Statistics in Biosciences
https://doi.org/10.1007/s12561-025-09491-8

ORIGINAL PAPER

Enhanced Variable Selection for Boosting Sparser and Less Complex Models in Distributional Copula Regression

Annika Strömer[1,2] · Nadja Klein[3] · Christian Staerk[4,5] · Florian Faschingbauer[6] · Hannah Klinkhammer[2,7] · Andreas Mayr[1]

Received: 17 April 2024 / Revised: 17 March 2025 / Accepted: 18 May 2025
© The Author(s) 2025

Abstract
Structured additive distributional copula regression allows to model the joint distribution of multivariate outcomes by relating all distribution parameters to covariates. Estimation via statistical boosting enables accounting for high-dimensional data and incorporating data-driven variable selection, both of which are useful given the complexity of the model class. However, as known from univariate (distributional) regression, the standard boosting algorithm tends to select too many variables with minor importance, particularly in settings with large sample sizes, leading to complex models with difficult interpretation. To counteract this behavior and to avoid selecting base-learners with only a negligible impact, we combine the ideas of probing, stability selection, and a new deselection approach with statistical boosting for distributional copula regression. In simulations and an application to the joint modeling of weight and length of newborns, we find that all proposed methods enhance variable selection by reducing the number of false positives. However, only stability selection and the deselection approach yield similar predictive performance to classical boosting. Finally, the deselection approach is better scalable to larger datasets and leads to competitive predictive performance, which we further illustrate in a genomic cohort study from the UK Biobank by modeling the joint genetic predisposition for two phenotypes.

Keywords Distributional regression · Multiple outcomes · Probing · Stability selection · UK Biobank · Variable selection

1 Introduction

Statistical boosting, an iterative, sequential fitting algorithm for statistical models originating from machine learning [6], has gained increasing interest as an alternative to classical (penalized) maximum likelihood estimation (PMLE) or

Extended author information available on the last page of the article

Published online: 17 June 2025

Bayesian inference. Since boosting is well suited for high-dimensional and complex data problems, it is also a useful tool for distributional regression, where the number of candidate models is typically large. Distributional regression generally has the aim of estimating complete conditional distributions of a quantity of interest as a function of covariates (see e.g., [14], for a recent review). A convenient framework for univariate distributional regression is the class of generalized additive models for location, scale, and shape (GAMLSS [27]), which allow relating each distribution parameter of a parametric response distribution to covariates. However, in the classical PMLE-based implementation, the response is restricted to be univariate and the complexity of the predictors is limited due to numerical instabilities when it comes to selecting smoothing parameters for regularization. While Bayesian estimation of GAMLSS based on Markov chain Monte Carlo simulations [15, 39] allows to overcome the latter issues, it is notoriously slow when the number of observations n and/or the number of covariates p is large. In such scenarios, statistical boosting is particularly beneficial as also demonstrated in various applications and extensions of the original boosting algorithm (see e.g., [16, 31]).

In this paper, we are not only concerned with situations, where either n or p is large or where even $p \gg n$, but particularly when the outcome Y is multivariate. By modeling dependent outcomes together, we can gain a better understanding of the relationship and identify relevant factors affecting their association. An example is the consideration of multiple phenotypes from deeply phenotyped cohort studies in genetic epidemiology. For distributional regression modeling of multiple outcomes, one approach is to consider the joint parametric distribution. A popular alternative in this context are copulas that offer increased flexibility by allowing the use of different marginals and different dependence structures through the copula function. This approach has been the subject of ongoing research and there is rich literature on copula modeling with regression data (see e.g., [24, 40, 43], for recent examples).

Statistical boosting was extended to multivariate distributional regression towards parametric distributions [34] and also using copula [9]. However, while this is done conceptually in these works, some practical aspects are still challenging. One challenge is that while boosting has an implicit variable selection mechanism, it often leads to relatively *large* models, that is, models with many included covariates despite having small to negligible effects. This happens because the algorithm typically optimizes prediction accuracy without explicitly considering sparsity. This was also observed for boosting copula regression, where especially the sub-models for the location parameter did result in rather large numbers of selected covariates [9]. This behavior is particularly undesirable in situations with a large number of candidate variables p, where sparse and interpretable models are practically relevant. Therefore, variable selection is of great importance, not only in the context of boosting (see e.g., [13, 38]). In distributional copula regression, a further interesting question that arises is how to decide in a data-driven manner if the overall complexity of the model could be reduced. In this context, the aim of statistical modeling is to select a model that is as complex as needed but also as simple as possible to facilitate efficient estimation and interpretability. For example, if certain distribution parameters like the association parameters do

not depend on covariates, then simpler univariate models might also be suitable. Statistical boosting can help to answer this question.

To tackle these yet unaddressed practical challenges in boosting distributional copula regression, we incorporate three existing approaches for refining variable selection within this framework. All three approaches have been already proposed or extended to boosting, but have never been integrated into boosting multivariate distributional regression via copulas. This new combination aims to reduce the complexity of the model, particularly when dealing with high-dimensional data. The three considered existing approaches for enhanced variable selection are the following: (i) Stability selection [22], which has been extended to boosting univariate distributional regression [37]; (ii) probing [36], which was proposed for boosting simple mean regression models, shifts the focus of early stopping from prediction accuracy directly to variable selection; and (iii) deselection [33], the newest approach, which pragmatically deselects base-learners that do not contribute enough to the overall model performance.

We initially investigate the performance of these three methods on simulated data, where the true data-generating process is known. Afterward, we consider two real data applications: first, the joint modeling of the weight and length of newborns (as in [9]), and second, the modeling of the joint genetic disposition towards continuous phenotypes based on a large cohort study (UK Biobank).

2 Methods

2.1 Boosting Distributional Copula Regression

In this section, we briefly review distributional copula regression models focusing on the bivariate case of two continuous outcomes and how statistical boosting algorithms can be applied for model estimation.

2.1.1 Distributional Copula Regression Models

A flexible modeling approach for the joint analysis of two continuous response variables $\boldsymbol{Y} = (Y_1, Y_2)^\top$ in terms of covariates are bivariate copula regression models, which describe the dependence structure through a copula [23]. According to Sklar's theorem, the joint conditional cumulative distribution function (CDF) of two responses given covariate information $\boldsymbol{x}$ can be written as

$$F(y_1, y_2 \mid \boldsymbol{\theta}) = C\big[F_1(y_1 \mid \boldsymbol{\theta}^{(1)}), F_2(y_2 \mid \boldsymbol{\theta}^{(2)}) \mid \boldsymbol{\theta}^{(c)}\big],$$

where $F_1(\cdot \mid \boldsymbol{\theta}^{(1)})$ and $F_2(\cdot \mid \boldsymbol{\theta}^{(2)})$ are the marginal conditional CDFs of the two responses $Y_1 = y_1$ and $Y_2 = y_2$ which are uniformly distributed on [0, 1]. The copula function $C(\cdot, \cdot \mid \boldsymbol{\theta}^{(c)})$ contains the information about the dependence structure between the two outcomes and is unique when the responses are continuous. The vector $\boldsymbol{\theta} = \{(\boldsymbol{\theta}^{(1)})^T, (\boldsymbol{\theta}^{(2)})^T, (\boldsymbol{\theta}^{(c)})^T\}^\top$ contains the model parameters $k = 1, \ldots, K$ of the marginal distributions and the copula, whereby all components of $\boldsymbol{\theta} \equiv \boldsymbol{\theta}(\boldsymbol{x})$ can be linked to a covariate vector via additive predictors and appropriate link functions.

The representation of the joint conditional CDF via a copula allows the separation of the marginal distributions and the dependence structure; different copula functions allow different structures to be modeled. The Clayton copula, for example, can capture asymmetric dependence (so-called lower tail dependence), where the two responses show a stronger positive association for smaller values than for larger values. In our work we will focus on Gaussian, Clayton, and Gumbel copulas (cf. [23]) to represent no, lower, and upper tail dependencies.

