Statistical Modelling 2018; **18(3–4)**: 365–384

Boosting for statistical modelling: A non-technical introduction

Andreas Mayr[1,2] and Benjamin Hofner[3]
[1]Institut für Statistik, Ludwig-Maxilians-Universität, München, Germany.
[2]Institut für Medizininformatik, Biometrie und Epidemiologie, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany.
[3]Paul-Ehrlich-Institut, Langen, Germany.

Abstract: Boosting algorithms were originally developed for machine learning but were later adapted to estimate statistical models—offering various practical advantages such as automated variable selection and implicit regularization of effect estimates. The interpretation of the resulting models, however, remains the same as if they had been fitted by classical methods. Boosting, hence, allows to use an advanced machine learning scheme to estimate various types of statistical models. This tutorial aims to highlight how boosting can be used for semi-parametric modelling, what practical implications follow from the design of the algorithm and what kind of drawbacks data analysts have to expect. We illustrate the application of boosting in the analysis of a stunting score from children in India and a high-dimensional dataset of tumour DNA to develop a biomarker for the occurrence of metastases in breast cancer patients.

Key words: variable selection, high-dimensional data, model choice, statistical learning

Received June 2017; **revised** July 2017; **accepted** August 2017

1 Introduction

The annual International Workshop on Statistical Modelling (IWSM), which lays the foundation for this journal, has some never-changing traditions: welcome reception on Monday, the social event on Wednesday and of course the conference dinner on Thursday, which sometimes lasts until the early Friday morning. Another tradition of the workshop is the short course on Sunday, followed by an informal gathering. The one in Göttingen in 2014 will for many always be remembered as the night when Germany won the Football World Cup. But some participants might perhaps also remember the topic of the short course, it was *Boosting for Statistical Modelling* presented by two young Germans.

We are now a few years older, are still working with boosting and want to provide in this article an assessable and non-technical introduction to the topic, aimed at scientists that are not yet familiar with this tool.

Address for correspondence: Andreas Mayr, Seminar für Angewandte Stochastik, Ludwig-Maximilians-Universität, Akademiestr. 1, 80799 München, Germany.
E-mail: andreas.mayr@stat.uni-muenchen.de

 10.1177/1471082X17748086

Although boosting originally emerged from the field of machine learning (Freund, 1990), over the last few years a lot of methodological research focused on developing and extending boosting algorithms for statistical modelling (for a recent overview, see Mayr et al., 2014a,b, and the references therein).

The reason for the success of these *statistical boosting* approaches is first of all that they offer various practical advantages for high-dimensional data situations—a setting data analysts are nowadays often confronted with, for example, in the field of *omics* or other big data applications like in ecology. Furthermore, boosting yields statistical models with data-driven variable selection, implicit penalization (Hepp et al., 2016) and shrinkage of effect estimates (similar to the least absolute shrinkage and selection operator *LASSO*; Tibshirani, 1996). These properties of the algorithm are controlled by one single parameter: the number of boosting iterations to be carried out, which reflects the trade-off between bias and variance. The boosting approach is robust against multicollinearity issues and very flexible when it comes to different types of effects (Hofner et al., 2014b): the resulting models can include not only linear and non-linear but also spatial or random effects. A drawback of statistical boosting is that due to the particular design of the algorithms, there are no estimates for the standard errors of resulting effects directly available.

From a methodological perspective, statistical boosting is the link between the areas of computer science and statistical modelling. It bridges the gap between two rather different points of view on how to gather information from data (Breiman, 2001): on the one hand, there is the classical statistical modelling view that focuses on structured additive models to *describe* the outcome in order to find an approximation of the underlying stochastic data generation process. On the other hand, there is the machine learning and predictive modelling view that focuses primarily on algorithmic models to *predict* the outcome while avoiding structural assumptions—treating the nature of the underlying process as unknown (cf. Mayr et al., 2017a). As statistical boosting, in fact, is a machine learning algorithm which is used to estimate classical statistical models, it inherits characteristics from both worlds. Also the terminology is partly influenced by machine-learning jargon, hence, we incorporate a Glossary table at the end of this article, describing often used vocabulary in the literature on boosting.

In this tutorial article, we want to give a non-technical introduction on (a) how to apply boosting for statistical modelling, (b) what choices have to be made for the analysis and (c) in which practical situations it might be helpful and in which not. We will focus on boosting algorithms that are based on *gradient* boosting (Bühlmann and Hothorn, 2007). A second group of algorithms uses a slightly different design (e.g., *likelihood-based* boosting; Tutz and Binder, 2006); however, the overall structure is the same: many features discussed here hence carry over also to those related approaches.

We will highlight the abilities of statistical boosting via two exemplary data analyses: one is a widely known dataset on childhood malnutrition in India, containing continuous and categorical predictors, as well as a regional effect. The second one is a high-dimensional microarray dataset for the prediction of breast cancer with more explanatory variables than observations ($p \gg n$). Both datasets are

publicly available; the underlying R-code to reproduce the analyses presented here is published via the online supplementary material.

2 Data

2.1 Childhood malnutrition in India

Childhood malnutrition in India is not necessarily a consequence of extreme poverty but can also be linked to cultural factors with strong regional differences (Arnold et al., 2009). Following a bulletin of the World Health Organization (WHO), growth assessments are the best way to define the health and nutritional status of children (de Onis et al., 1993). Stunted growth is defined as a reduced growth rate compared to a standard population and is considered as the first consequence of malnutrition of the mother during pregnancy, or malnutrition of the child during the first months after birth. Stunted growth is often measured via a Z score, which compares the anthropometric measures of the child with a reference population. In our case, we compare the height of children (H_i) to the median height in the reference population divided by the standard deviation of height in the reference population:

$$Z_i = \frac{H_i - \text{med}(H)}{s(H)}$$

This Z score will be denoted as *stunting score* in the following. Negative values of the score indicate that the child's growth is below the expected growth of a child with normal nutrition. The stunting score will be the outcome (response) variable in our data example: we analyse the relationship of the mother's age and body mass index (BMI) as well as age of the child with stunted growth resulting from malnutrition in early childhood. Furthermore, we will investigate regional differences by including the district of India in which the child is growing up. The raw distribution of the average stunting score per district is depicted in Figure 1.

