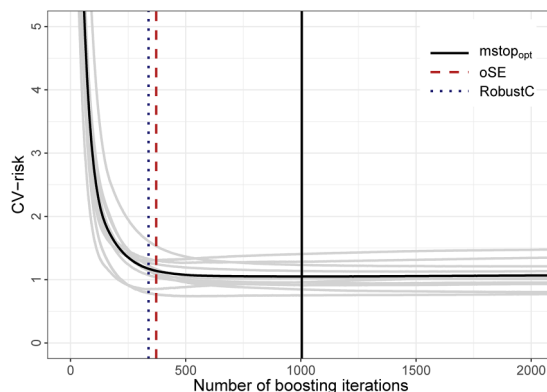
### Earlier stopping strategies

Due to the influence of the number of boosting iterations  $m_{\text{stop}}$  on the variables finally selected by the algorithm, one approach to achieve sparser models is to enforce *earlier* stopping of the algorithm, that is, selecting a smaller  $m_{\text{stop}}$ . With this approach it is assumed that variables that are updated in early iterations of the algorithm have a greater influence on the prediction of the model than variables that are added later to the model. Typically, classical early stopping selects the stopping iteration  $m_{\text{stop\_opt}}$  that leads to the smallest (optimal) cross-validated prediction risk (CV).

The one standard error rule (oSE) is one approach to enforce earlier stopping and has already been used in context of penalized regression and regression trees.<sup>12; 25</sup> With this approach, the tuning parameter  $m_{\text{stop}}$  is chosen as the smallest iteration for which the CV is within one standard error of the minimal CV (cf. Friedman et al.<sup>25</sup>; Hastie et al.<sup>26</sup>)

$$\text{CV}(m_{\text{stop}}) \leq \text{CV}(m_{\text{stop\_opt}}) + \text{se}(\text{CV}(m_{\text{stop\_opt}})).$$

The minimal cross-validated predictive risk  $\text{CV}(m_{\text{stop\_opt}})$  corresponds to the CV of the optimal stopping iteration. Furthermore,  $\text{se}(\text{CV}(m_{\text{stop\_opt}}))$  represents the standard error of the minimum over the CV folds. Consequently, this method has dependencies on the number of CV folds and the sample size.



**Figure 2.** Application of the oSE and RobustC on the cross-validated estimation of the empirical risk with 10-fold cross-validation. The vertical solid black line reflects the optimal stopping iteration via cross-validation ( $m_{\text{stop}} = 1004$ ), the red dashed one displays the oSE ( $m_{\text{stop}} = 372$ ) and the blue dotted one RobustC ( $m_{\text{stop}} = 339$ ) with  $c_{rC} = 1.1$ .

Based on the idea of the oSE approach, Ellenbach et al.<sup>13</sup> proposed an alternative more robust approach, called RobustC. Here the smallest  $m_{\text{stop}}$  is chosen, whose CV is still within a range of a fixed additional tuning parameter  $c_{rC}$  multiplied with the minimum CV

$$CV(m_{\text{stop}}) \leq c_{rC} \times CV(m_{\text{stop, opt}}).$$

Ellenbach et al.<sup>13</sup> suggested the values  $c_{rC} \in \{1, 1.1, 1.3, 1.5, 2\}$  for the case of a binary outcome. The authors aimed for a less complex predictive rule and for choosing a robust tuning parameter, which is essential in cross-study predictions.<sup>13</sup>

Considering the example from the introduction, Figure 2 shows the CV-risk for 10-fold cross-validation with 2000 boosting iterations. The vertical solid black line shows the optimal stopping iteration and corresponds to the minimum average risk over the 10-fold cross-validation samples. The vertical dashed red line is the stopping iteration which yields the oSE, while the blue dotted line corresponds to the optimal iteration according to RobustC. One can observe that the stopping iterations of the earlier stopping strategies are less than half as large as the original  $m_{\text{stop}}$ .

A further alternative approach to obtain sparser models is probing. The idea is to extend the data set by random noise variables, the so-called probes, which are randomly shuffled versions of the originally observed variables. The algorithm stops when the first probe is selected. For more details on this approach see Thomas et al.<sup>27</sup>

## Deselection of base-learners with a small risk reduction

Several other approaches have been developed to enhance the sparsity of boosting models (e.g. Hofner et al.<sup>28</sup>; Thomas et al.<sup>27</sup>). Most of them focus on the selection step in the algorithm, or on the tuning of the stopping iteration  $m_{\text{stop}}$  (section ‘Earlier stopping strategies’). Our new procedure is based on *actively* deselecting variables that have been selected by the algorithm, but result in only minor importance regarding the predictions of the model.

We address this issue with an approach that aims at eliminating variables with a small impact and directly enforce the sparsity of the model. The general idea is to first apply a standard boosting algorithm with early stopping via cross-validation or resampling techniques; then, we determine the variables selected by boosting with a minor importance for the model and deselect those components. Afterwards, we boost again incorporating only the selected variables that survived as candidate variables. In this context, our procedure shows analogies to the twin boosting approach.<sup>10</sup> In our deselection procedure, we consider the risk reduction as a measure for variable importance and deselect those variables that only represent a small percentage of the total risk reduction.

The risk reduction by base-learner  $j$  after  $m_{\text{stop}}$  boosting iterations can be defined as the *attributable* risk reduction  $R_j$

$$R_j = \sum_{m=1}^{m_{\text{stop}}} I(j = j^{*[m]}) (r^{[m-1]} - r^{[m]}), \quad j = 1, \dots, p, \quad (1)$$

where  $I$  denotes the indicator function and  $j^{*[m]}$  is the selected base-learner in iteration  $m$ . Furthermore,  $r^{[m-1]} - r^{[m]}$  represents the risk reduction in iteration  $m$ , for risks  $r^{[m]}$  and  $r^{[m-1]}$  at iterations  $m$  and  $m - 1$ . For a given threshold  $\tau \in (0, 1)$ , we

deselect base-learner  $j$  if

$$R_j < \tau \cdot (r^{[0]} - r^{[m_{\text{stop}}]}), \quad (2)$$

where  $r^{[0]} - r^{[m_{\text{stop}}]}$  represents the total risk reduction and  $R_j$  denotes the attributable risk reduction of base-learner  $j$ .

A schematic overview of the proposed procedure can be found in Box 1. Step 1 of the procedure consists of the initial boosting for which the coefficient paths are shown in Figure 3 (left), corresponding to the simulation example discussed earlier (see Figures 1 and 2). Overall, 23 variables (of the 50 variables) were selected (shown as horizontal red and black paths) after  $m_{\text{stop}} = 1004$  boosting iterations which were tuned by 10-fold cross-validation (indicated by the vertical black line). For the deselection in Step 2, the attributable risk reduction along the iterations is shown for each individual base-learner in the central plot of Figure 3). To illustrate the effect of the deselection step of the proposed method, consider the thresholds  $\tau = 0.01$  (horizontal dashed line) and  $\tau = 0.1$  (horizontal dotted line). Here, it can be observed that our deselection procedure is fundamentally different to earlier stopping approaches discussed in section ‘Earlier stopping strategies’, as the choice of the threshold for the deselection corresponds to a vertical view on the individual risk reductions after a given number of boosting iterations (see central plot of Figure 3; on the other hand, earlier stopping simply corresponds to a horizontal shift on the usual regularization paths of boosting (see Figure 2).

In the following, we consider a threshold value of  $\tau = 0.01$  and accordingly deselect those variables which contribute less than 1% to the total risk reduction. The black paths correspond to the variables included in the model after applying the deselection approach, while the red paths do not cross the 1% line and the corresponding variables are deselected from the model. We can observe that these variables contribute only slightly to the risk reduction and are incorporated with a coefficient size around zero in the initial boosting model (as shown in Figure 3, left). In this example, the deselection approach with threshold  $\tau = 0.01$  deselects all noise variables from the model, but not the signal variables  $X_1, \dots, X_6$ . Variables  $X_1, X_2, X_5$ , and  $X_6$  have by far the greatest individual contributions to the total risk reduction; however, variables  $X_3$  and  $X_4$  also exceed the 1% threshold (but not the 10% threshold). After the deselection step, we boost again (Step 3) with only the remaining variables and receive the final model (see Figure 3, right) which contains here exclusively the six informative variables.