The joint density $f(y_1, y_2 \mid \boldsymbol{\theta})$ of a distributional copula regression model can be expressed via

$$\begin{aligned} f(y_1, y_2 \mid \boldsymbol{\theta}) &= \frac{\partial^2}{\partial F_1 \partial F_2} F(y_1, y_2 \mid \boldsymbol{\theta}) \\ &= c\left[F_1(y_1 \mid \boldsymbol{\theta}^{(1)}), F_2(y_2 \mid \boldsymbol{\theta}^{(2)}) \mid \boldsymbol{\theta}^{(c)}\right] f_1(y_1 \mid \boldsymbol{\theta}^{(1)}) f_2(y_2 \mid \boldsymbol{\theta}^{(2)}), \end{aligned}$$

where $f_1(\cdot \mid \boldsymbol{\theta}^{(1)})$ and $f_2(\cdot \mid \boldsymbol{\theta}^{(2)})$ are the marginal probability density functions and $c(\cdot, \cdot \mid \boldsymbol{\theta}^{(c)})$ is the copula density of C. Based on our applications, the most relevant marginal distributions in this work are the log-logistic and the log-normal distributions.

Finally, for a dataset of n independent pairs $\{(\boldsymbol{y}_i, \boldsymbol{x}_i)\}_{i=1}^n$ of bivariate responses $\boldsymbol{y}_i = (y_{i1}, y_{i2})^\top$ with covariate information $\boldsymbol{x}_i$, the joint log-likelihood function is given by

$$l(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log\left\{c\left[F_1(y_{i1} \mid \boldsymbol{\theta}^{(1)}), F_2(y_{i2} \mid \boldsymbol{\theta}^{(2)}) \mid \boldsymbol{\theta}^{(c)}\right]\right\} + \sum_{i=1}^{n} \sum_{d \in \{1,2\}} \log\left\{f_d(y_{id} \mid \boldsymbol{\theta}^{(d)})\right\}.$$

2.1.2 Structured Additive Predictors

In distributional copula regression, each distribution parameter θ_k, $k = 1, \ldots, K$ is modeled via a structured additive predictor η_k [3, 32] with parameter-specific monotonic link functions g_k, such that $g_k(\theta_k) = \eta_k$ and $g_k^{-1}(\eta_k) = \theta_k$, where g_k^{-1} is the inverse of g_k. The additive predictors η_k depend on (possibly different) subsets of $\boldsymbol{x}$,

$$g_k(\theta_k) = \eta_k = \beta_{0k} + \sum_{j=1}^{p_k} f_{jk}(\boldsymbol{x}_{jk}), \text{ for } k = 1, \ldots, K,$$

where β_{0k} are the intercepts and each f_{jk}, $j = 1, \ldots, p_k$, represents functional effects of covariates $\boldsymbol{x}_{jk}$, whereby $\boldsymbol{x}_{jk}$ is a covariate subset of $\boldsymbol{x}$. The effects can be chosen in a flexible manner [4], for instance we incorporate linear and non-linear effects in Sections 3 and 4. Linear effects can be represented by $f_{jk}(\boldsymbol{x}) = \boldsymbol{x}_{jk}^T \boldsymbol{\beta}_{jk}$ where $\boldsymbol{\beta}_{jk}$ are the regression coefficients. Non-linear effects can be included using appropriate basis functions, such as B-splines.

2.1.3 Estimation Via Model-Based Boosting

Component-wise gradient boosting with regression-type base-learners, also referred to as *statistical boosting* [18], originates from the gradient boosting approach of [6], who translated the original concept from the machine learning literature to statistical modeling. Its basic idea is to iteratively minimize a pre-specified loss function $\mathcal{L}$ by fitting the so-called base-learners separately to the negative gradient $\mathcal{L}$ and by then adding only a small amount of the "best-fitting" base-learner—that is, the base-learner that yields the steepest descent in the direction of the current gradient—to the overall regression predictor in each step of the boosting algorithm. In our case, a base-learner represents one effect in the additive regression predictor (see [12] for a detailed overview). In this way, the overall predictor is built sequentially, where more and more variables are selected the longer the algorithm runs, such that *early stopping* yields implicit variable selection. In likelihood-based statistical boosting, the loss $\mathcal{L}$ is the negative log-likelihood $l \equiv l(\boldsymbol{\theta})$, but more general functionals such as proper scoring rules are possible.

The boosting algorithm is a flexible alternative to classical estimation approaches. It has several practical advantages, such as dealing with high-dimensional data in which classical inferential methods are no longer applicable. As mentioned above, the algorithm performs data-driven variable selection, which is controlled by the number of boosting iterations m_{stop} [19]: Variables whose corresponding base-learner has never been selected until m_{stop} is reached are excluded from the final model. Therefore, the number of boosting iterations is the main tuning parameter and is typically optimized by cross-validation or resampling techniques. Another parameter of the algorithm is the fixed step length ν, with which the best-fitting base-learner is multiplied before being included into the predictor. This parameter is set to a small fixed value within the range of $0 < \nu < 1$ [28]. For boosting copula regression, [9] suggests a value of $\nu = 0.01$.

In the boosting approach for distributional copula regression [9], all distribution parameters are modeled simultaneously by combining the properties of GAMLSS and the main features of statistical boosting. In every iteration, the partial derivatives $u_k = \partial l / \partial \theta_k$ of the negative log-likelihood l with respect to the different distribution parameters θ_k are calculated and each base-learner $h_{jk} \equiv f_{jk}(\boldsymbol{x}_{jk})$ is separately fitted to the gradient. Then, the best-fitting base-learner (and the corresponding update) for each distribution parameter is determined and compared across the different dimensions. Only the overall best-performing update is finally performed using a non-cyclic version of the algorithm [37]. For more details on fitting distributional copula regression via boosting, we refer to [9].

2.2 Complexity Reduction and Enhanced Variable selection

In the following, we present different techniques for enhanced variable selection that we will integrate in our boosting distributional copula framework. Probing (Sect. 2.2.1) had been introduced to statistical boosting by [36] and since then been applied or used as a

benchmark approach for mean regression models with only one dimension [2, 31] or joint models of time-to-event and longitudinal data [8]. Stability selection (Sect. 2.2.2) is a more general approach [22] and has been introduced to boosting mean regression models by [11], before the approach was extended to the context of univariate distributional regression [37]. Deselection (Section 2.2.3) is the most recent enhancement and was directly introduced for boosting mean regression as well as distributional regression [33]. None of the three approaches have ever been extended towards boosting multivariate distributional regression or even to copula regression. In the process of integrating these enhanced variable selection approaches in our framework, we also allow for constant distribution parameters that do not depend on covariates. This is particularly attractive in copula regression, where for example copula parameters not depending on covariates reflect situations in which the dependence structure between the outcomes does not vary across observations with distinct feature values. This can lead to a substantial reduction in the complexity of the final model.

2.2.1 Probing

Probing is based on the inclusion of random noise variables, the so-called probes, to determine the stopping iteration by stopping when the algorithm starts selecting those (Algorithm 1): First, randomly generated shuffled versions (probes) of the covariates are added to the original dataset. Second, a boosting model is fitted on the expanded dataset and the algorithm stops when the first probe is selected. The idea is that, in each iteration, the base-learner with the highest loss reduction is updated and the selection of a probe means that the best possible improvement is based on information known to be unrelated to the outcome. Because each parameter may depend on a potentially different set of variables, the randomly shuffled probes are simply added for each of the distribution parameters. In our model class, the distribution parameters may represent parameters of the marginal distributions or the copula. In each boosting iteration, a single base-learner is updated, i.e., the algorithm stops when the first probe is selected for any of the distribution parameters. While probing does not require optimizing the stopping criterion via computationally expensive cross-validation or resampling, it optimizes towards sparse models and does not maximize prediction performance. As a consequence, probing typically yields sparse models with strongly regularized predictor effects [36].