The variables used here are only a very small subset of the available variables. For an in-depth analysis based on boosted quantile regression, see Fenske et al. (2011) and Fenske et al. (2013). The dataset that we use in this analysis is a random subset of 4 000 observations based on the Standard Demographic and Health Survey, 1998–99, on malnutrition of children in India, which can be downloaded after registration from http://www.measuredhs.com.

2.2 DNA signature to predict metastases of small node-negative breast carcinoma

Modern biomedical and epidemiological studies often gather vast amounts of molecular or genetic, so-called *omics*, data. The overall aim is, on the one hand, to identify the most informative variables from these large datasets in order to investigate their influence on clinical outcomes. On the other hand, these selected variables are

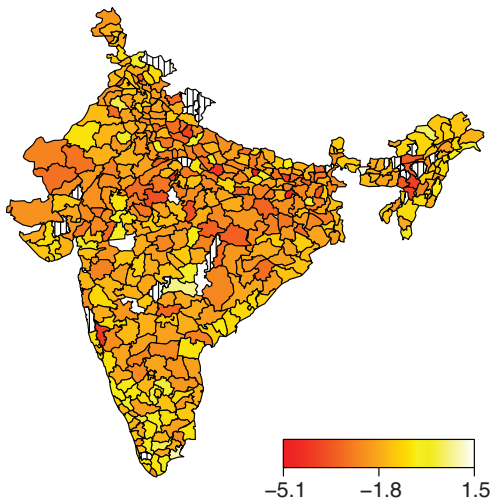

Figure 1 Geographical distribution of the stunting score; the raw mean per district is depicted, ranging from dark red (low stunting score) to light yellow (higher scores). Dashed regions represent regions without data

then also used to construct some kind of prediction rule (i.e., a biomarker) based on a small subset of omics variables sometimes combined with other clinical variables, which can later be used to improve the treatment of patients based on their individual risk.

Statistical boosting algorithms in this context are favourable, because they can fulfil both tasks simultaneously by selecting and fitting a prediction model at the same time (cf. Mayr and Schmid, 2014).

In the dataset for this example we have tumour DNA from the invasive ductal carcinomas (the most common form of breast cancer) without axillary lymph node involvement (T1T2N0) in 168 patients. The tumour DNA was compared to non-tumour DNA via comparative genomic hybridization (CGH). During 5 years after the diagnosis, 111 of these patients developed metastases and 57 did not. The aim of the analysis is now to develop a prediction rule that discriminates well between these two groups based on 2 905 DNA features from CGH arrays. The original dataset is available on GEO (https://www.ncbi.nlm.nih.gov/geo, accession code GSE19159) and is based on the work of Gravier et al. (2010). A copy of the data is also available via GitHub (Ramey, 2016).

3 Statistical boosting

Let us consider statistical models for the two examples given earlier. To model malnutrition of Indian children, that is, the z-transformed stunting score (in the following denoted as outcome y), we can specify linear effects for the age (x_1) and

BMI (x_2) of the mother, and the age (x_3) of the child. As the effects on stunting might be more complex we can also think about flexible, smooth effects. Additionally, we want to model the spatial variation to capture additional heteroscedasticity, that is, the effect of the district (x_s).

Hence, a model for the conditional expectation could look as follows:

$$\mathbb{E}(y_i \mid \boldsymbol{x}_i) = \beta_0 + \sum_{j=1}^{3} f_j(x_{ij}) + f_s(x_{is}), \quad i = 1, \ldots, n \tag{3.1}$$

where β_0 is the intercept, f_j is a smooth effect and f_s is a (smooth) spatial effect. For the conditional distribution of the stunting score, we assume a Gaussian distribution.

In order to predict if patients suffering from small node-negative breast cancer will develop metastases based on CGH array data, we can use a linear logistic regression model (for a binomial distribution):

$$\text{logit}(\mathbb{E}(y_i \mid \boldsymbol{x}_i)) = \text{logit}(\mathbb{P}(y_i = 1 \mid \boldsymbol{x}_i)) = \boldsymbol{x}_i^\top \boldsymbol{\beta}, \tag{3.2}$$

where $\mathbb{P}$ denotes the conditional probability that $y_i = 1$ given the covariate vector $\boldsymbol{x}_i$. Regression coefficients are denoted as $\boldsymbol{\beta}$, and $\text{logit}(p) = \log \dfrac{p}{1-p}$.

Due to the modular nature (see Section 3.6), statistical boosting allows us to fit both these very different models without changing anything in the structure of the algorithm. In the stunting model, we are faced with an (Gaussian) additive model which contains additionally a spatial effect. In the second example, we want to fit a linear logistic model with much more genetic predictors than observations. In this high-dimensional setting, statistical boosting will be used not only to estimate the model but also to select the most influential DNA features (variable selection, see Section 3.2).

3.1 The general design of the algorithm

First and foremost, statistical boosting can be seen as one of many possible algorithms to fit a statistical model such as least squares regression or maximum likelihood. Essentially, the boosting algorithm minimizes a specific *loss function* (which quantifies the discrepancy between observed data and the model) in an iterative fashion.

It mimics least squares optimization if we use a quadratic error loss or it mimics maximum likelihood estimation if we use the negative likelihood as loss function. However, due to the iterative nature of the algorithm, special strengths and drawbacks exist which have to be kept in mind. We will discuss these after a bit more formal introduction of the method.

Next to the loss function to be minimized, the user needs to specify the effect types, for example, linear or smooth effects, which are used to model the influence on the outcome. These effect types are specified via so-called base-learners (see Glossary). We will discuss specifics of base-learners in Section 3.3.