### Box 1: Deselection procedure

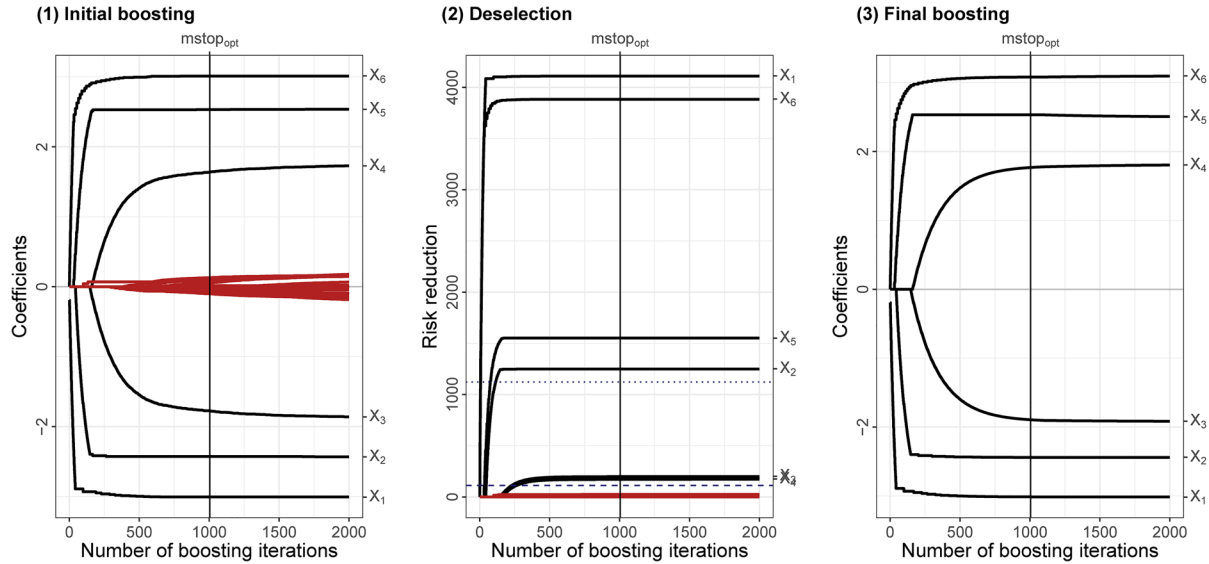
1. Initial boosting:
  - Early stopping: Tuning of  $m_{\text{stop}}$  based on cross-validation or resampling.
2. Deselection:
  - Identify the base-learners with the smallest impact on the risk reduction according to  $R_j < T \cdot (r^{[0]} - r^{[m_{\text{stop}}]})$  (2) and remove them from the model.
3. Final boosting:
  - Boost again with the remaining variables and the  $m_{\text{stop}}$  of step 1.

### Deselection of base-learners for distributional regression

In classical statistical models, the relationship between a response variable and covariates is most often modeled only based on the expected value. For example, a generalized additive model (GAM)<sup>29</sup> in which the conditional mean  $\mu = \mathbb{E}(y|x)$  relates to an additive predictor  $\eta$  via a link-function  $g$ , is given by

$$g(\mu) = \eta(x) = \beta_0 + \sum_{j=1}^p f_j(x_j)$$

with the intercept  $\beta_0$  and the additive effects  $f_j$  for the covariates  $X_j$  with  $j = 1, \dots, p$  (including linear, smooth or random effects). Consider, for example, a Gaussian distribution, which has two parameters: the expected value  $\mu$  and the scale parameter  $\sigma$ . In a classical GAM, we assume that  $\sigma$  is fixed and only model the mean parameter  $\mu$  in terms of the covariates.



**Figure 3.** Simulation example for the deselection procedure (see Box 1). The left plot shows the coefficient paths of the initial boosting. The central plot displays the attributable risk reduction for the individual variables, together with the 1% threshold (dashed line) and 10% threshold (dotted line) of the total risk reduction. The coefficient paths of the right plot correspond to the final boosting.

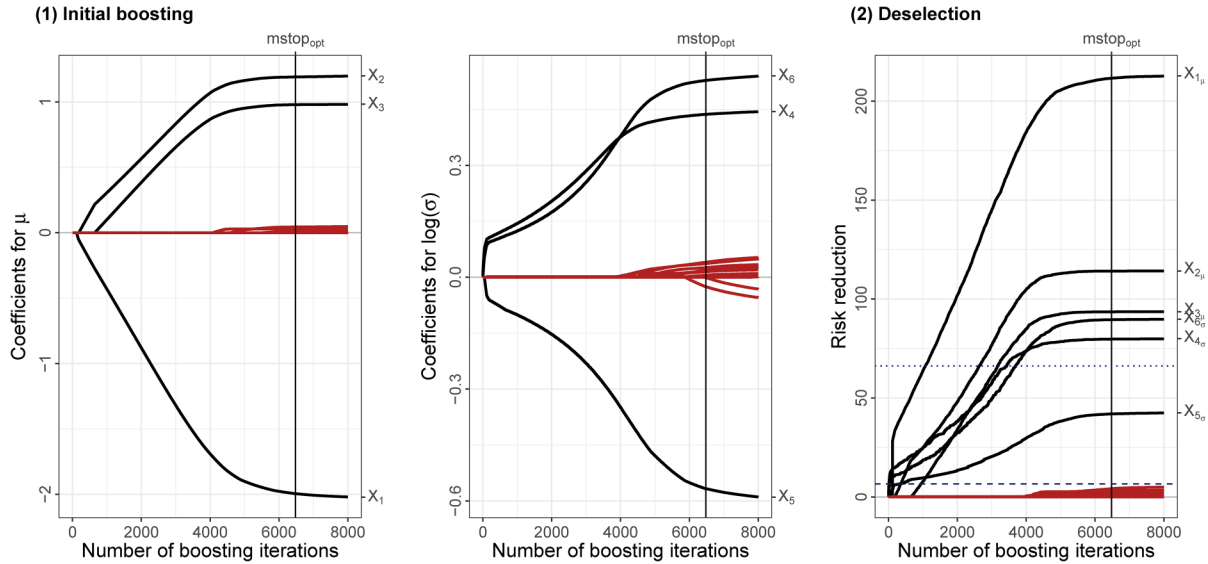
In some cases, this may lead to an overly restrictive point of view, for example, in the presence of heteroscedasticity. In addition, skewness and kurtosis may be large so that more complex non-symmetric distributions are required where potentially skewness or higher order moments could be modeled through covariates to obtain a more accurate model. Following this idea, GAMs have been extended to generalized additive models for location, scale and shape (GAMLSS) by Rigby and Stasinopoulos,<sup>30</sup> where a general parametric density  $P(y|\theta_1, \dots, \theta_K)$  with distributional parameters  $\theta_k$  can be employed. Here, each distribution parameter  $\theta_k$ , with  $k = 1, \dots, K$ , can be modeled by an additive predictor  $\eta_k$  depending on covariates. Furthermore, for each parameter  $\theta_k$ , we have parameter-specific link-functions  $g_k(\cdot)$  as well as parameter-specific covariates  $x_{k1}, \dots, x_{kp_k}$ . In general, the linear predictors in a GAMLSS for  $K$  distributional parameters can be written as follows:

$$g_k(\theta_k) = \eta_k = \beta_{0k} + \sum_{j=1}^{p_k} f_{jk}(x_{kj}), \quad k = 1, \dots, K,$$

where  $\beta_{0k}$  are the intercepts for the distributional parameters  $\theta_k$  and  $f_{jk}$  denote the functions of the effect of variable  $X_j$  on the parameter  $\theta_k$ .

GAMLSS can also be fitted via statistical boosting with the package **gamboostLSS**.<sup>31</sup> As in the classical setting of boosting GAMs, the main tuning parameter is the stopping iteration  $m_{stop}$  which controls shrinkage of effect estimates and variable selection. Here, we focus on a non-cyclical boosting approach,<sup>32</sup> which performs in every iteration only the overall best-performing update among the available candidate variables (base-learners) and distribution parameters. So the term *component-wise* boosting in this context does not only refer to the components of  $X$ , but also to the components of the parameter space  $\theta_1, \dots, \theta_K$  of the corresponding likelihood. To receive the overall best performing base-learner, the empirical risk (the negative log-likelihood) of the best fitted base-learner is determined for each distribution parameter and then compared across the different dimensions.

The updates are independent for the parameters and each additive predictor may depend on different variables with the guarantee of data-driven variable selection in every submodel: Figure 4 displays the estimated coefficient paths in a linear Gaussian location-scale model with distributional parameters  $\mu$  (left) and  $\sigma$  (center). The data set consists of  $n = 500$  observations and  $p = 20$  variables, where the first three variables  $X_1, X_2, X_3$  are informative for the mean parameter  $\mu$  with  $\beta_{\mu_{inf}} = (-2, 1.25, 1)^T$ , while variables  $X_4, X_5, X_6$  are informative for the scale parameter  $\sigma$  with  $\beta_{\sigma_{inf}} = (0.5, -0.5, 0.5)^T$ . All other variables are non-informative with  $\beta_{\mu} = 0$  and  $\beta_{\sigma} = 0$ . The explanatory variables were simulated from a multivariate normal distribution with Toeplitz covariance structure and a correlation of  $\rho = 0.5$ . The optimal number of boosting iterations is  $m_{stop} = 6479$ , optimized via 10-fold cross-validation. Note that in every iteration only a single component is updated for one parameter.



**Figure 4.** Simulation example for gamboostLSS with three informative variables for  $\mu$  ( $X_1, X_2, X_3$ ) and three for  $\sigma$  ( $X_4, X_5, X_6$ ). The first two plots display the coefficient paths (for  $\mu$  (left) and  $\sigma$  (center)) and the third plot shows the attributable risk reductions for the individual variables for both distributional parameters together with the 1% (dashed) and 10% line (dotted) of the total risk reduction. The variables corresponding to the black coefficient path are still in the model after deselection with  $\tau = 0.01$ .