Algorithm 1 Probing for boosting distributional copula regression.

1: Shuffle probes $\tilde{\boldsymbol{x}}_{jk}$ for each of the covariates $\boldsymbol{x}_{jk}$ with $j = 1, \ldots, p_k$ and $k = 1, \ldots, K$.
2: Perform boosting on the expanded set of variables $x_1, \ldots, x_{p_k}, \tilde{x}_1, \ldots, \tilde{x}_{p_k}$ for each distribution parameter $\theta_k, k = 1, \ldots, K$.
3: Stop when the first probe $\tilde{x}_{jk}$ of any distribution parameter is selected.
4: Use final model from the previous iteration (containing only original variables).

2.2.2 Stability selection

A popular enhanced variable selection technique is *stability selection*, which yields a stable set of covariates by repeated model fitting using subsamples of the original dataset [22, 29]. In the context of boosting, Thomas et al. [37] introduced stability selection for boosted GAMLSS. As outlined in Algorithm 2, the general idea is to draw B random subsets of the data with size $\lfloor n/2 \rfloor$ of the original dataset and to fit separate boosting models for each subset. The boosting algorithm runs on each subset until a pre-specified number of covariates q have been selected. Every variable has a selection frequency defined by the fraction of subsets in which the variable j was selected. If the selection frequency exceeds the threshold π_{thr}, the variable is considered stable and is included in the final model fit [11]. Stability selection provides a sparse solution, controlling the number of false discoveries by defining an upper bound for the per-family error rate (PFER), i.e., the expected number $\mathbb{E}(V)$ of noninformative variables included in the final model. The upper bound is given by $\mathbb{E}(V) \leq q^2/((2\pi_{\text{thr}} - 1)p)$, where $p = \sum_{k=1}^{K} p_k$ is the total number of predictor variables and q the number of selected variables.

For practical use, the most important aspect is the choice of the parameters q, π_{thr} and PFER, whereby the PFER can be derived from the upper bound and visa versa. It is recommended to specify PFER and either q or π_{thr} [12]. Meinshausen and Bühlmann [22] state that the number of selected base-learners q should be chosen sufficiently large concerning the informative variables, or at least as high as the number of informative variables, which, however, are usually unknown. The threshold π_{thr} should be in the range of $\pi_{\text{thr}} \in (0.6, 0.9)$, meaning a variable should be selected in more than half of the fitted models in order to be considered stable. The choice of B is of minor importance as long as it is sufficiently large to ensure accurate estimation of $\hat{\pi}_j$ across various scenarios [22].

Algorithm 2 Stability selection for boosting distributional copula regression.

1: **for** $b = 1, \ldots, B$ **do**
2: Select a random subset from the data of size $\lfloor n/2 \rfloor$.
3: Fit a boosting model until q base-learner are selected.
4: **end for**
5: Compute the relative selection frequencies per base-learner $\hat{\pi}_j = \frac{1}{B} \sum I_{j \in \hat{S}_b}$, where $\hat{S}_b$ denotes the set of selected base-learner.
6: Select the stable set of base-learner $\hat{S}_{\text{stable}} := \{j : \hat{\pi}_j \geq \pi_{\text{thr}}\}$.
7: Fit a boosting model with the stable set of base-learners.

2.2.3 Deselection of Base-Learners

Another approach to encourage variable selection and sparsity is to deselect and remove base-learners with a negligible impact on the model's predictive

performance. The general idea is to start with a classical boosted model tuned by cross-validation or resampling techniques. Then, the base-learners that were selected but only have a minor impact on the model are identified and deselected. Afterward, the model is boosted again with the remaining variables. This idea was introduced by [33] for univariate GAMLSS and is now extended to distributional copula regression. The importance of a base-learner is based here on the risk reduction and can be defined for base-learner j after m_{stop} boosting iterations with

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

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 $m-1$, respectively. Note that in the case of distributional copula regression, all distribution parameters are considered together and each parameter $\theta_k, k = 1, \ldots, K$ can depend on a different number of variables p_k. Here, we do not distinguish between the different parameters, such that $p = \sum p_k$.

For a given threshold $\tau \in (0, 1)$, we deselect base-learner j if

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

where $r^{[0]} - r^{[m_{\text{stop}}]}$ represents the total risk reduction and R_j denotes risk reduction attributable to base-learner j. In other words, only base-learners whose contribution R_j to the total risk reduction is larger than the relative τ threshold (e.g., 1% [33]) will remain in the model after the deselection step.

Algorithm 3 Deselection for boosting distributional copula regression.

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

2.2.4 Illustration of the Different Approaches

Figure 1 displays the coefficient paths resulting from the classical boosted copula regression and the final models after applying the different approaches for reducing the model complexity on simulated data (more details on this example can be found in the supplement, Section A). Overall the coefficient paths of the different approaches yield similar final models. Applying probing leads to earlier

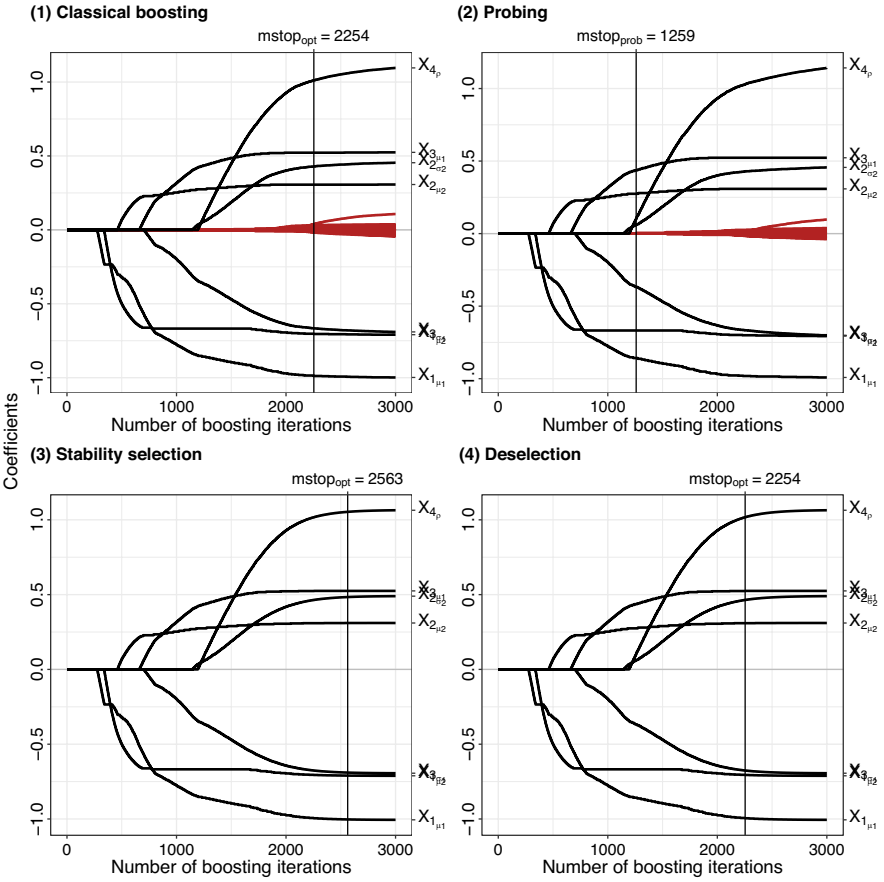

Fig. 1 The resulting coefficient paths along the number of boosting iterations for a simulated example (for more details see the supplement, Section A for the classical boosting, probing, stability selection, and the deselection approach (1%). The coefficient paths of the informative variables are colored in black, the noninformative in red. The intercept was removed for clarity. For stability selection and deselection, only the final model is plotted

stopping than the classical model with a stopping iteration at 1259 iterations. Therefore, the effect estimates are shrunken and fewer variables are included in the model (all informative but also one noninformative variable). As described in Sect. 2.2.1, the shrinkage of the effect estimates might not be optimal for predictive performance. The resulting model for stability selection is shown in the third plot, with selection frequencies across the B subsets for the different base-learners provided in the supplement (Figure A1). The performance of stability selection depends strongly on the choice of the parameters, here we choose $q = 20$ and PFER = 5, but for example smaller q and PFER would lead to worse results as most informative variables would not be included, leading to poorer predictive performance. The choice of q is informed by our comprehensive simulations

detailed in Sect. 3. For a representation of how different q values impact the results, we refer to Appendix B (Table A3), for details.