Boosting now proceeds in an iterative fashion to minimize the loss function (see Figure 2 for a schematic overview). In every iteration, one computes the negative gradient of the loss function (e.g., the first derivative of the likelihood) and evaluates it for the outcome and the model at the current iteration, which can be considered as some kind of residuals. Next, each of the base-learners is fitted separately to the negative gradient. Only the best-fitting base-learner, that is, the most influential effect, is selected for an update of the model. One recomputes the negative gradient ('the residuals') and repeats the procedure by fitting again *all* base-learners (including the one we just added) to the updated negative gradient.

There are two main differences to forward stepwise regression, the first is that we do not use the outcome itself but a transformation of the outcome by using the negative gradient of the loss function. This can be considered as using problem specific residuals which are derived from the loss function.

Another difference is that we allow a base-learner to be used multiple times. To avoid overfitting and to slowly move towards the minimal loss, the estimated effect of the selected base-learner is multiplied by a constant step-length ν (usually $\nu = 0.1$), before being added to the model.

The major tuning parameter is the number of boosting iterations (often called m_{stop}). The final model in iteration m_{stop} is then simply the sum of all selected effects multiplied by the step-length. Hence, we obtain an additive model, which can be interpreted in the usual fashion as we will show in our examples later. It can be shown that this approach minimizes the loss function and the estimated model converges towards the maximum likelihood (or least squares) estimate if $m_{\text{stop}} \to \infty$ (Bühlmann and Hothorn, 2007).

3.2 Model tuning and variable selection

In order to tune the model via the number of boosting iterations m_{stop}, a process also often denoted as model selection or hyperparameter optimization, one usually uses cross-validation techniques such as bootstrap, k-fold cross-validation or subsampling (Mayr et al., 2012b). Common to all these techniques is that one fits the model on a random subset of the data and uses the remaining data to evaluate the performance of the model. We do this by evaluating the loss function of the model for the observations not used for model fitting (the so-called out-of-bag data) for a sequence of boosting iterations. An example of predictive risk (see Glossary: risk is just the observed loss) for the breast cancer application based on 25-fold bootstrap is given in Figure 3. The underlying idea is that we are not really interested in the best model-fit for the observed data, but in the best generalization of the underlying structure, which hence also works well for new observations.

As in each boosting iteration only one single base-learner is updated and added to the model, and as we can add one base-learner more than once, we have at most m base-learners selected up to iteration m. If we only use a small number of boosting iterations, we thus select only the most important variables. An example of the progression of the regression coefficients from the breast cancer example over the

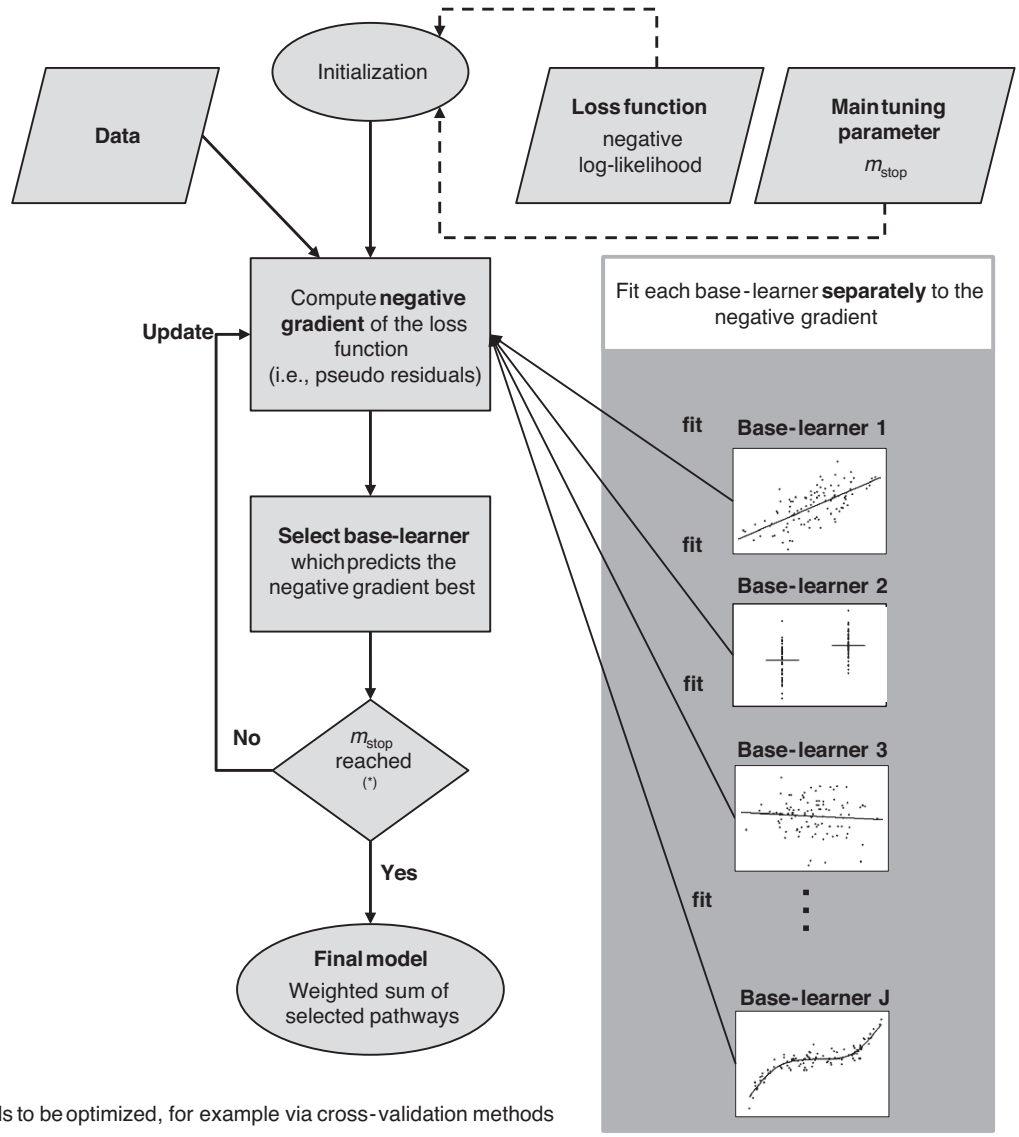

Figure 2 Graphical representation of the main features of the boosting algorithm. First, the user needs to specify the loss function to determine the model to be fitted. Other parameters might need specification. Usually, this is only the number of iterations m_{stop}. The negative gradient of the loss function is computed and serves as outcome. Now, we fit each of the base-learners separately to the negative gradient: for example, a linear model (base-learner 1), a linear model for a categorical effect (base-learner 2), a linear model where the variable has no real influence on the outcome (base-learner 3), or a smooth effects model (base-learner J). We select the best fitting base-learner and redo the whole process with an updated negative gradient unless m_{stop} is reached