We notice that in the first iterations, components of parameter  $\sigma$  were more often updated, which can be observed by the increase of the coefficient sizes for the variables  $X_4, X_5, X_6$  in the first iterations. In total, the boosting model contains 18 of the 50 variables, with 11 variables selected for  $\mu$  and 14 variables selected for  $\sigma$  (where seven variables were selected for both  $\mu$  and  $\sigma$ ). Hence, additional variable selection can be advantageous to obtain sparser and thus more interpretable models, which only include the informative variables.

Using equation (1 and considering  $j = 1, \dots, \sum p_k$ , the risk reduction in a GAMLSS for component  $j$  can be defined similar as before. For the deselection of variables with a low impact on the risk reduction for distributional regression, we consider the distributional parameters together, where each parameter can depend on different variables. Analogous to equation (2, we deselect component  $j$  if

$$R_j < \tau \cdot (r^{[0]} - r^{[m_{\text{stop}}]})$$

with fraction  $\tau \in (0, 1)$  and total risk reduction  $r^{[0]} - r^{[m_{\text{stop}}]}$ . Note that the deselected components may arise from different distributional parameters and that with this definition, GAMs are included as a special case in the general formulation for a GAMLSS with  $p_k \equiv p$  and  $k = 1$ .

For the simulation example, the risk reduction of the variables for  $\mu$  and  $\sigma$  is shown in the right plot (Figure 4). As shown in Figure 3, the threshold value is chosen as  $\tau = 0.01$  (horizontal dashed line) and  $\tau = 0.1$  (horizontal dotted line). The black paths correspond to the variables remaining in the model after applying the deselection procedure (with  $\tau = 0.01$ ) for distributional regression and have by far the highest impact on the risk reduction. The deselection results in a model including only the six informative variables (instead of the 18 initially selected variables). For the choice of an appropriate value for the threshold parameter  $\tau$ , we examined different potential values observing the attributable risk reduction of the base-learner as shown in Figure 3 (second plot) and Figure 4 (third plot).

Considering Figure 3, the variables  $X_1, X_2, X_5$ , and  $X_6$  have the largest impact on the risk reduction. All of those variables remain in the model with a deselection threshold of 1% as well as the other two informative variables  $X_3$  and  $X_4$ . For the 10% boundary,  $X_3$  and  $X_4$  would not enter the model because of a smaller risk reduction. Even for the data example in Figure 4, 10% is not an appropriate choice, since the variables  $X_1, \dots, X_6$  have a noticeable impact on the risk reduction, but  $X_5$  would fall out at this limit. A threshold of 1% appears to be reasonable in the considered situation. However, in non-sparse situations, when many base-learners contribute only with a small risk reduction to the model, multiple signal variables may be deselected with a threshold of  $\tau = 0.01$ . This extreme scenario should be rare, and in such non-sparse data situations, enforcing variable selection might not be favorable in general.

An implementation of the enhanced variable selection approach is available at GitHub (<https://github.com/AnnikaStr/DeselectBoost>).

## Simulation study

To evaluate the performance of our new approach for different data settings, we conduct a simulation study focusing on the variable selection properties as well as the prediction accuracy in comparison with the methods for earlier stopping, described in Section ‘Earlier stopping strategies’.

Specifically, the questions to be investigated in the simulation study are as follows:

1. Is the direct deselection approach able to identify the truly informative variables (decreasing the number of false positive variables selected by classical boosting)?
2. How does the reduction in selected variables affect the prediction accuracy?
3. How does the new procedure perform in comparison to the earlier stopping strategies, e.g. oSE and RobustC?
4. What is an appropriate value for  $\tau$  in the proposed deselection approach?

## Settings

To examine those questions, different settings are considered: First, we start with classical mean regression models (linear, non-linear, and logistic regression) and afterwards, we consider the deselection approach in the context of distributional regression models.

For all simulations, the explanatory variables  $X_1, \dots, X_p$  were simulated from a multivariate normal distribution  $N(\mathbf{0}, \Sigma)$  with a Toeplitz covariance structure  $\Sigma_{ij} = \rho^{|i-j|}$  for  $1 \leq i, j \leq p$ , where  $\rho \in (0, 1)$  is the correlation between consecutive variables  $X_j$  and  $X_{j+1}$ . For an alternative block-wise covariance structure, see the corresponding results in Supplemental Material A.1. Overall, we considered two different dimensions of the data problem: (i) a low-dimensional setting ( $p < n$ ) with  $n = 500$  observations and  $p = 20$  variables and (ii) a high-dimensional setting ( $p > n$ ) with  $n = 500$  observations and  $p = 1000$  variables. In total, six of the included variables were informative (for the distributional regression, three for each parameter). Furthermore, a low-correlated scenario with  $\rho = 0.2$  and a high-correlated scenario with  $\rho = 0.8$  was considered for each setting. Additionally, we consider a variation of signal-to-noise ratios (SNRs) and the corresponding effect of different threshold values  $\tau$  with  $\text{SNR} \in \{0.15, 6, 14.64\}$  and  $\tau \in \{0.005, 0.0075, 0.01, 0.025, 0.05, 0.075, 0.1, 0.125\}$ .

For evaluation, we generated test data sets with 1000 observations from the same distribution as the training data sets. As in the illustrative examples, the number of boosting iterations was tuned via 10-fold cross-validation. The fixed step size is set to  $\nu = 0.1$  and was not varied in the simulation, considering that it does not largely affect the risk reduction as long as the step size is chosen reasonably small.<sup>33</sup> We additionally compared the deselection approach (with  $\tau = 0.01$ ) with the earlier stopping strategies, oSE and RobustC (additional comparison with probing is given in Supplemental Material A.2). The parameter value for RobustC is chosen as  $c_{rC} = 1.05$  for a continuous outcome variable and  $c_{rC} = 1.1$  for a binary response, following the recommendation of Ellenbach et al.<sup>13</sup>

For each setting, 100 simulation runs were conducted and the data sets were generated from the following models:

**Scenario A** (linear regression): The true linear model for the continuous outcome variable  $Y$  is given by

$$y = -2x_1 - 1.5x_2 - x_3 + x_4 + 1.5x_5 + 2x_6 + \epsilon,$$

with  $\epsilon \sim N(0, 1)$ . The base-learners correspond to simple linear models and the performance was assessed using the mean squared error of prediction (MSEP).

**Scenario B** (non-linear regression): The outcome variable  $Y$  was generated from the model

$$y = 1.5 \sin(x_1) + x_2 - 0.25x_3^2 - 0.25x_4 - x_5 - 1.5x_6 + \epsilon,$$

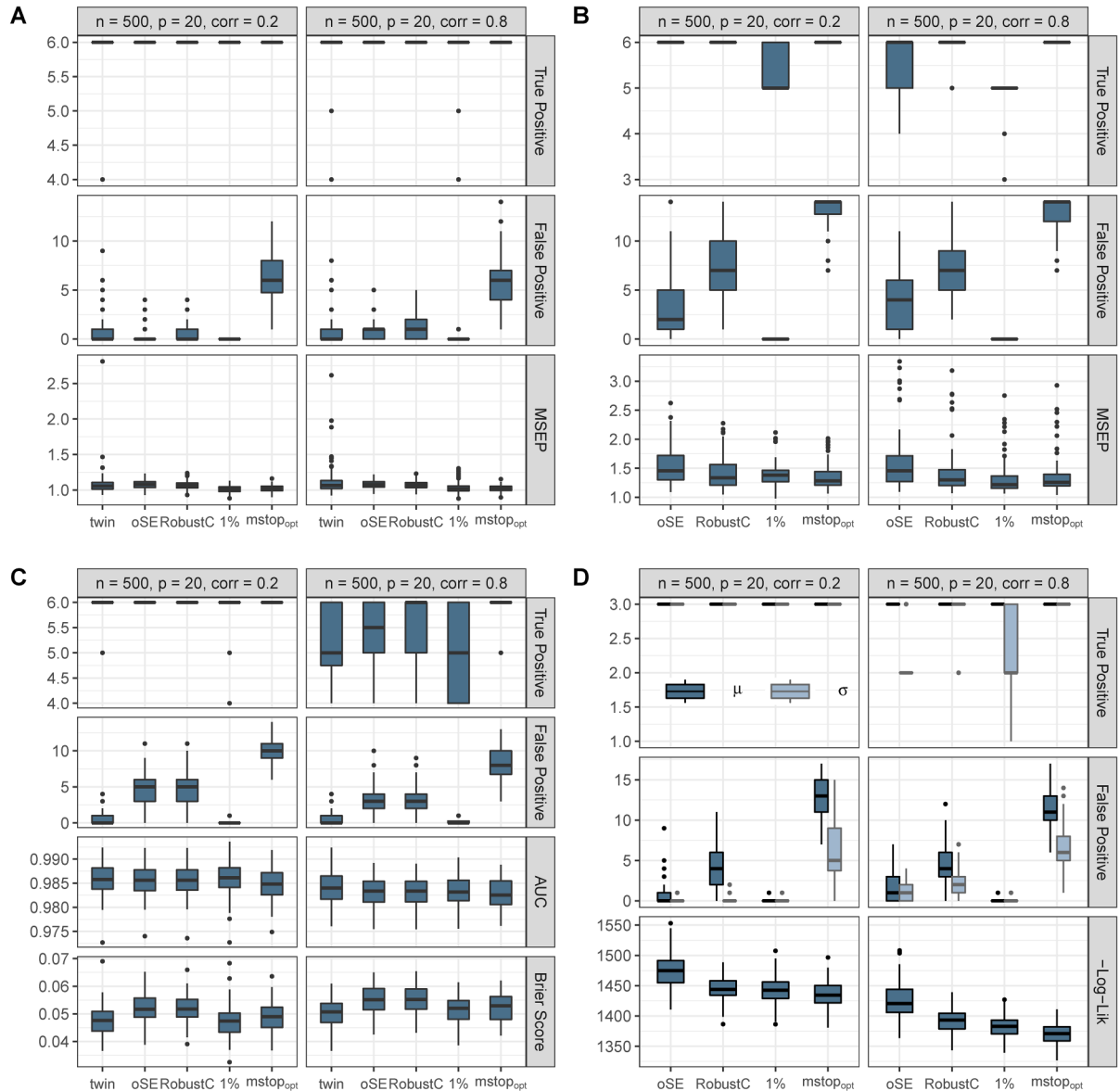
with  $\epsilon \sim N(0, 1)$ . Smooth P-splines were used as base-learners and the MSEP was used for evaluation.