The deselection approach with a threshold value of 1% is similar to stability selection. Stable covariates are the ones with the highest risk reduction in the deselection approach. The final deselection model contains also here only the informative variables with the same number of boosting iterations as the classical model and similar coefficient estimates. The corresponding risk reduction for the different variables can be found in Figure A2 in the supplement with different threshold values (0.1 and 1%). Higher threshold values would lead to the elimination of informative variables, whereby for smaller values such as 0.1% (dotted line in the risk reduction plot Figure A2 in the supplement) noninformative variables would remain in the model.

2.3 Computational Details and Implementation

Boosting for distributional copula regression is implemented via the R package **gamboostLSS**. For tuning of m_{stop}, cross-validation, resampling techniques, or evaluation on a single test dataset can be used. The process is facilitated using a provided function that directly works on the model object.

From a computational and implementation perspective, probing can be very easily utilized, because no computationally intensive techniques for optimizing the stopping iteration and no additional tuning parameters are required. Stability selection for copula regression can be realized using the fitted boosting model and the `stabs()` function in the package `gamboostLSS`. One needs to specify two of the parameters beforehand, the per-family error rate and either the number of base-learners q or the threshold π_{thr}. The stopping iteration of the boosting model has to be chosen sufficiently large so that the q base-learners can be selected. The function returns the stable set of base-learner for each distribution parameter. To obtain the final model, one can again run a boosting model with only these stable base-learners. As in any classical statistical boosting model, the stopping iteration needs to be optimized by cross-validation, resampling techniques, or on an additional validation data set (if available). Moreover, the function encompasses various options for assumptions. It is important to note that the described approach in Sect. 2.2.2 does not involve any additional assumptions (`assumption = "none"`). The number of subsamples B should be sufficiently large to ensure reliable results. Typically, it is set to $B = 100$ [22]. By default, the implementation uses complementary pairs for subsampling with $B = 50$, which means $2 \cdot B$ subsamples in total.

The implementation of the deselection approaches is available at GitHub https://github.com/AnnikaStr/ComplRedBoostCop and is accessed with the `DeselectBoost()` function, which requires a boosting model with early stopping and the specification of an appropriate threshold value (e.g., 1%). The refitting of the model with the remaining base-learners to obtain the final model is already included in the function.

3 Simulations

To evaluate the performance of the different approaches for reducing the model complexity of boosted bivariate distributional copula regression models, we conducted a simulation study. We compared probing, stability selection, and the deselection of base-learners with a focus on their variable selection properties, the prediction performance, and runtime. Our specific objectives were to determine the following: (i) Can the variable selection approaches identify the truly informative variables while decreasing the number of false positives? (ii) How do the approaches perform in comparison to each other? (ii) Can the complexity of the model be reduced by simplifying complete additive predictors for distribution parameters to an intercept?

A detailed description of the simulation design of the following scenarios can be found in the supplement, Section B. Furthermore, here we provide a descriptive summary of the simulation results, while detailed numerical results can be also found in the supplement, Section B. All codes to reproduce the results can be found on GitHub https://github.com/AnnikaStr/ComplRedBoostCop. The simulations were conducted in `R` using the add-on package **gamboostLSS** for estimating the copula regression models. The **copula** and **gamlss** packages were used for data generation.

3.1 Simulation Design

To investigate these questions, we considered four different bivariate scenarios for continuous outcomes, each with five distribution parameters (marginal means μ_1 and μ_2, marginal variances σ_1^2 and σ_2^2, and association parameter ρ):

Scenario A Same simulation setup as in [9] with four informative variables $x_1, \ldots, x_4$. Cubic P-splines with 20 equidistant knots were included as base-learners. The log-normal and log-logistic distributions were used as marginal distributions.
Scenario B Modification of Scenario A:

1. σ_1 does not depend on explanatory variables.
2. ρ does not depend on explanatory variables.

Scenario C More informative variables: Ten informative variables for each distribution parameter $p_k = 10, k = 1, \ldots, 5$ with $x_1, \ldots, x_{50}$ with Gaussian marginal distributions for both outcomes. The base-learners correspond to simple linear models.

Detailed insights for each scenario are provided in Supplement B. A total of 100 simulation runs were performed for each simulation setting. For each scenario, $n = 1000$ observations were considered, where the covariates $x_1, \ldots, x_p$ were independently drawn from a uniform distribution on $(-1, 1)$. The simulations cover a

low-dimensional case ($p < n$) with $p = 20$ variables for **Scenario A**, B.1, and B.2 and $p = 200$ for **Scenario C**. Furthermore, a high-dimensional case ($p > n$) with $p = 1000$ variables was investigated for each scenario. The Gaussian, the Clayton, and the Gumbel copula were considered. For fitting the models, all covariates were considered for each distribution parameter simultaneously. The stopping iteration m_{stop} was optimized by minimizing the empirical risk on an additional validation dataset with 1500 observations. For all simulations, the step length of the boosting algorithm was set to a fixed value of $\nu = 0.01$ as suggested in [9] for boosting copula regression. For the deselection approach, we specified the threshold parameter τ with 0.1 and 1%. For stability selection, the number of variables to be included in the model was set as $q = 20$ and the per-family error rate was chosen to be PFER = 5. We employed $B = 50$ complementary pairs for subsampling. The m_{stop} for the boosting model for stability selection was set to five times the number of observations ($5 \cdot n$) ensuring that q base-learners can be selected. For the final model with only the stable covariates, the optimal stopping iteration was determined using an additional dataset, as in the classical boosted model. Note that due to the high computational cost, stability selection could not be applied for the high-dimensional settings.

To evaluate the prediction performance we used multivariate proper scoring rules, namely, the negative log-likelihood and the energy score. The energy score generalizes the continuous ranked probability score to multivariate quantities [7] and is defined as follows. Let $\boldsymbol{y} = (y_1, y_2)^T \in \mathbb{R}^2$ represent the vector of observations and let $\hat{F}$ denote a forecast distribution on $\mathbb{R}^2$. Assume $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n$ are n independent realizations from $\hat{F}$, where each realization is given by $\boldsymbol{Y}_i = (Y_{i1}, Y_{i2}) \in \mathbb{R}^2$ for $i = 1, \ldots, n$. The energy score is defined as

$$\text{ES}(F, \boldsymbol{y}) = \frac{1}{n} \sum_{i=1}^{n} \|\boldsymbol{Y}_i - \boldsymbol{y}\| - \frac{1}{2n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \|\boldsymbol{Y}_i - \boldsymbol{Y}_j\|,$$

where $\|\cdot\|$ denotes the Euclidean norm on $\mathbb{R}^2$.

3.2 Summary of Simulation Results

In Scenario A, B.1, and B.2, classical boosting is able to correctly select the informative variables for each distribution parameter, while noninformative variables were included mainly for the mean parameters (supplement, Section B.1, and B.2). However, in Scenario C, not all informative variables were selected for the dependence parameter and many noninformative variables were included for the mean and scale parameters (see the supplement, Section B.3).