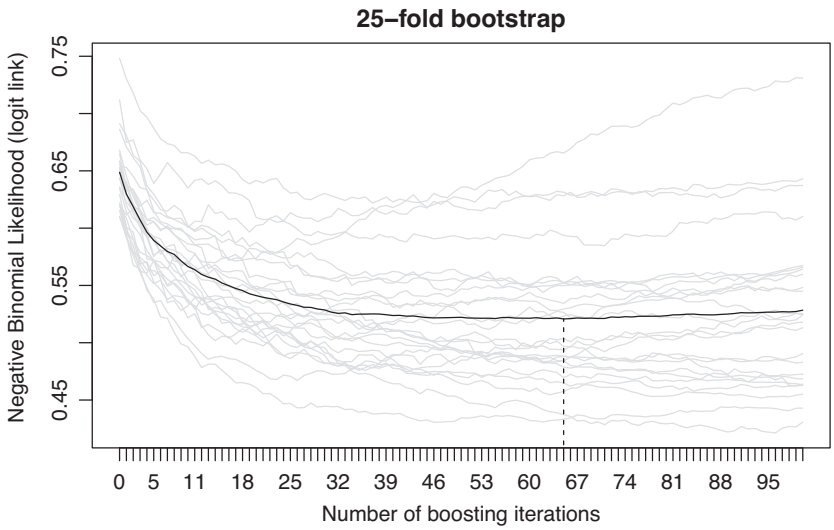

Figure 3 Predictive risk for the breast cancer example via 25-fold bootstrap. Grey lines indicate the performance on out-of-bag observations, for the single models while the black line is the average over all 25 models. The dashed line indicates the minimal risk, that is, the optimal model complexity

first four iterations is given in Figure 4. The number of iterations plays a similar role as the penalty parameter λ in case of L_1 regularization (e.g., LASSO), where the number and size of coefficients increases with decreasing λ (see Hepp et al., 2016, for details).

What is particularly interesting in the context of statistical modelling is that, in fact, m_{stop} is the only tuning parameter that is optimized. It controls not only the variable selection properties of the algorithm but also the implicit penalization for the different types of effects, as will be outlined in the next section.

3.3 Type of effects

So far, we only very briefly sketched base-learners as representatives of the effect type to be modelled. Each base-learner is a simple regression model relating the predictor to the outcome. For linear effects, we use linear base-learners, that is, simple linear regression models. These are always fitted to the negative gradient via least squares regression, irrespective of the nature of the actual outcome and the regression problem one is interested in. The conditional distribution is captured via the loss function which is to be minimized.

Similarly, we can define base-learners for other effect types such as smooth effects, where we use P-splines (Schmid and Hothorn, 2008) which are fitted to the negative gradient via penalized least squares. Bivariate P-splines can be used to model spatial effects or smooth interaction surfaces. Spatial effects of regions can be defined via Markov random fields (see India example and Sobotka and Kneib, 2012).

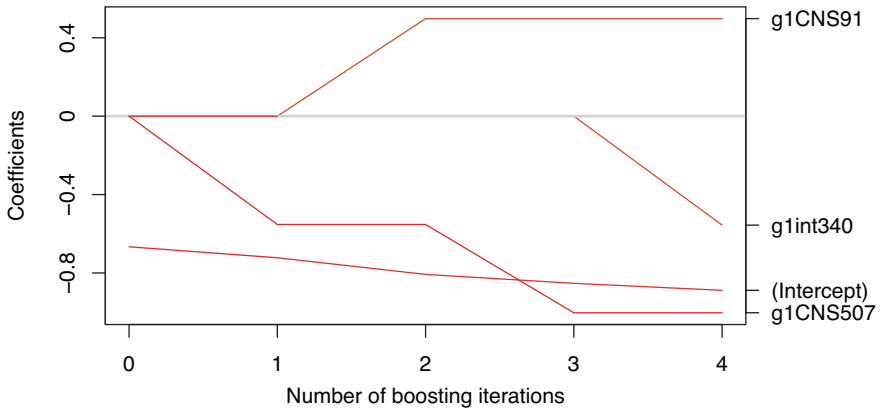

Figure 4 Coefficient paths for first five boosting iterations in the breast cancer example. One can see that in the first step `g1CNS507` is selected (with a negative effect) as all other effects are still zero. The intercept is implicitly updated in each iteration. In the second iteration, `g1CNS91` is updated and the effect of `g1CNS507` stays unaltered. In the third iteration, `g1CNS507` is updated again, followed by `g1int340`

Other base-learners include random effects base-learners (Kneib et al., 2009, Web Appendix), base-learners for constrained effect estimates including monotonic categorical effects and monotonic P-splines, convex or concave categorical effects and P-splines (Hofner et al., 2014a), and cyclic effects (e.g., for temporal effects with recurring pattern; Hofner et al., 2014a). All these effects are fitted via penalized least squares base-learners. For more technical details on each of the base-learners and advanced use cases we refer to the given citations and to Hofner et al. (2014b).