**Scenario C** (logistic regression): The logistic regression model for covariates with only linear effects on the response was simulated according to

$$\log\left(\frac{\mathbb{P}(y = 1|x)}{\mathbb{P}(y = 0|x)}\right) = -5x_1 + -2.5x_2 - x_3 + x_4 + 2.5x_5 + 5x_6.$$

As evaluation criteria, the Brier score and the area under the curve (AUC) were analyzed on test data.





**Figure 5.** Low-dimensional setting: Comparison of the oSE, RobustC, the deselection approach with  $\tau = 0.01$  (1%) and the classical boosted model with  $m_{\text{stop\_opt}}$  regarding the true positives, false positives and the prediction performance for (A) linear regression, (B) non-linear regression, (C) logistic regression, and (D) distributional regression. Additional comparisons with twin boosting (twin) for the linear and logistic regression scenarios.

**Scenario D** (distributional regression): For the distributional regression model, we consider a Gaussian regression with expected value  $\mu$  and scale parameter  $\sigma$ . Both parameters depend on three different covariates

$$\begin{aligned} \mu &= -2x_1 + 1.25x_2 + x_3, \\ \log(\sigma) &= 0.5x_4 - 0.5x_5 + 0.5x_6. \end{aligned}$$

The boosting model was configured with simple linear models as base-learners and the performance was evaluated via the negative log-likelihood.

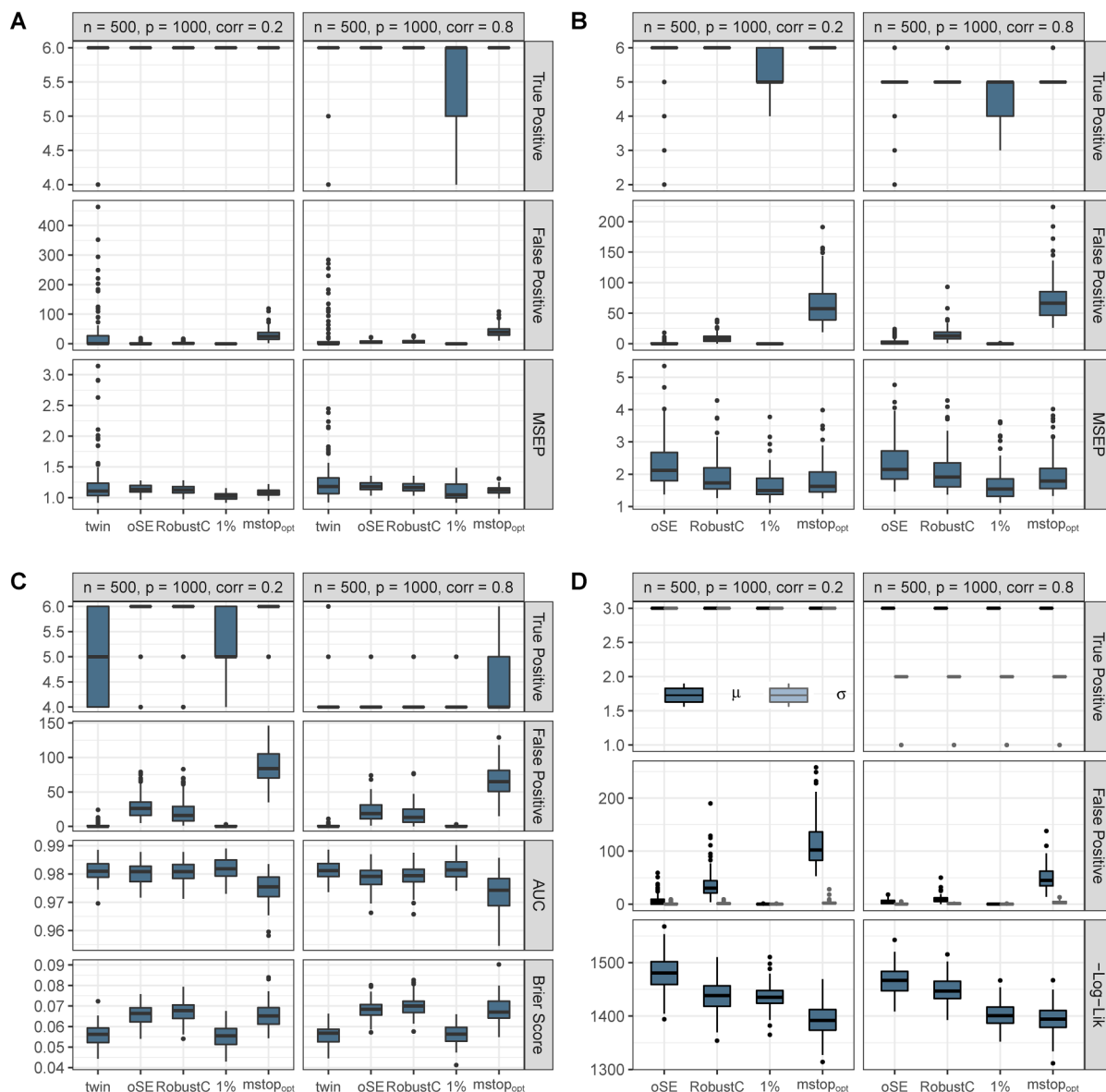
All simulations were conducted in the statistical computing environment **R**<sup>34</sup> using the add-on package **mboost**<sup>35</sup> for model-based boosting. The algorithm for fitting GAMLSS models via component-wise gradient boosting is implemented in **gamboostLSS**.<sup>31</sup> Twin boosting is implemented in the package **bst**.<sup>36</sup> The **R** code to reproduce the following simulation results can be found online at GitHub (<https://github.com/AnnikaStr/DeselectBoost>).

## Results

Figure 5 shows the results of the low-dimensional simulations regarding the four previously described models for the low- as well as the high-correlated settings, respectively. For each setting, the true positives, false positives, and the predictive performance for the respective model are shown for the earlier stopping strategies, the deselection procedure with  $\tau = 0.01$  and the classical boosted model.

In general, the two main strategies (earlier stopping and deselection) resulted in a reduction of false positives. In each of the four models, the fewest false positives were obtained with the proposed deselection procedure; more precisely, almost all false positives were deselected. For some models, one can observe that the selection of informative variables was slightly influenced by the earlier stopping and deselection approach, particularly for the high-correlated settings.