In comparison, probing, stability selection, and deselection led to much sparser models in both low- and high-dimensional cases. Specifically, in Scenario A and Scenario B.1, the final models generally contained all informative variables except when using probing, which occasionally missed the informative variable for the dependence parameter ρ. Stability selection also occasionally missed the informative variable with a Clayton copula. With the deselection approach, all informative variables remained in the model. The fewest false

positives were obtained when τ was set to 1%, almost completely eliminating them. With a threshold value of 0.1% also many noninformative variables were excluded but to a lower extent than with 1% (see the supplement, Section B.1 and B.2). Similarly, in Scenario B.2, all approaches perform well in terms of true positive selection, although again probing failed to include all informative variables in some cases for σ_1. As before, none of the approaches can eliminate all false positives and excluding false positives for the association parameter is particularly challenging. However, this depends on the strength of the association between the outcomes. With stronger association, it is more difficult to eliminate the noninformative variables. With weaker association, the classical boosting tends to include few false positives for ρ, particularly for high-dimensional scenarios (see the additional setting for Scenario B.2 in supplement Section B.2 for more details).

In Scenario C, only for the Gaussian copula all informative variables were selected by classical boosting; for the other copulas, it was already difficult to select all true positives for the dependence parameter. Most false positives were included for the mean and scale parameters. The resulting models for probing and stability selection for Scenario C had difficulties in selecting the informative variables. The average number of true positives is relatively low for both approaches. The deselection approach with a threshold value of 0.1% only slightly influenced the average number of true positives. For all other parameters, the informative variables remained in the model in every simulation run. The number of noninformative variables were considerably reduced but there are still false positives left in the model. A higher threshold value would lead to a higher decrease in false positives but also to a reduction of correctly identified informative variables (see the supplement, Section B.3).

For the predictive performance on test data, evaluated with the negative log-likelihood and the energy score (smaller values are better), the deselection approach as well as stability selection had a comparable predictive performance and led to an improvement in the negative log-likelihood compared to the classical boosting for Scenario A, B.1 and B.2. For the energy score, the approaches resulted in similar values. Only probing showed a worse performance compared to the classical boosted model for the negative log-likelihood and the energy score (see the supplement, Section B.1 and B.2). For Scenario C, probing and stability selection led to a worse predictive performance due to the exclusion of informative variables. The deselection approach yielded an improvement in the negative log-likelihood for a threshold value of 0.1% and provided comparable performance to the classical approach regarding the energy score (see the supplement, Section B.3).

Overall, all approaches can drastically reduce the number of false positives in the final boosting model, whereby probing yielded the smallest runtime, as there is no need for an additional optimization of the stopping iteration. Due to its second boosting step, the deselection approach took slightly longer than the classic approach ($\approx$ 1–2 min). Stability selection had the longest runtime because B boosting models have to be fitted on the subsamples.

3.3 Characteristics of Enhanced Variable Selection Approaches

Based on our simulation study, we have summarized the key aspects of the different variable selection approaches to guide researchers in choosing a suitable method for specific data challenges. Table 1 provides an overview of probing, stability selection and deselection approaches, evaluating each method across several important characteristics, including the number of parameters to be specified, computational cost, and ease of use. Note that not every category can be considered by itself.

Probing emerges as the simplest method in terms of parameter specification and computational efficiency. It requires no specification of additional parameters and offers a low computational burden. However, this simplicity leads to limitations in variable selection, coefficient estimation and prediction performance. As discussed in Sect. 2.2.1, probing is intended to yield sparse models rather than optimized predictive performance. In addition, while probing is generally computationally efficient regarding runtime, it can cause memory problems when applied to large or high-dimensional data. This is because the method generates the additional probe variables, effectively doubling the dimensionality of the data set. As a result, even larger matrices must be stored and processed. This makes probing less practical for extremely high-dimensional scenarios, despite its otherwise efficient nature. Stability selection, on the other hand, provides effective variable selection and, after refitting with the stable covariates, provides reasonable coefficient estimation and comparable predictive performance to the classical boosting model. It offers error control, which is an advantage in many applications. However, these advantages come at the cost of increased complexity. Stability selection requires the specification of multiple parameters and is computationally intensive due to the repeated subsampling, making it more suitable for low-dimensional data sets. Deselection provides effective variable selection and coefficient estimation and achieves predictive performance comparable to the classical boosting model. It is relatively easy to use, similar to probing, and shows particular strengths when handling many potential variables, i.e., it is better scalable for large or high-dimensional data than probing

Table 1 Comparison of enhanced variable selection approaches: ✓ indicates an advantage, ○ represents a moderate or neutral characteristic and ✖ signifies a limitation or disadvantage

Characteristic	Probing	Stability selection	Deselection
Number of parameters	✓	✖	○
Computational cost	✓	✖	○
True positive selection	○	✓	✓
False positive reduction	✓	✓	✓
Coefficient estimates	○	✓	✓
Predictive performance	✖	✓	✓
Per-family error rate control	✖	✓	✖
High number of potential variables	○	✖	✓
True model not sparse	✖	✖	○
Simple to use	✓	○	✓

and stability selection. While deselection requires the specification of one parameter, this parameter is generally more intuitive to select. In general, $\tau = 0.01$ serves as a reasonable default for many scenarios. This combination of characteristics makes deselection a practical option for a variety of data problems.

In general, the selection of an appropriate variable selection method should be guided by a careful consideration of the specific requirements, the characteristics of the dataset and the available computational resources. As expected, all three methods perform better when the true underlying model is sparse.

4 Real Data Illustrations

4.1 Analysis of Fetal Ultrasound Data

Motivated by the analysis of fetal ultrasound data using boosted copula regression of [9], which resulted in rather large sub-models for the different distribution parameters, we examined and compared the variable selection and the predictive performance of this analysis with the models resulting from the enhanced variable selection techniques introduced in Sect. 2.2. The considered dataset was collected from 2006 to 2016 at the Department of Obstetrics and Gynecology of the Erlangen University Hospital and contains 6103 observations and 36 variables, including sonographic variables, e.g., abdominal anteroposterior diameter, abdominal transverse diameter, the interaction between these sonographic variables, and clinical variables, e.g., weight, height and body-mass index (BMI) of the mother. For more details on the data, we refer to [5].

The response variables of interest are the birth length and weight, which were modeled via copula regression with log-logistic marginal distributions and the Gaussian copula. We split the dataset into a training dataset with $n = 4,103$ observations and a test dataset for evaluation with 2000 observations. The step length was set to $\nu = 0.01$ and the stopping iteration was optimized by 10-fold cross-validation. All variables were considered for each distribution parameter. For continuous variables, cubic P-splines with 20 equidistant knots, a second-order difference penalty and 4 degrees of freedom were used as base-learners. Sex of the fetus and gestational diabetes were included via linear base-learners. Furthermore, we applied a gradient stabilization to ensure comparable gradients for the distribution parameters [12]. For the deselection approach, threshold values of 0.1 and 1% were considered. The parameters for stability selection were specified as $q = 20$ for the number of variables to be included in the model and PFER $= 5$ for the per-family error rate.

Table 2 shows the numbers of selected variables for each distribution parameter, the predictive performance in terms of the negative log-likelihood as well as the resulting optimal m_{stop}. An overview of the included variables for the different approaches can be found in the supplement, Section D. The classical boosted copula model selected almost all considered variables for the mean parameters μ_1 and μ_2. Fewer variables were selected for the shape parameters and the dependence parameter. The approaches for enhanced variable selection reduced the model complexity substantially and led to fewer included variables in the final models. The deselection

Table 2 Numbers of selected variables for distribution parameters μ_1, σ_1, μ_2, σ_2 and ρ, negative log-likelihood values and stopping iteration m_{stop} for classical boosting, deselection with threshold values of 0.1 and 1%, probing and stability selection

Method	μ_1	σ_1	μ_2	σ_2	ρ	−Log-Lik	m_{stop}
Classic	33	15	30	13	9	4156.58	5536
Deselection 0.1%	6	9	8	7	4	4184.07	5536
Deselection 1%	–	5	2	4	1	4843.41	5536
Probing	9	9	8	9	5	4654.58	808
Stability selection	5	3	3	3	–	4273.18	4194

approach with a threshold value of 1% deselected all variables for the mean parameter μ_1. Still, it contained variables for the other distribution parameters, more precisely interactions of the sonographic variables and the gestational age for the scale parameters. As expected, it resulted in a slightly worse negative log-likelihood compared to the classical approach.