It has to be noted, however, that inside the different base-learners no further hyperparameters are needed to be optimized or tuned: they are iteratively applied with constant penalty terms; as the base-learner can be updated multiple times, also the final level of penalization (e.g., the final smoothness in case of a non-linear effect) depends on the number of boosting iterations m_{stop} (see Figure 5 for an example): The same spline (fixed equidistant knots, constant degrees of freedom) is updated various times. As a result, the spline coefficients are simply summed up, the smoothness reduces from iteration to iteration and the spline will eventually overfit. The final level of penalization, hence, depends only on the number of boosting iterations m_{stop}, all other parameters are kept constant.

3.4 Model choice

As discussed in Section 3.2, boosting with early stopping (see Glossary) leads to variable selection. If we specify multiple base-learners for a single variable, boosting selects the best fitting base-learner(s) and thus conducts model selection.

A common scenario is that researchers want to fit a model which is as simple as possible, yet as flexible as necessary. In that case, one can specify linear base-learners and smooth base-learners for each continuous variable. The boosting algorithm now

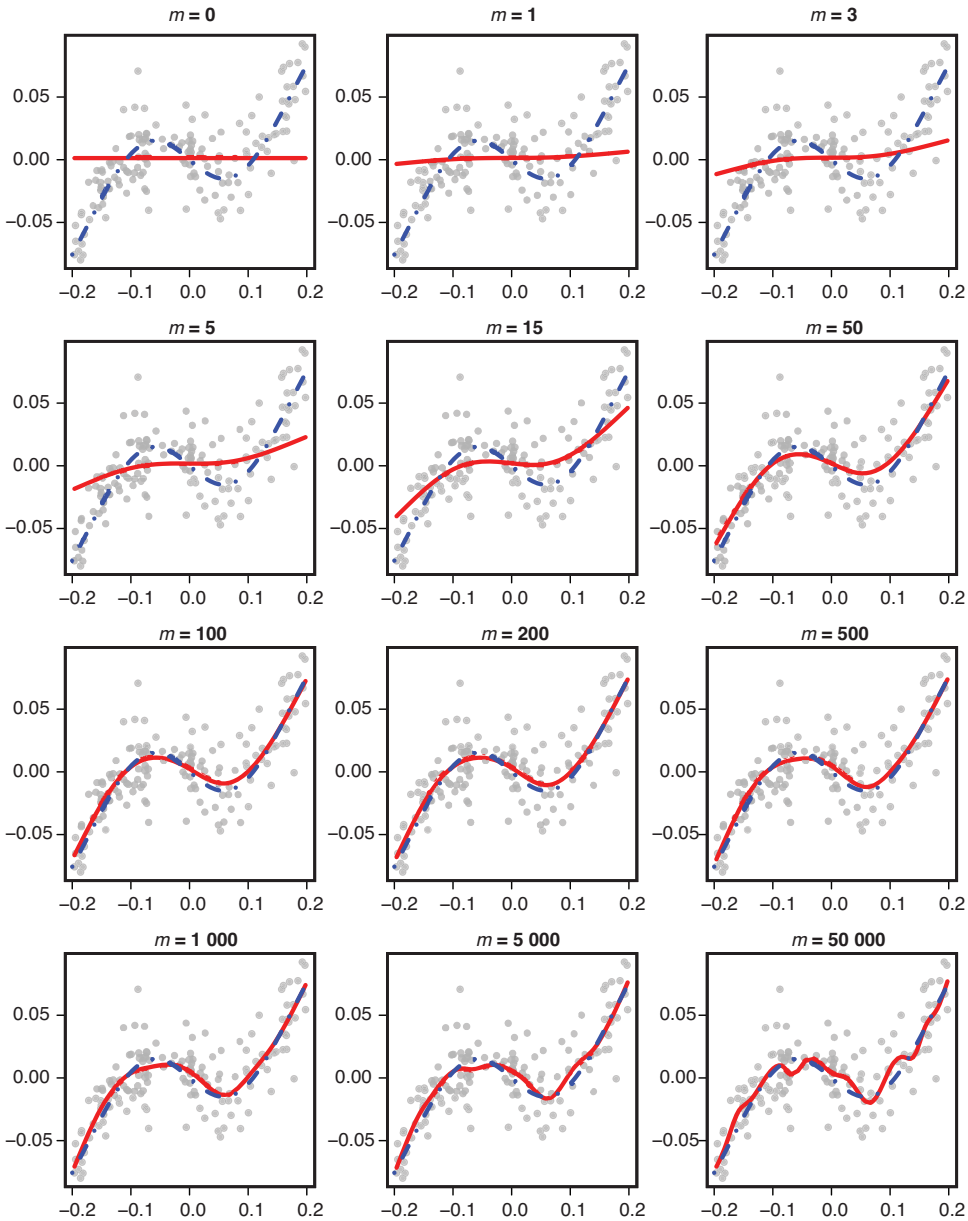

Figure 5 Example on how m_{stop} affects the smoothness of non-linear effect estimates (simulated data). The dash-dotted blue line refers to the true effect, the solid red lines represent the estimated effect between 0 and 50 000 boosting iterations. The optimal number of boosting iterations would be around 100. Afterwards the model (slowly) starts to overfit the training data

decides in each iteration if a linear effect is sufficient or a smooth effect is needed. The final effect of a variable is then the sum over the linear and smooth effect (if both were chosen at least once). If for one variable only the linear base-learner was selected, one can conclude that a linear effect is sufficient to model the influence of that variable. If none of the base-learners was chosen, this variable seemingly has no influence on the outcome (given the other variables already in the model).

This idea can also be extended to interactions (or spatial effects), where one can separately model linear marginal effects, linear interaction effects, smooth marginal effects and smooth interaction effects (see, e.g., Kneib et al., 2009).

Note that smooth effects are usually preferred over linear effects for the same variable as they contain linear effects as a special case but offer more flexibility. To reduce this selection bias, one can re-parameterize smooth P-splines such that one subtracts the linear effect and only models smooth deviations from linearity. For technical details, regarding the choice for the degrees of freedom to ensure unbiased selection, see Hofner et al. (2011).