In comparison with classical boosting, the deselection procedure yielded comparable or slightly better predictive performances for simulated data based on Scenario A, Scenario B, and Scenario C. Furthermore, the earlier stopping strategies

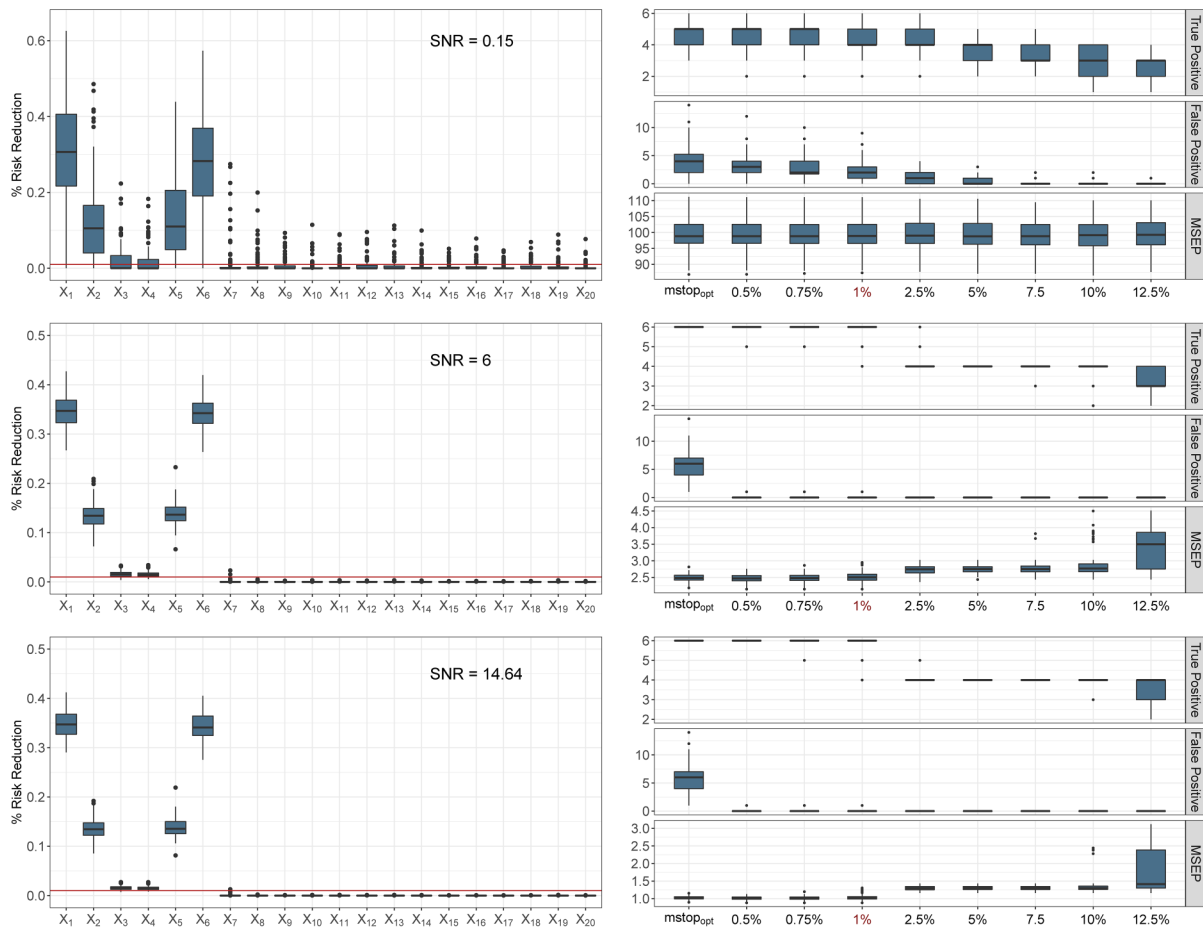


**Figure 6.** High-dimensional setting: Comparison of the oSE, RobustC, the deselection approach with  $\tau = 0.01$  (1%) and the classical boosted model with  $m_{stop\_opt}$  regarding the true positives, false positives and the prediction performance for (A) linear regression, (B) non-linear regression, (C) logistic regression, and (D) distributional regression. Additional comparisons with twin boosting (twin) for the linear and logistic regression scenarios.

usually performed not as well as our approach. Only the AUC in Scenario B is very similar. For distributional regression (Scenario D), the classical approach yielded the best results concerning the negative log-likelihood but contained a lot of non-informative variables for both distributional parameters. The deselection approach reduced the false positives almost completely and had only a slightly worse prediction performance.

Figure 6 presents the results of the high-dimensional setting. As in Figure 5, the true positives, false positives, and predictive performances are shown. For the high-correlated cases of Scenario B, Scenario C, and Scenario D, the classical boosting model had already difficulties to select all informative variables. Concerning Scenario C only four of the six true positives were selected on average. In comparison with the classical approaches, the earlier stopping and deselection approaches resulted in an average lower number of true positives. For the false positives, we can observe a noticeable reduction with the earlier stopping strategies, but the number of false positives reduced even more with the deselection procedure and the final models contained almost only informative variables. The greatest reduction can be observed for Scenario D where the classical approach contains 100 false positives on average for parameter  $\mu$ . After applying the deselection approach, the number of false positives decreased to almost zero with all informative variables still present. Due to the strong reduction of non-informative variables, in most cases, the deselection procedure showed a better predictive performance in comparison to earlier stopping and the classical boosting. Although in most of the simulation runs of Scenario C, not all informative variables were selected by the proposed deselection approach, it yielded a significantly lower Brier score and a better discriminatory power.

Furthermore, we compared the new deselection approach as well as earlier stopping strategies to twin boosting in the context of linear and logistic regression models (see Figures 5 and 6). Considering the results for twin boosting of Scenario A, the number of false positives was reduced (as for oSE, RobustC, and the new deselection procedure), but it shows larger variability, particularly in the high-dimensional setting. In one bootstrap sample, the model contained about 450 false positives (for



**Figure 7.** Relative risk reduction (in %) of each base-learner (left) and the variable selection and predictive performance for different  $\tau$  values (right) for the low-dimensional setting with correlation  $\rho = 0.8$  for increasing signal-to-noise ratios (from top with SNR = 0.15 to bottom with SNR = 14.64) for Scenario A.

low correlation). That is much more than we observed with the classical boosting approach, which had the maximum at about 100 selected non-informative variables. However, it should be noted that these different results for twin boosting are related to a different implementation. The highest decrease in false positives was observed for the deselection approach. The prediction accuracy was influenced by the outliers and also showed some higher MSE values for twin boosting. We obtained the best model for the deselection approach regarding the number of false positives as well as predictive performance. The results for logistic regression showed a slight reduction of the selected informative variables for each approach.

On average, twin boosting contained fewer false positives and had a better prediction accuracy than the earlier stopping strategies. Compared with the new deselection procedure, twin boosting tended to include more false positives, but was similar in terms of predictive performance. Here, twin boosting showed favorable properties in comparison to the linear regression model.

Finally, we investigate how the SNR affects the choice of the threshold parameter  $\tau$ . Figure 7 shows the results for Scenario A concerning different SNRs with  $\text{SNR} \in \{0.15, 6, 14.64\}$ . In this case, we consider the low-dimensional setting (for illustrative purposes) with a correlation of 0.8, where  $\text{SNR} = 14.64$  corresponds to the simulation setting presented before (results for a correlation of 0.2 and the high-dimensional settings are given in Supplemental Material A.3). On the left side, the relative risk reduction (in %) is depicted for each base-learner for the three different SNRs. The red horizontal line represents a threshold  $\tau$  of 1%. The right side shows the corresponding results of the true positives and false positives as well as the predictive performance for various  $\tau$  values. Overall, the relative risk reduction for all SNR values was very similar and the highest values always referred to informative variables, of which  $X_3$  and  $X_4$  showed the lowest risk reduction. The risk reduction for a SNR of 0.15 varied more and the non-informative variables showed a higher contribution to the risk reduction.

Considering the variable selection and predictive performance for various  $\tau$  thresholds, the true positives and false positives for a SNR of 6 and 14.64 were very similar over the different  $\tau$  values. For a SNR of 0.15, the classical boosting model had larger difficulties to identify all informative variables. Hence, it is also more challenging to deselect the non-informative variables without a further reduction in the true positives. Therefore, smaller  $\tau$  values are more appropriate, causing less deselection and more noise variables, but the signal variables still remain in the model. For the other SNRs, only variable  $X_7$  contributes to the risk reduction for small threshold values (0.5%, 0.75%, and 1%). Here, the relative risk reduction showed that a small  $\tau$  value is sufficient to remove almost all false positives (red horizontal line for 1%). Furthermore, a higher  $\tau$  value can lead not only to a reduction in informative variables included in the model but also to a worse MSE. Due to the noise, the predictive performance for the SNR of 0.15 was very poor and showed no discernible differences between the threshold values. However,  $\tau$  values above 2.5% lead to a decrease in performance for larger SNRs. Furthermore, the previous simulation results have shown that a low value for  $\tau$  (in this case 1%) reduces the number of false positives and additionally leads to a comparable predictive performance to classical boosting.

Overall, the number of false positives in the resulting models could be significantly reduced by earlier stopping or deselection as well as twin boosting compared to classical boosting. However, in most cases, the reduction of false positives for oSE and RobustC resulted in worse prediction performance. A comparison with probing for Scenario A, Scenario B, and Scenario C showed similar behavior (given in Supplemental Material A.2). Probing also led to a reduction in the number of false positives, but resulted in worse prediction performance, particularly for Scenario B. Furthermore, the earlier stopping strategies removed a few informative variables from the model in some settings.

The new procedure also deselected some informative variables from time to time, but removed the non-informative variables almost completely and resulted in favorable prediction performance. In some settings, the new approach even resulted in better predictive performance than the classical boosting model. Additional simulation results for the high-dimensional setting with a block structure for the covariance matrix are provided in Supplemental Material A.1 and showed very similar results compared with the Toeplitz covariance structure.

From the consideration of different SNRs, we conclude that the relative risk reduction attributed to a base-learner does not depend much on the overall SNR but on the distribution of the signal among the base-learners. To ensure that not too many informative variables are de-selected and to achieve a favorable predictive performance, our results suggest that the threshold value should be chosen rather small (e.g. 1%). For larger SNR values (6 and 14.64), almost all noise variables were eliminated even for small  $\tau$ . Higher threshold values resulted in worse performance and a significant reduction of the informative variables. The choice of 1% for the threshold  $\tau$  resulted in a reasonable tradeoff between sparsity and prediction performance in all considered settings. The *best* threshold, however, will always depend on the actual goal of the analysis and the general data situation.