With a smaller threshold value (0.1%), the final model contained variables for each distribution parameter and led to a comparable predictive performance than the classical boosted model. Here the model included mostly interactions of the sonographic variables as well but also a few other variables, e.g., sex for the location and gestational age for each distribution parameter except the dependence parameter. Probing resulted in a similar model as the deselection approach with 0.1%, but led to worse predictive performance. The most likely reason is the stronger shrinkage of the effect estimates due to the much smaller number of iterations. Via stability selection no covariates were selected as stable for the dependence parameter implying conditional independence of the two responses. This resulted in a poorer predictive performance compared to the classical model.

4.2 Joint Modeling of Cholesterol Phenotypes

We analyzed data from the UK Biobank (application number 81202), which is a large biomedical cohort study containing genetic and health information from over half a million British participants [35]. Using the boosting algorithm for distribution copula regression, we aim to model the polygenic contribution to the individual distributions of different phenotypes, but also to estimate the dependence between these phenotypes as a function of genetic variants. We want to identify the most relevant variants and therefore apply the methods presented in Sect. 2.2 to obtain sparse solutions.

The focus in the following is on three bivariate combinations of phenotypes, namely LDL (*Low-Density Lipoprotein*) and ApoB (*Apolipoprotein B*), LDL and cholesterol, and HDL (*High-Density Lipoprotein*) and ApoA (*Apolipoprotein A*). We considered these combinations because they have high empirical association based on an analysis of genetic blood and urine biomarkers in the UK Biobank [30], suggesting potential benefits in modeling these phenotypes jointly. All of these phenotypes are components of cholesterol metabolism. Cholesterol can be split mainly into two groups: i) LDL cholesterol, which is responsible for the transportation of cholesterol from the liver to various tissues and can be attached to specific receptors

on the cell surface with the help of ApoB, ii) HDL cholesterol, the counterpart of LDL which is accountable for the removal of excess LDL cholesterol from the body, with ApoA supporting this process [41].

The considered dataset for each combination of phenotypes consists of $n = 20{,}000$ randomly sampled observations with white British ancestry. Additionally, 15,000 observations were used for validation and 20,000 observations were used to evaluate the prediction performance via the negative log-likelihood. For each phenotype, 1000 variants were selected in a pre-screening step based on the largest marginal associations between the variants and the phenotype, which were computed with the PLINK2 function `--variant-score` [1, 26]. Variants with minor allele frequency not less than 1% were randomly sampled with the `--thin-count` function. Missing genotypes were imputed by the reference allele using the R package **bigsnpr** [25]. After the pre-screening, the dataset contains 1156 variants for LDL and ApoB (844 variants were selected for both phenotypes), 1179 variants for LDL and cholesterol (821 common variants), and 1249 variants for HDL and ApoA (751 common variants).

For each combination of two phenotypes, the marginal distributions and copulas were chosen which minimize the predictive risk (see Table 3). All variants were considered for each distribution parameter and incorporated with linear base-learners and step length $\nu = 0.01$. Stability selection unfortunately could not be applied to these data because of the high computational cost.

Table 3 shows the results for the joint analysis of the different combinations of phenotypes. Furthermore, Fig. 2 displays Manhattan-type plots for the phenotype combination LDL and cholesterol for every distribution parameter of the copula model. For each combination of phenotypes, the classical boosting approach selected several variants for each distribution parameter. Most genetic variants were selected for the location parameters. Each model included variants for the dependence

Table 3 Number of selected variants for distribution parameters μ_1, σ_1, μ_2, σ_2 and ρ, negative log-likelihood values and stopping iteration m_{stop} for the classical boosted model, the deselection approach with threshold values of 0.1 and 1%, and probing for the different combinations of phenotypes

Phenotype	Marginals	Copula	Method	μ_1	σ_1	μ_2	σ_2	ρ	−Log-Lik	m_{stop}
LDL	Log-logistic	Gaussian	Classic	441	26	386	67	47	10535.16	4965
ApoB	Gamma		Deselection 0.1%	121	2	71	9	15	10749.28	4965
			Deselection 1%	8	–	3	–	–	11647.92	4965
			Probing	–	–	–	–	–	14792.60	863
LDL	Log-logistic	Gumbel	Classic	286	89	266	100	44	31105.03	13,975
Cholesterol	Log-logistic		Deselection 0.1%	81	5	54	7	9	31090.53	13,975
			Deselection 1%	12	–	6	–	–	31817.39	13,975
			Probing	45	–	15	3	–	32834.92	5314
HDL	Log-normal	Gaussian	Classic	171	40	197	69	28	79820.43	1954
ApoA	Log-normal		Deselection 0.1%	81	15	83	24	9	79868.42	1954
			Deselection 1%	8	–	9	–	–	80337.71	1954
			Probing	113	20	185	36	6	80074.35	905

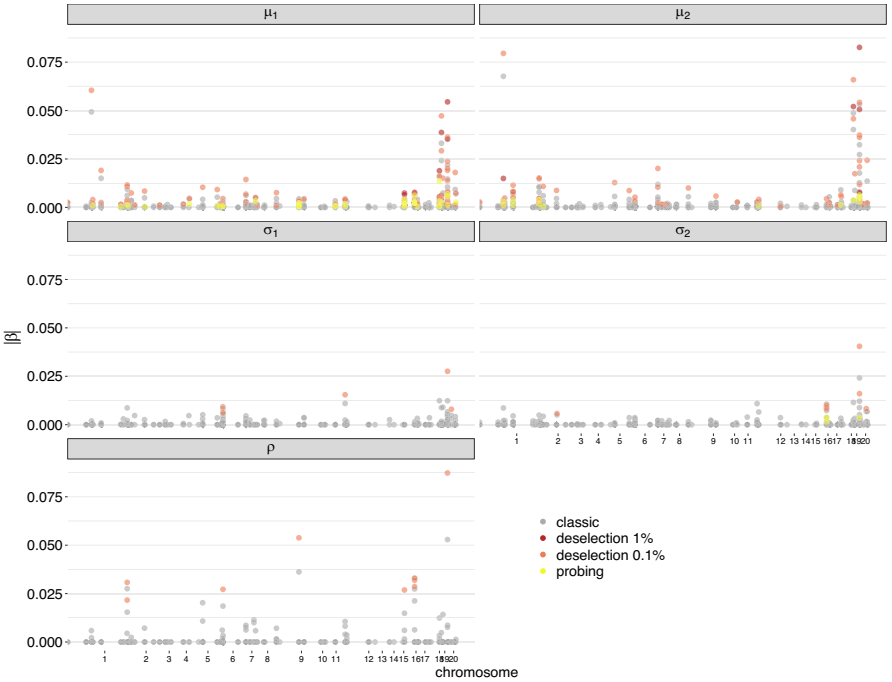

Fig. 2 Manhattan-type plots (chromosomes on x-axis) for the absolute coefficients of boosted copula regression for the joint analysis of LDL and cholesterol

parameter, indicating that different variants affect the associations between phenotypes and the potential benefit of modeling these phenotypes together. Considering the total number of selected variants, a relatively high number of the pre-filtered variants were included in the classical boosting model. In particular, for LDL and ApoB, almost half of all variants were included for the mean parameters. Despite the intrinsic variable selection of the boosting algorithm, we still obtain large models with a potentially difficult interpretation. Therefore, we aim to reduce the model complexity by enhancing variable selection.