3.5 Model classes

In our data examples, we want to model a classical Gaussian additive model for the stunting score and a generalized additive logit-model for the development of metastases in breast cancer patients. These different model classes, however, can still be estimated by basically the same algorithm. As statistical boosting does not fit the base-learners on the actual observations, but on the gradient vector of the loss function, only this loss function, determines the particular regression setting.

As a result, statistical boosting can be used to find any GAM model by using the negative log-likelihood as loss function; however, the scope is much broader: the only restriction for the loss function is that it should be convex and differentiable (first order) with respect to the model term. Statistical boosting can, hence, be also adapted to fit regression situations that are not based on a likelihood. A popular example for such a scenario is quantile regression (Koenker et al., 1994), which relies on the optimization of the weighted absolute deviation from observations and fitted quantiles (i.e., the so-called *check-function*) that can be optimized by boosting (Fenske et al., 2011; Mayr et al., 2012c). Another example is the *C*-index introduced by Harrell et al. (1982), which can be used as a discriminatory measure for survival data (Harrell et al., 1984): via using the negative *C*-index as loss function one can optimize statistical models with respect to their ability to discriminate well between patients with longer and shorter survival times (Mayr and Schmid, 2014; Mayr et al., 2016).

But statistical boosting can also be adapted to even more complex model classes, such as generalized additive models for location scale shape (GAMLSS; Rigby and Stasinopoulos, 2005) or joint models (Waldmann et al., 2017). In GAMLSS, not only the expected value of a distribution, but all parameters are modelled to the covariates (see also the tutorials by Stasinopoulos et al. (2018) and Umlauf and Kneib (2018) in this special issue). As a result, the boosting algorithm needs to estimate various models

simultaneously by circling through the different dimensions (Mayr et al., 2012a). For a recent tutorial on this type of boosting algorithms, see Hofner et al. (2016).

3.6 Modular nature of boosting

One reason for the growing methodological research on statistical boosting algorithms (Mayr et al., 2014b) is that they are relatively easy to extend towards new regression settings (i.e., to include new loss functions) or to incorporate new types of effects (i.e., new base-learners). In fact, if one flicks through the different tutorials in this special issue, all model classes presented there can also be fitted by boosting: Not only quantile regression and distributional regression as mentioned earlier but also conditional transformation models (Hothorn, 2018) via **mboost**, functional regression (Bauer et al., 2018) via the package **FDboost** (Brockhaus et al., 2017) and advanced survival analysis (Bender et al., 2018; Berger and Schmid, 2018, see the references therein).

For practical purposes, another advantage is the modular nature of the algorithm (Bühlmann et al., 2014), which basically allows to combine any base-learner with any type of loss function. In other words, all implemented regressions settings and all new extensions (e.g., GAMs, quantile regression and C-index; see Section 3.5) can be fitted with any type of covariate effect (e.g., linear, smooth, spatial and monotonic; see Section 3.3) without the need to adapt the algorithm itself.

As the base-learners are applied to the negative gradient of the loss function, the structure of the algorithm does not change at all if we simply replace one loss function with another. For example, if we want to fit the model for the stunting score not regarding the expected value but the median of the distribution, all we have to do is to replace the L_2 loss with the L_1 (Fenske et al., 2011). A nice example is given in the tutorial article by Waldmann (2018) in this special issue.

3.7 Implementation

This modular structure of the algorithm carries over to its implementation in the statistical programming environment R (R Development Core Team, 2016). The most flexible add-on package for the gradient boosting variant we are presenting here is the **mboost** package (model-based boosting, Hothorn et al., 2016).

The main **mboost** functions to carry out boosting are `glmboost()` for linear models and `gamboost()` for additive models. The different types of effects (i.e., the base-learners) can be specified in a formula environment:

```
model1 <- gamboost(stunting ~ bbs(mage) + bbs(mbmi) + bbs(cage)
                   + bmrf(mcdist, bnd = neighborhood),
                   data = india, family = Gaussian())
```

In this case, `stunting` is the outcome, `mage`, `mbmi` and `cage` are included via penalized B-spline base-learners (`bbs()`), and `mcdist` via a Markov random

field base-learner (`bmrf()`). The model class is specified (in this case for a classical Gaussian regression) via `family = Gaussian()`.

To fit an initial model, the user, hence, has to specify (a) with which type of effects the covariates should be included in the model via the base-learners (or stick to linear models via `glmboost()`) and (b) what type of loss function should be applied. When these two points are selected, the next choice is the number of boosting iterations. Per default, `gamboost` performs 100 iterations; however, it is not recommended to simply use this ad-hoc value for the final model. Tuning the model via m_{stop} (see Section 3.2) can be done via the `cvrisk()` function which provides different resampling procedures (default is 25-fold bootstrap) to select the best m_{stop}:

```
cvr <- cvrisk(model1, grid = 1:2000)
```

In this case, `cvrisk()` automatically searches for the best model (with respect to the predictive risk) on a grid from 1 to 2 000 iterations. Details on **mboost**, including tables with available loss functions and base-learners, can be found in a recent tutorial (Hofner et al., 2014b).

4 Examples

In order to apply statistical boosting in practice, the data analyst, hence, faces three major choices: (a) loss function, (b) base-learners and (c) the stopping iteration. The latter, however, is typically tuned in a data-driven fashion. In our two examples, modelling of the stunting score and the DNA signature to predict metastases in breast cancer patients, we will particularly highlight how these three choices are performed. The code to reproduce both analyses and the figures in this article is included as supplementary material.

4.1 Childhood malnutrition in India

For the stunting score, which in fact is based on a z-transformation, a sensible loss function could be the L_2 loss leading to classical Gaussian regression of the mean. Regarding the base-learners, we chose to incorporate the continuous variables BMI and age of the mother and age of the child as potentially non-linear predictors via P-spline base-learners. Additionally, we included a spatial effect for the 422 districts of India via a Markov random field base-learner, modelling the neighbouring structure of those districts to account for spatial variation that is not explained by subject-specific variables.