Additionally, our approach also performed well compared to twin boosting, particularly for the linear regression model. An advantage of our method is the possibility to enhance variable selection for non-linear and distributional regression, which to the best of our knowledge is currently not available for twin boosting.

**Table 1.** Results for GCKD data in terms of the mean (sd) number of selected variables for the parameters  $\mu$  and  $\phi$  as well as the negative log-likelihood representing the prediction performance on the 1000 bootstrap replicates.

Model	$\mu$	$\phi$	−log-likelihood
classical boosted model	26.43 (7.10)	14.55 (6.00)	−1457.08 (40.24)
deselected ( $\tau = 0.01$ )	12.58 (1.39)	7.87 (1.61)	−1441.73 (39.41)
oSE	8.06 (2.68)	2.92 (1.63)	−1295.34 (37.88)
RobustC	7.63 (2.31)	2.70 (1.34)	−1290.14 (32.88)

### Quality of life of chronic kidney disease patients

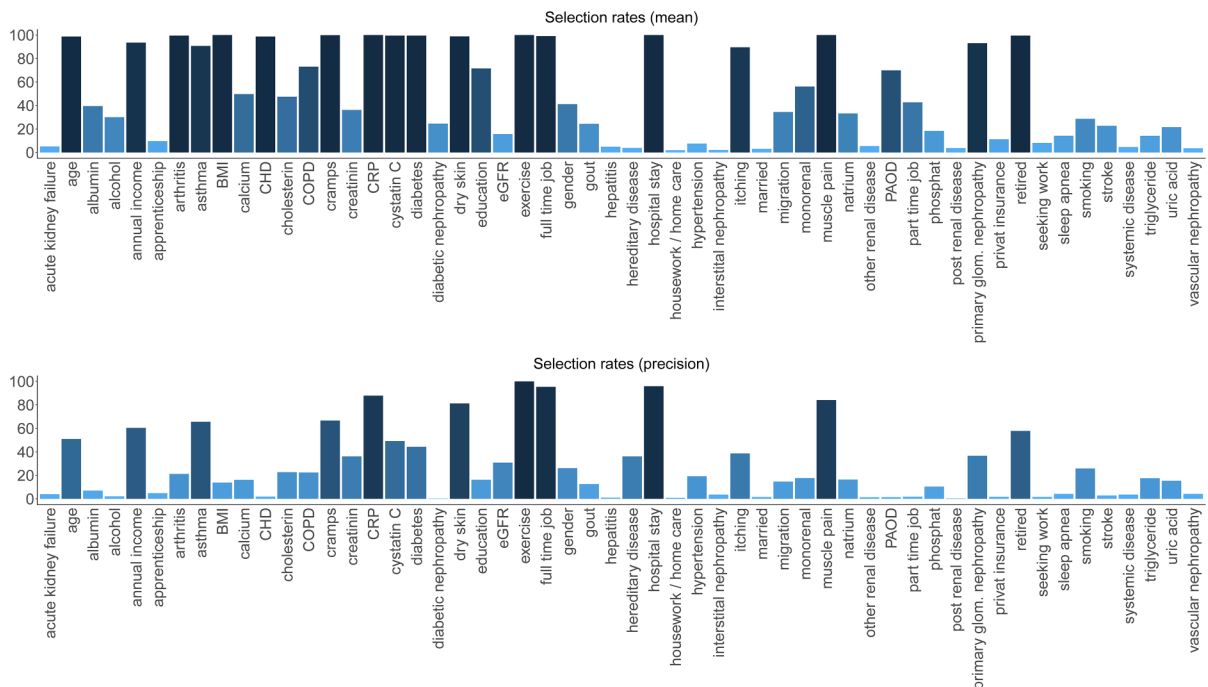
The following analysis aims to identify the most important predictors for the QoL of stage III chronic kidney disease patients based on an ongoing German cohort study (German Chronic Kidney Disease Study, GCKD). A similar analysis has already been published (cf. Mayr et al.<sup>9</sup>) and led to the selection of rather large models which partly motivated the current new methodological developments.

The analysis is based on beta regression,<sup>37</sup> which is a very flexible approach to model bounded outcome variables like proportions. It is also a well-known tool in the analysis of health-related QoL scores,<sup>38; 39</sup> which typically range from 0 (lowest possible value) to 100 (highest possible value). The density function of a beta distribution with expected value  $\mu$  and precision parameter  $\phi$  is given by

$$f(y; \mu, \phi) = \frac{\Gamma(\phi)}{\Gamma(\mu\phi)\Gamma((1-\mu)\phi)} y^{\mu\phi-1} (1-y)^{(1-\mu)\phi-1}, \quad 0 < y < 1,$$

where  $\Gamma(\cdot)$  denotes the gamma function. In context of distributional beta regression, which refers to a generalized additive model for location, scale and shape (GAMLSS), we model  $\mu$  and additionally  $\phi$  in terms of several explanatory variables.

The GCKD study<sup>40</sup> is an ongoing cohort study for patients with stage III chronic kidney disease. We analyzed part of the cross-sectional baseline-data with  $n = 3522$  observations and 54 explanatory variables. We aimed to select the most informative variables for the QoL of chronic kidney disease patients<sup>9</sup> using the **R** add-on package **betaboost**. The effects of the predictors on the quality of life are represented by base-learners. For continuous covariates we incorporated spline effects as base-learners. For factor variables (e.g. education and exercise) we used linear base-learners providing joint updates of the



**Figure 8.** The selection rates of the explanatory variables for  $\mu$  and  $\phi$  of the classical boosting algorithm in 1000 bootstrap samples.

effects for the different categories in the boosting iterations. Therefore, our approach yields potential deselection (sparsity) on the full factor level and not on the level of different categories of a factor. Alternatively, multi-categorical factors may also be re-coded as several binary dummy variables, so that categories could be selected (and deselected) independently.

We drew 1000 bootstrap replicates and fitted a beta regression model without and with the new deselection procedure using  $\tau = 1\%$  for each bootstrap sample (results for different  $\tau$  values are given in Supplemental Material A.4). To evaluate the predictive performance of the resulting models, the negative log-likelihood was computed on the *out-of-bag* bootstrap samples. The optimal number of boosting iterations were selected via 10-fold cross-validation. For comparison, we additionally considered the oSE and RobustC methods.

### Results

Table 1 displays the mean number (with standard deviations) of selected variables for  $\mu$  and  $\phi$  as well as the average negative log-likelihood for the different models on the 1000 bootstrap replicates. One can observe that more variables are included for the expected value than for the precision parameter. The earlier stopping strategies contain fewer variables than the proposed deselection approach for boosting.

In addition to Table 1, we consider the selection rates for each variable (for  $\mu$  and  $\phi$ ) on the 1000 bootstrap replicates. Figure 8 displays the selection rates of the classical boosting approach. As described in Mayr et al.,<sup>9</sup> the highest selection rates for parameter  $\mu$  were obtained for age, body mass index (BMI), exercise, and variables related to pain such as arthritis, cramps, and muscle pain. Furthermore, variables that are indicators of kidney failure and inflammation also had higher rates, for example, cystatin C. For the precision parameter  $\phi$ , 15 variables were included on average, with the highest rates for the variables exercise, employment in a full-time job and hospital stay.

The selection rates after additionally applying the deselection approach in Figure 9 show that the new procedure achieved a significant reduction in the number of included variables; some variables that were rarely selected by classical boosting were never included with the new approach (e.g. alcohol, gender). On the other hand, the variables with the highest selection rates from the classical model were still present at the highest selection rates.

To evaluate the predictive performance of the resulting models, we considered the negative log-likelihood on test data as a scoring rule. The results in Figure 10 suggest that the new deselection procedure outperforms the earlier stopping strategies oSE and RobustC. The smallest negative log-likelihood was obtained with the classical boosting model with an average value of 1457.08 (see Table 1), whereby we achieved a comparable performance for deselection with  $\tau = 0.01$ .