With the deselection approach, the model complexity could be drastically reduced. When considering a threshold value of 1%, for all phenotype combinations only variants for the location parameters remained after deselection, which resulted in two univariate models. One can argue that this threshold value may be too strong for the data situation as there are several variants with only a small-to-medium effect (see for example Fig. 2) and therefore a minor impact on risk reduction. Also, owing to the pre-filtering, all variants in our analysis have some association with one of the outcomes, making it harder for single variants to pass the relative threshold. Using a smaller threshold value (0.1%) also led to sparser models, but for each distribution parameter several variants remained in the final model. The negative log-likelihood indicated a comparable predictive performance, whereby even a slight improvement in the performance for the phenotype combination LDL and cholesterol could be observed.

Probing also resulted in sparser models for each combination. In fact, for the phenotypes LDL and ApoB, no variants were included in the resulting model: in the first iterations, only the intercept was updated and stopping after 863 boosting iterations (when the first probe was selected) resulted in an intercept model, which led to a considerably worse predictive performance. For the other phenotypes, several variants were included after stopping when the first probe was selected. However, due to the smaller number of boosting iterations, the effect estimates were more shrunken (see Fig. 2 for LDL and cholesterol) and therefore the predictive performance deteriorated in comparison to the classical boosted model but also to the deselection approach, particularly for a threshold value of 0.1%.

5 Discussion and Conclusion

To reduce model complexity and to enhance variable selection for boosting multivariate distributional copula regression, we have integrated probing [36], stability selection [22], and also the recent deselection approach [33] in the boosting framework for this model class. This combination of classical boosting with all three approaches leads to considerably sparser models, thereby improving the interpretability of the obtained prediction models, which is desirable in practice [17, 42].

Regarding the specific approaches, the results of stability selection show similarities to the ones from deselection, even though the initial goals of the two methods differ. All three approaches perform better when the true model is sparse, whereby deselection can still lead to reasonable results when many variables are informative. The probing approach is the most favorable regarding computational runtime, but typically stops the algorithm also very early, leading often to underfitting and reduced predictive performance. As also observed in our first application on the weight and length of newborns, stability selection and deselection are more often able to maintain the predictive performance with smaller models. However, only deselection is also scalable to large, high-dimensional data as in our genetic application.

Our results additionally suggest that deselection not only yields much sparser models but can even lead to simpler univariate regression models in comparison to the classical boosted copula model in situations where the association parameter is close to zero. The proposed methods for enhanced variable selection could hence also represent tools for data-driven model choice [21]. The prediction performance typically does not improve after deselection but can lead to comparable accuracy as the classical boosting model with fewer predictors. Further improvements could be achieved in the future by optimizing the stopping iteration of the final boosting model, potentially leading to reduced shrinkage and slightly higher predictive performance. Stability selection works similarly, providing stable covariates that are then re-fitted in a final model with an optimized tuning parameter. The same principle could be applied to the deselection approach, which would, however, increase again the computational burden. In addition, also probing could be considered only as an extended method of variable selection followed by refitting the model only using the selected base-learners and tuning the stopping iteration. These

methodological extensions are beyond the scope of our current comparison, but offer promising directions for future investigation.

The deselection procedure is controlled via a threshold value τ, which represents the minimum amount of total risk reduction that should be attributed to a corresponding base-learner to avoid deselection. This can be interpreted as a threshold value for the importance of the particular predictor variable. Depending on the data situation, different thresholds may be appropriate; however, tuning is not straightforward because the true number of informative variables is not known in practice and the best model regarding predictive risk is naturally the one without any deselection. Further research is warranted on how to specify the threshold τ in this context.

Besides the practical advantages of the proposed tools, the natural limitation of all boosting algorithms applies: Due to early stopping and therefore shrinkage of the effect estimates, providing standard errors of the resulting coefficients is not an easy task as there are no closed formulas. To overcome this, one could apply permutation tests to carry out significance testing and provide p-values [20], but this would drastically increase the computational cost.

In conclusion, while statistical models should be as complex as needed to be able to capture the underlying nature of the data-generating process, they should also remain as simple as possible to facilitate interpretation [10]. To navigate this conceptual trade-off, we have proposed three competing approaches to simplify distributional copula regression models by reducing the model complexity and to enhance the variable selection properties of statistical boosting without considerably reducing the prediction accuracy of the resulting models.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s12561-025-09491-8.

Funding Open Access funding enabled and organized by Projekt DEAL. The work on this article was supported by the Deutsche Forschungsgemeinschaft (DFG, grant number 428239776, KL3037/2-1, MA7304/1-1).

Data Availability The code used for the simulations and the biomedical applications is available at GitHub https://github.com/AnnikaStr/ComplRedBoostCop. The fetal ultrasound data are not publicly available. However, to facilitate reproducibility, an artificial dataset that mimics the characteristics of the original data is also available in the GitHub repository. The genomic cohort data are available upon request from the UK Biobank at https://www.ukbiobank.ac.uk/.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Chang CC, Chow CC, Tellier LC, Vattikuti S, Purcell SM et al (2015) Second-generation PLINK: rising to the challenge of larger and richer datasets. GigaScience 4(1):7. https://doi.org/10.1186/s13742-015-0047-8
2. Dikheel TR, Alwa SH (2022) Using cross-validation, probing, and lasso in gradient boosting variable selection. Nat Volatiles Essent Oils 9:13620–13630. https://doi.org/10.53555/nveo.v9i1.5583
3. Fahrmeir L, Kneib T, Lang S (2004) Penalized structured additive regression for space-time data: a Bayesian perspective. Stat Sin 14(3):731–761
4. Fahrmeir L, Kneib T, Lang S, Marx B (2013) Regression: models, methods and applications, 1st edn. Springer, Berlin
5. Faschingbauer F, Dammer U, Raabe E, Kehl S, Schmid M et al (2016) A new sonographic weight estimation formula for small-for-gestational-age fetuses. J Med Ultrasound 35(8):1713–1724. https://doi.org/10.7863/ultra.15.09084
6. Freund Y (1995) Boosting a weak learning algorithm by majority. Inf Comput 12(2):256–285. https://doi.org/10.1006/inco.1995.1136
7. Gneiting T, Stanberry LI, Grimit EP, Held L, Johnson NA (2008) Assessing probabilistic forecasts of multivariate quantities, with an application to ensemble predictions of surface winds. TEST 17:211–235. https://doi.org/10.1007/s11749-008-0114-x
8. Griesbach C, Mayr A, Bergherr E (2023) Variable selection and allocation in joint models via gradient boosting techniques. Mathematics 11(2):411. https://doi.org/10.3390/math11020411
9. Hans N, Klein N, Faschingbauer F, Schneider M, Mayr A (2022) Boosting distributional copula regression. Biometrics 79(3):2298–2310. https://doi.org/10.1111/biom.13765
10. Heller GZ (2024) Simple or complex statistical models: non-traditional regression models with intuitive interpretations. Stat Model 24(6):503–519. https://doi.org/10.1177/1471082X241274405
11. Hofner B, Boccuto L, Göker M (2015) Controlling false discoveries in high-dimensional situations: boosting with stability selection. BMC Bioinformatics 16(1):144. https://doi.org/10.1186/s12859-015-0575-3
12. Hofner B, Mayr A, Robinzonov N, Schmid M (2014) 02. Model-based boosting in R: A hands-on tutorial using the R package mboost. Comput Stat 29:3–35. https://doi.org/10.1007/s00180-012-0382-5
13. Keil AP, O'Brien KM (2023) Considerations and targeted approaches to identifying bad actors in exposure mixtures. Stat Biosci 16(2):459–481. https://doi.org/10.1007/s12561-023-09409-2
14. Klein N (2023) Distributional regression for data analysis. Ann Rev Stat Appl. https://doi.org/10.1146/annurev-statistics-040722-053607
15. Klein N, Kneib T, Klasen S, Lang S (2015) Bayesian structured additive distributional regression for multivariate responses. J R Stat Soc C Appl 64(4):569–591. https://doi.org/10.1111/rssc.12090
16. Klinkhammer H, Staerk C, Maj C, Krawitz PM, Mayr A (2023) A statistical boosting framework for polygenic risk scores based on large-scale genotype data. Front Genet 13:1076440. https://doi.org/10.3389/fgene.2022.1076440
17. Markowetz F (2024) All models are wrong and yours are useless: making clinical prediction models impactful for patients. NPJ Precision Oncol 8:54. https://doi.org/10.1038/s41698-024-00553-6
18. Mayr A, Binder H, Gefeller O, Schmid M (2014) The evolution of boosting algorithms - from machine learning to statistical modelling. Methods Inf Med 53(06):419–427. https://doi.org/10.3414/ME13-01-0122
19. Mayr A, Hofner B, Schmid M (2012) The importance of knowing when to stop. A sequential stopping rule for component-wise gradient boosting. Methods Inf Med 51(2):178–186
20. Mayr A, Schmid M, Pfahlberg A, Uter W, Gefeller O (2017) A permutation test to analyse systematic bias and random measurement errors of medical devices via boosting location and scale models. Stat Methods Med Res 26(3):1443–1460. https://doi.org/10.1177/0962280215581855
21. Mayr A, Wistuba T, Speller J, Gude F, Hofner B (2023) Linear or smooth? enhanced model choice in boosting via deselection of base-learners. Stat Model 23(5–6):441–455. https://doi.org/10.1177/1471082X231170045
22. Meinshausen N, Bühlmann P (2010) Stability selection. J R Stat Soc B Stat Methodol 72(4):417–473. https://doi.org/10.1111/j.1467-9868.2010.00740.x
23. Nelsen RB (2006) An introduction to copulas. Springer, New York