We fitted the model (see model 1 in Section 3.7) on all 4 000 observations and selected the optimal stopping iteration via 25-fold subsampling (with sampling probability $\pi = 0.5$). The optimal value for m_{stop} was 1 941. This relatively large value for m_{stop} is typical for situations with large n but small p where boosting shows

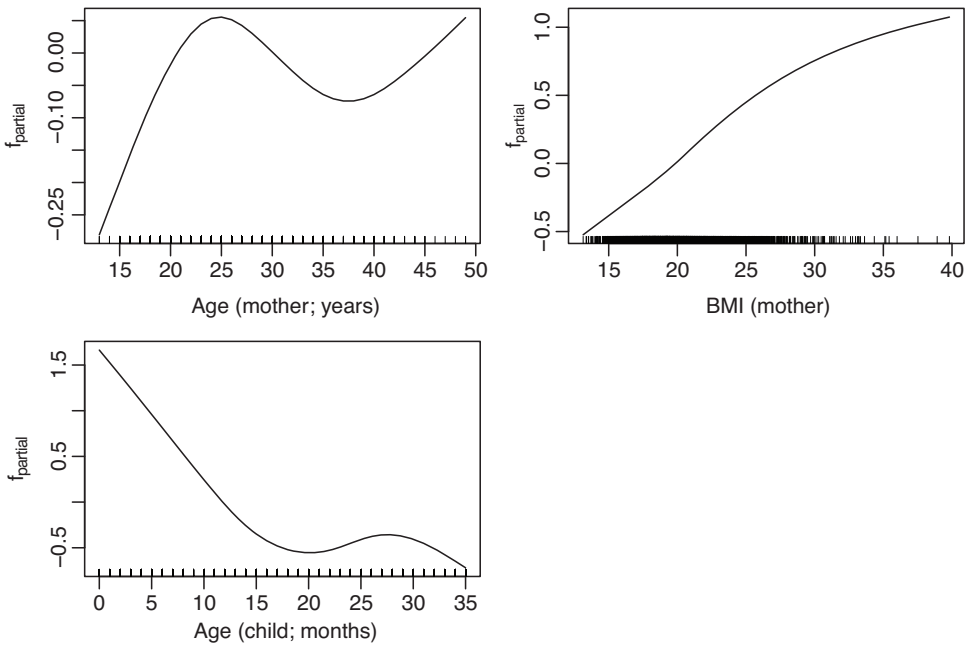

Figure 6 Smooth estimated partial effects of the model for malnutrition in India

a rather slow overfitting behaviour (Bühlmann and Hothorn, 2007). We then refitted the model with this m_{stop}, see Figure 6 for the resulting smooth effects and Figure 7 for the spatial effect estimates.

Following our model, the problem of stunted growth of children in India is most pronounced for young mothers (negative effects for mothers from the age of 15 to 21 years) with low BMI (negative effects for mother with BMI from 15 to 20). When it comes to the age of the children, the greatest risk for stunted growth seems to be reached around the second birthday of the child.

Figure 7 reveals regional effects on the stunting score. Following our model, stunted growth is most pronounced in the northern regions of India (negative effect). In the south and also in the north-eastern regions, on the other hand, the growth of children seems to be above average. These regional variation, could be a marker for economic or cultural differences between regions of India.

4.2 DNA signature to predict metastases of small node-negative breast carcinoma

For the breast cancer data, we first fitted a prognostic model for the development of metastases on all patients. As the outcome is binary, we chose as loss function the negative log-likelihood of the Bernoulli distribution. For all potential 2 905 DNA predictors, we considered simple linear models as base-learners.

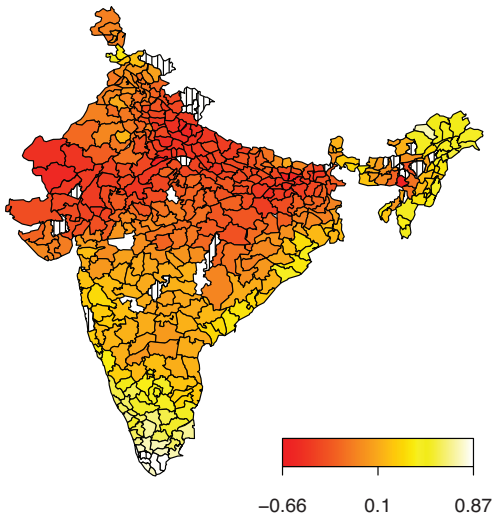

Figure 7 Estimated partial spatial effect of the districts on malnutrition in India

The optimal stopping iteration was obtained via 25-fold bootstrap and was estimated as 65 (see Figure 3). This early stopping led to the selection of only 29 variables. For an overview of the selected variables with effect estimates see the corresponding table in the supplementary material.

With these 29 features, we could predict the establishment of metastases with very high accuracy. The AUC (area under the curve, a value around 0.5 refers to a non-informative prediction rule while a perfect discrimination results in a value of 1) on the same dataset was 0.981. The corresponding ROC curve is given in Figure 8 (left). In order to investigate if this high prediction accuracy was due to overfitting, we repeated the analysis by splitting the data into training and test set. On the training data, we fitted the model and optimized the number of iterations via 25-fold bootstrap. We then used the test set to compare the predicted outcome for this new data with the true outcome. The AUC was again 0.981, differing only in the fourth digit. The corresponding ROC curve is given in Figure 8 (right).

Even though the prediction accuracy was essentially equal, we obtained a different prognostic signature. The overlap of DNA variables in the two models was 10, while 19 variables were only selected on the full dataset and 12 were only selected on the training data (cf. table in the supplementary material). A reason for this good and similar performance of the DNA signature, although relying partly on different variables, could be the high grade of information included in several of these 2 905 tumour variables that were generated by comparing tumour and non-tumour DNA of the patients (Gravier et al., 2010). The variables which do not overlap between the two signatures are, hence, replaceable without losing accuracy, either because their effect is of minor importance or because other variables incorporate the same information.