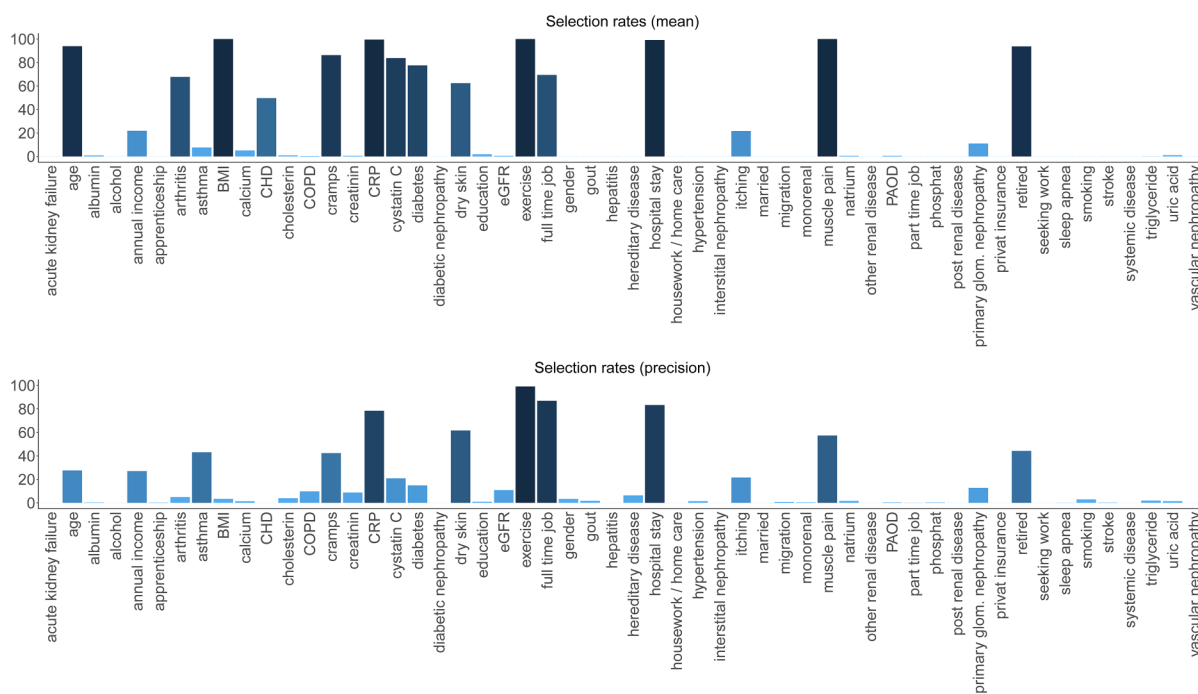
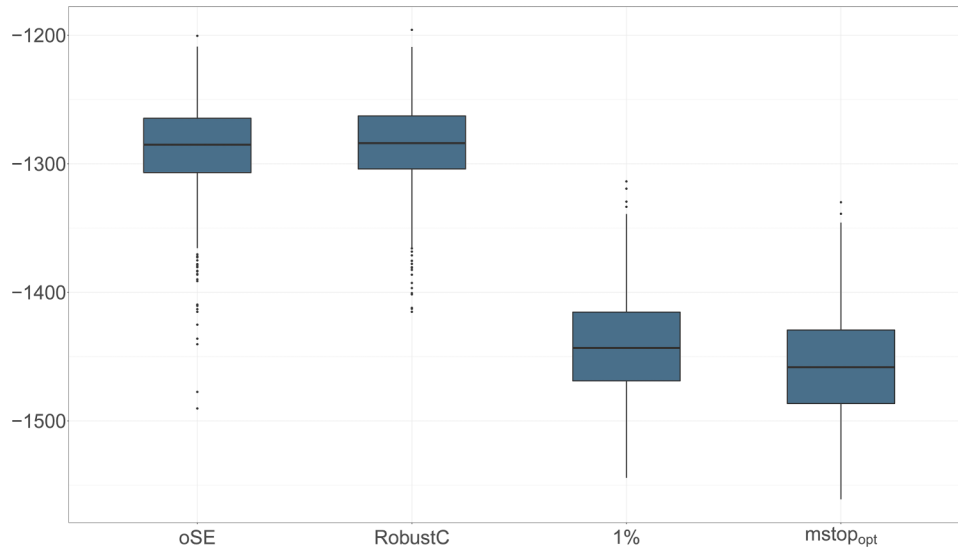


Figure 9. The selection rates of the explanatory variables for  $\mu$  and  $\phi$  after applying the new deselection approach with  $\tau = 0.01$  in 1000 bootstrap samples.



**Figure 10.** Negative log-likelihood of the oSE, RobustC, the deselection procedure with a threshold value of 1% and the classical boosted model on *out-of-bag* bootstrap samples.

Overall, the deselection approach based on 1% of the total risk reduction was able to enhance the sparsity of boosting models by selecting less predictors for the health-related QoL in chronic kidney disease patients (on average 12.6 for the expected value and 7.9 for the precision parameter from 54 candidate variables) in comparison to the classical boosting approach. However, many predictors contribute to the overall risk reduction to a small extent (see Supplemental Material, Figure A7). That indicates that the "true" underlying model is not as sparse as in the simulations. Earlier stopping strategies can further increase the sparsity, however leading to much poorer predictive performance on *out-of-bag* data (Figure 10). In comparison, the deselection approach leads to a slightly worse predictive performance in comparison to classical boosting, but yields much smaller and more interpretable models.

In practice, we have to deal with the tradeoff between sparsity and predictive performance, which is regulated by the threshold parameter  $\tau$ . Therefore, higher variable deselection (i.e. larger values of  $\tau$ ) leads to smaller models but, at least in this considered application, also to poorer predictive performance.

## Discussion and conclusion

The presented approach to deselect base-learners for enhanced variable selection in statistical boosting is a new technique to obtain sparser models with simpler interpretation via the removal of irrelevant predictors with negligible impact. As the deselection is based on the risk reduction, this approach is suitable for any type of base-learners, for example, linear models, splines, and spatial effects. Furthermore, the deselection approach can also be combined with a wide range of regression settings, including multi-dimensional optimization problems like distributional regression.

Compared to the similar twin-boosting,<sup>10</sup> our approach actively deselects base-learners via a threshold value. This is somehow an analogy to stability selection<sup>41; 42</sup> but our method focuses on providing a sparse prediction model instead of a set of stable predictors.<sup>43; 44</sup> Furthermore, it does not include additional resampling steps. Other approaches for enhanced variable selection in the context of boosting focused on strategies for earlier stopping<sup>27; 13</sup> which typically also increases the amount of shrinkage on effect estimates which might not be necessarily favorable. Our approach hence focuses on a vertical view on the regularization paths (in contrast to the horizontal view with earlier stopping) which has already been discussed in the literature on the lasso.<sup>15; 16</sup>

The new approach is particularly suitable for high-dimensional data (with more potential predictors than observations) as one can obtain a simplified model with the most relevant variables yielding in many cases almost the same prediction accuracy as the classical boosting approach without deselection. Consequently, the interpretability of resulting prediction models improves, which makes their application in practice more likely.<sup>45</sup>

The results of the simulation study suggest that our procedure can yield much sparser models by deselecting wrongly selected variables; in many cases deselection was associated with an almost complete elimination of false positives. In practice, one could assume that this might often lead also to a decreased prediction accuracy: the standard boosting approach

already selects the optimal prediction model by optimizing the stopping iteration. However, at least in some of the simulation settings, the deselection of false positives led even to a slightly improved prediction performance.

To select the most informative predictors for the health-related QoL of chronic kidney disease patients (GCKD study), the deselection procedure resulted in a drastically reduced set of variables compared to a recent analysis,<sup>9</sup> which partly motivated the new methodological development. However, we did also observe a slight worsening of the model performance (w.r.t. the likelihood on test data).

The deselection procedure is controlled via a threshold value  $\tau$ : it represents the minimum amount of total risk reduction which should be attributed to a corresponding base-learner in order to avoid deselection. In the simulation study, a threshold of  $\tau = 0.01$  (i.e. 1% of total risk reduction) was considered to be appropriate overall. However, the general tradeoff between a more complex model with the highest possible prediction accuracy and a sparser, more interpretable model (higher *descriptive accuracy*<sup>46</sup> with potentially reduced prediction accuracy) should be guided by the researcher, depending on the research question and the context of the problem. As an alternative, the threshold parameter could also be chosen via resampling techniques or cross-validation which might further increase the performance but leads also to higher computational burden, particularly for high-dimensional data.

A limitation of our procedure is the assumption of sparsity. In cases where this is not fulfilled, it might deselect too many variables: If, for example, many predictors affect the model with minor impact (e.g. 200 variables with equal importance), our approach with  $\tau = 0.01$  may deselect all variables. This is due to the dependency of our approach on the distribution of risk reduction across the base-learners. In theory, it would be beneficial to select  $\tau$  based on the minimal signal strength, for example, the minimal risk reduction attributed to an informative predictor. As the truly informative variables, however, are unknown—this choice remains challenging in practical applications. An alternative technique, particular for non-sparse settings, could be to consider the *cumulative* risk reduction. Instead of considering only the risk reduction attributed to the corresponding base-learner, the cumulative risk considers the risk reduction of all base-learners that are to be deselected from the model. Thus, this procedure accounts for the complete tail of the base-learners with low importance. This variant would typically yield larger models when used with the same threshold. We investigated also the deselection via the cumulative risk reduction (results for the simulation and the application are given in Supplemental Material B). This alternative version is also implemented and available together with the corresponding code to reproduce the simulations and can be applied by specifying `method = "cumulative"`.