24. Nguyen PH, Herring AH, Engel SM (2023) Power analysis of exposure mixture studies via Monte Carlo simulations. Stat Biosci 16(2):321–346. https://doi.org/10.1007/s12561-023-09385-7
25. Privé F, Aschard H, Ziyatdinov A, Blum MGB (2018) Efficient analysis of large-scale genome-wide data with two R packages: bigstatsr and bigsnpr. Bioinformatics 34(16):2781–2787. https://doi.org/10.1093/bioinformatics/bty185
26. Purcell S, Chang C (2015) Plink 2.0. www.cog-genomics.org/plink/2.0/
27. Rigby RA, Stasinopoulos DM (2005) Generalized additive models for location, scale and shape. J R Stat Soc C Appl 54(3):507–554. https://doi.org/10.1111/j.1467-9876.2005.00510.x
28. Schmid M, Hothorn T (2008) Boosting additive models using component-wise P-splines. Comput Stat Data Anal 53(2):298–311. https://doi.org/10.1016/j.csda.2008.09.009
29. Shah RD, Samworth RJ (2013) Variable selection with error control: another look at stability selection. J R Stat Soc Ser B Methodol 75(1):55–80. https://doi.org/10.1111/j.1467-9868.2011.01034.x
30. Sinnott-Armstrong N, Tanigawa Y, Amar D, Mars N, Benner C et al (2021) Genetics of 35 blood and urine biomarkers in the UK Biobank. Nat Genet 53(2):185–194. https://doi.org/10.1038/s41588-020-00757-z
31. Staerk C, Mayr A (2021) Randomized boosting with multivariable base-learners for high-dimensional variable selection and prediction. BMC Bioinformatics 22:1–28. https://doi.org/10.1186/s12859-021-04340-z
32. Stasinopoulos DM, Rigby RA, Heller GZ, De Bastiani F (2023) P-splines and GAMLSS: a powerful combination, with an application to zero-adjusted distributions. Stat Model 23(5–6):510–524. https://doi.org/10.1177/1471082X231176635
33. Strömer A, Staerk C, Klein N, Weinhold L, Titze S, Mayr A (2022) Deselection of base-learners for statistical boosting - with an application to distributional regression. Stat Methods Med Res 31(2):207–224. https://doi.org/10.1177/09622802211051088
34. Strömer A, Klein N, Staerk C, Klinkhammer H, Mayr A (2023) Boosting multivariate structured additive distributional regression models. Stat Med 42(11):1779–1801. https://doi.org/10.1002/sim.9699
35. Sudlow C, Gallacher J, Allen N, Beral V, Burton P et al (2015) UK Biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. PLoS Med 12(3):e1001779. https://doi.org/10.1371/journal.pmed.1001779
36. Thomas J, Hepp T, Mayr A, Bischl B (2017) Probing for sparse and fast variable selection with model-based boosting. Comput Math Methods Med 2017:1421409. https://doi.org/10.1155/2017/1421409
37. Thomas J, Mayr A, Bischl B, Schmid M, Smith A, Hofner B (2018) 05. Gradient boosting for distributional regression: Faster tuning and improved variable selection via noncyclical updates. Stat Comput 28:673–687. https://doi.org/10.1007/s11222-017-9754-6
38. Tian T, Sun J (2024) Variable selection for nonlinear covariate effects with interval-censored failure time data. Stat Biosci 16(1):185–202. https://doi.org/10.1007/s12561-023-09391-9
39. Umlauf N, Klein N, Zeileis A (2018) Bamlss: Bayesian additive models for location, scale, and shape (and beyond). J Comput Graph Stat 27(3):612–627. https://doi.org/10.1080/10618600.2017.1407325
40. Verhasselt A, Flórez AJ, Molenberghs G, Van Keilegom I (2024) Copula-based pairwise estimator for quantile regression with hierarchical missing data. Stat Model 25(2):129–149. https://doi.org/10.1177/1471082x231225806
41. Walldius G, Jungner I (2004) Apolipoprotein B and apolipoprotein A-I: Risk indicators of coronary heart disease and targets for lipid-modifying therapy. J Intern Med 255(2):188–205. https://doi.org/10.1046/j.1365-2796.2003.01276.x
42. Wyatt JC, Altman DG (1995) Prognostic models: clinically useful or quickly forgotten? Br Med J 311(7019):1539–1541. https://doi.org/10.1136/bmj.311.7019.1539
43. Yang L, Czado C (2022) Two-part d-vine copula models for longitudinal insurance claim data. Scand J Stat 49(4):1534–1561. https://doi.org/10.1111/sjos.12566

Authors and Affiliations

Annika Strömer[1,2] · Nadja Klein[3] · Christian Staerk[4,5] · Florian Faschingbauer[6] · Hannah Klinkhammer[2,7] · Andreas Mayr[1]

✉ Annika Strömer
annika.stroemer@uni-marburg.de

Nadja Klein
nadja.klein@kit.edu

Christian Staerk
staerk@statistik.tu-dortmund.de

Florian Faschingbauer
Florian.Faschingbauer@uk-erlangen.de

Hannah Klinkhammer
klinkhammer@imbie.uni-bonn.de

Andreas Mayr
andreas.mayr@uni-marburg.de

1 Institute for Medical Biometry and Statistics, University of Marburg, Marburg, Germany

2 Department of Medical Biometrics, Informatics and Epidemiology, University Hospital Bonn, Bonn, Germany

3 Scientific Computing Center, Karlsruhe Institute of Technology, Karlsruhe, Germany

4 IUF - Leibniz Research Institute for Environmental Medicine, Düsseldorf, Germany

5 Department of Statistics, TU Dortmund University, Dortmund, Germany

6 Department of Obstetrics and Gynecology, University Hospital of Erlangen, Erlangen, Germany

7 Institute for Genomic Statistics and Bioinformatics, University Hospital Bonn, Bonn, Germany