The favorable performance of our new approach motivates research in this direction in the future, in particular for distributional regression,<sup>47</sup> where sparse and interpretable models are of particular importance. For instance, deselection could be also extended to the level of distribution parameters in order to deselect a complete model-dimension (e.g. decreasing a GAMLSS to a GAM) when the contribution to the overall risk reduction is limited. Another line of potential research could focus on the combination of earlier stopping with deselection to avoid the disadvantage of exaggerated shrinkage.<sup>14</sup>

Altogether, we conclude that in our simulation and application the new deselection approach was able to outperform existing methods for earlier stopping, concerning the number of selected variables and the predictive performance. However, it should be noted that these approaches pursue various different goals like variable selection, prediction performance and/or interpretability. Fitting one model, that is able to achieve the best solution for all potential goals, simply might often not be feasible (cf. Hothorn<sup>48</sup>).

## Acknowledgements

The authors thank Benjamin Hofner for fruitful discussions on the underlying methodology of the new deselection procedure.


## Declaration of conflicting interest


The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The work on this article was supported by the Deutsche Forschungsgemeinschaft (DFG, grant nos. 428239776, KL3037/2-1, MA7304/1-1). The GCKD study was funded by grants from the German Ministry of Education and Research (BMBF) (<http://www.gesundheitsforschung-bmbf.de/de/2101.php>; grant no. 01ER0804) and the KfH Foundation for Preventive Medicine (<http://www.kfh-stiftung-praeventivmedizin.de/content/stiftung>).

## ORCID iDs

Annika Strömer  <https://orcid.org/0000-0002-1284-3318>

Andreas Mayr  <https://orcid.org/0000-0001-7106-9732>



## Supplemental Material

Supplemental material for this article is available online. It contains an alternative deselection approach via the cumulative risk reduction and corresponding results of the simulation study and the GCKD data.

## References

1. Fan J and Lv J. A selective overview of variable selection in high dimensional feature space. *Stat Sin* 2010; **20**: 101–148.
2. Chen TH, Chatterjee N, Landi MT, et al. A penalized regression framework for building polygenic risk models based on summary statistics from genome-wide association studies and incorporating external information. *J Am Stat Assoc* 2020; **0**: 1–11.
3. Choi SW, Mak TSH and O'Reilly PF. Tutorial: a guide to performing polygenic risk score analyses. *Nat Protoc* 2020; **15**: 2759–2772.
4. Steyerberg EW and Vergouwe Y. Towards better clinical prediction models: seven steps for development and an ABCD for validation. *Eur Heart J* 2014; **35**: 1925–1931.
5. Sauerbrei W, Perperoglou A, Schmid M, et al. State of the art in selection of variables and functional forms in multivariable analysis—outstanding issues. *Diagnostic Prognostic Res* 2020; **4**: 1–18.
6. Bühlmann P and Hothorn T. Boosting algorithms: Regularization, prediction and model fitting. *Stat Sci* 2007; **22**: 477–505.
7. Mayr A, Hofner B, Waldmann E, et al. An update on statistical boosting in biomedicine. *Comput Math Methods Med* 2017; **2017**: 1–12.
8. Bühlmann P, Gertheiss J, Hieke S, et al. Discussion of the evolution of boosting algorithms and extending statistical boosting. *Methods Inf Med* 2014; **53**: 436–445.
9. Mayr A, Weinhold L, Hofner B, et al. The betaboost package – a software tool for modelling bounded outcome variables in potentially high-dimensional epidemiological data. *Int J Epidemiol* 2018; **47**: 1383–1388.
10. Bühlmann P and Hothorn T. Twin boosting: improved feature selection and prediction. *Stat Comput* 2010; **20**: 119–138.
11. Zou H. The adaptive LASSO and its Oracle properties. *J Am Stat Assoc* 2006; **101**: 1418–1429.
12. Breiman L, Friedman J, Stone CJ, et al. *Classification and Regression Trees*. Boca Raton: CRC Press, 1984.
13. Ellenbach N, Boulesteix AL, Bischl B, et al. Improved outcome prediction across data sources through robust parameter tuning. *J Classif*, 2021; **38**: 212–231.
14. Van Calster B, van Smeden M, De Cock B, et al. Regression shrinkage methods for clinical prediction models do not guarantee improved performance: Simulation study. *Stat Methods Med Res* 2020; **29**: 3166–3178.
15. Zhou S. Thresholding procedures for high dimensional variable selection and statistical estimation. In *Proceedings of the 22nd International Conference on Neural Information Processing Systems (NIPS'09)*. Curran Associates Inc., Red Hook, NY, USA, 2009. 2304–2312.
16. Weinstein A, Su WJ, Bogdan M, et al. A Power Analysis for Knockoffs with the Lasso Coefficient-Difference Statistic; 2020. arXiv preprint arXiv:2007.15346.
17. Freund, Y. Boosting a weak learning algorithm by majority. *Information and Computation* 1995; **12**: 256–285.
18. Freund Y and Schapire RE. Experiments with a New Boosting Algorithm. In: *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*. Bari, Italie: Morgan Kaufmann Publishers Inc, 1996. pp.148–156.
19. Friedman J, Hastie T and Tibshirani R. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *Ann Stat* 2000; **28**: 337–407.
20. Friedman J. Greedy function approximation: A gradient boosting machine. *Ann Stat* 2001; **29**: 1189–1232.
21. Mayr A, Binder H, Gefeller O, et al. The evolution of boosting algorithms. *Methods Inf Med* 2014; **53**: 419–427.
22. Mayr A, Binder H, Gefeller O, et al. Extending statistical boosting. *Methods Inf Med* 2014; **53**: 428–435.
23. Hofner B, Mayr A, Robinzonov N, et al. Model-based boosting in R: A hands-on tutorial using the R package mboost. *Comput Stat* 2014; **29**: 3–35.
24. Mayr A, Hofner B and Schmid M. The importance of knowing when to stop: A sequential stopping rule for component-wise gradient boosting. *Methods Inf Med* 2012; **51**: 178–186.
25. Friedman J, Hastie T and Tibshirani R. Regularization paths for generalized linear models via coordinate descent. *J Stat Softw* 2010; **33**: 1–22.
26. Hastie T, Tibshirani R and Friedman JH. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer, 2009.
27. Thomas J, Hepp T, Mayr A and Bischl B. Probing for Sparse and Fast Variable Selection with model-based boosting. *Computational and Mathematical Methods in Medicine* 2017; **2017**: 1421409.
28. Hofner B, Boccuto L and Göker M. Controlling false discoveries in high-dimensional situations: Boosting with stability selection. *BMC Bioinformatics* 2015; **16**: 144.
29. Wood SN. *Generalized Additive Models: An Introduction with R*. 2nd ed. Boca Raton: CRC Press, 2017.
30. Rigby RA and Stasinopoulos DM. Generalized additive models for location, scale and shape. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 2005; **54**: 507–554.
31. Mayr A, Fenske N, Hofner B, et al. Generalized additive models for location, scale and shape for high dimensional data—a flexible approach based on boosting. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 2012; **61**: 403–427.

32. Thomas J, Mayr A, Bischl B, et al. Gradient boosting for distributional regression: faster tuning and improved variable selection via noncyclical updates. *Stat Comput* 2017; **28**: 1–15.
33. Schmid M and Hothorn T. Boosting additive models using component-wise P-splines. *Comput Stat Data Anal* 2008; **53**: 298–311.
34. R Core Team.: R: A Language and Environment for Statistical Computing. Vienna, Austria; 2019.
35. Hothorn T, Bühlmann P, Kneib T, et al. mboost: Model-Based Boosting; 2020. R package version 2.9-2.
36. Wang Z. bst: Gradient Boosting; 2020. R package version 0.3-23.
37. Ferrari S and Cribari-Neto F. Beta regression or modeling rates and proportions. *J Appl Stat* 2004; **31**: 799–815.
38. Hunger M, Baumert J and Holle R. Analysis of SF-6D index data: is beta regression appropriate? *Value Health* 2011; **14**: 759–767.
39. Hunger M, Döring A and Holle R. Longitudinal beta regression models for analyzing health-related quality of life scores over time. *BMC Med Res Methodol* 2012; **12**: 144.
40. Eckardt KU, Bärthlein B, Baid-Agrawal S, et al. The German chronic kidney disease (GCKD) study: design and methods. *Nephrol Dial Transplant* 2012; **27**: 1454–1460.
41. Meinshausen N and Bühlmann P. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 2010; **72**: 417–473.
42. Shah RD and Samworth RJ. Variable selection with error control: another look at stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 2013; **75**: 55–80.
43. Hofner B, Boccutto L and Göker M. Controlling false discoveries in high-dimensional situations: boosting with stability selection. *BMC Bioinformatics* 2015; **16**: 144.
44. Mayr A, Hofner B and Schmid M. Boosting the discriminatory power of sparse survival models via optimization of the concordance index and stability selection. *BMC Bioinformatics* 2016; **17**: 288.
45. Wyatt JC and Altman DG. Prognostic models: clinically useful or quickly forgotten? *Br Med J* 1995; **311**: 1539–1541.
46. Murdoch WJ, Singh C, Kumbier K, Abbasi-Asl R, and Yu B. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences* 2019; **116**: 22071–22080.
47. Stasinopoulos MD, Rigby RA, Heller GZ, et al. *Definitions, methods, and applications in interpretable machine learning*. CRC Press, 2017.
48. Hothorn T. Invited discussion on “Meinshausen and Bühlmann: Stability selection”. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 2010; **72**: 463–464.