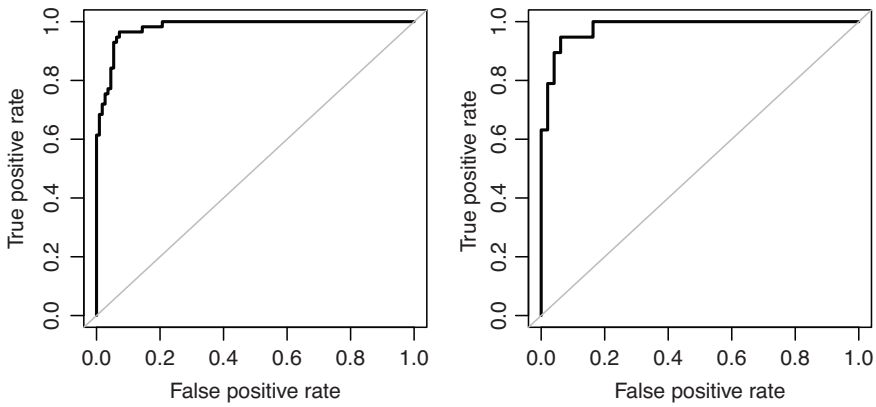

Figure 8 ROC results for the model fitted on the full dataset (left) and the model fitted on a learning set and evaluated on a test set (right)

5 Conclusion

Statistical boosting is a flexible alternative to fit regression models. Although the concept emerged from machine-learning where most algorithmic models must be seen as black-box prediction schemes without any straight-forward interpretation of covariate effects, in this case, the resulting statistical models follow the same structure as if they had been estimated by classical approaches with the same interpretability. In fact, in case of low-dimensional data ($n > p$), boosting models converge (with growing m_{stop}) to the same solution as classical maximum likelihood estimation.

Statistical boosting algorithms lead to several advantages in practice, however, also have some limitations. Advantageous are (a) the intrinsic variable selection properties and shrinkage of effect estimates leading to better prediction accuracy, (b) the robustness towards multicollinearity issues, (c) their flexibility to combine different covariate effects with different regression settings in one unified framework and (d) that they still work for high-dimensional data with more candidate variables than observations ($p > n$). Limitations are (a) the need for model tuning via the stopping iteration, (b) the relatively long run-time (compared to, e.g., LASSO algorithms) particularly for model tuning and most importantly (c) the lack of theoretic results on standard errors for effect estimates. As a result of the last point, we are not able to provide confidence intervals or hypothesis tests for single covariate effects. There exist work-arounds for these issues based on resampling procedures (Hofner et al., 2014a; Mayr et al., 2017b); however, these ad-hoc solutions further increase the computational complexity and run-time.

Putting these advantages and limitations into perspective, it gets clear for which regression settings boosting algorithms are favourable:

- Prediction models for high-dimensional data, particularly when covariate effects should be interpretable.

- General regression settings where variable selection or model choice (e.g., linear versus smooth effects) is necessary.
- Statistical models with multicollinearity problems (high correlation between different covariates).
- Statistical models for regression settings, where other estimation schemes are either not feasible (e.g., C-index) or are not as flexible regarding different types of effects (e.g., spatial effects for quantile regression).

On the other hand, statistical boosting might not be the favourable choice for the following regression settings:

- Statistical models where the focus is primarily on inference for covariate effects, for example, low-dimensional clinical studies where the primary focus is to assess the statistically significant benefit of an intervention.
- General low-dimensional models where no variable selection is needed for classical regression settings (e.g., linear models or GAMs).

In conclusion, statistical boosting can be a very helpful method in the toolbox of a modern statistician when it comes to fitting a regression model. It is not the gold standard but it is nice to know that it is there and it might be very handy in various situations. Due to the different vocabulary, it might sometimes appear rather complicated, but in fact, for most settings it is as simple to apply as classical approaches.

Glossary

Boosting jargon	What does that mean?
base-learner	Underlying regression functions that the algorithm applies iteratively. Typically, each base-learner refers to a single covariate and defines its type of effect, for example, a linear model leads to a linear effect.
loss function	Describes the discrepancy between model and data and is the objective function to be optimized. Defines the regression setting, for example, the L_2 loss leads to classical regression of the mean, L_1 to median regression, the negative log-likelihood leads to the corresponding GLM or GAM.
training/learning data	Data that was used for the estimation of the model, in contrast to *test data* which describes new observations.
out-of-bag observations	Often used term for observations that are part of the test data which was generated via resampling procedures. The opposite are *in-bag observations* that are part of the training data.
empirical risk	Sum over the evaluated loss function on observations that are part of the *training data*.
predictive risk	Sum over the loss on observations that are part of the *test data*, sometimes also called *out-of-bag risk*.
pseudo residuals	Describes the negative gradient vector of the loss on which the base-learners are fitted. In case of the L_2-loss, this is equivalent to fitting the residuals of the previous iteration.
early stopping	Stopping the algorithm before convergence, leads to variable selection and shrinkage, controlled by m_{stop}.

Acknowledgements

The work on this article was supported by the Deutsche Forschungsgemeinschaft (DFG) (www.dfg.de), grant SCHM 2966/1-2, the Interdisciplinary Center for Clinical Research (IZKF) of the Friedrich-Alexander-University Erlangen-Nürnberg (Project J49).

References

Arnold F, Parasuraman S, Arokiasamy P and Kothari M (2009) Nutrition in India: National Family Health Survey (NFHS-3), India, 2005–06 (Technical Report). Mumbai: International Institute for Population Sciences; Calverton, Maryland, USA: ICF Macro.

Bauer A, Scheipl F, Küchenhoff H and Gabriel AA (2018) An introduction to semiparametric function-on-scalar regression. *Statistical Modelling*, **18**, 345–64.

Bender A, Groll A and Scheipl F (2018) A generalized additive model approach to time-to-event analysis. *Statistical Modelling*, **18**, 299–321.

Berger M and Schmid M (2018) Semiparametric regression for discrete time-to-event data. *Statistical Modelling*, **18**, 322–45.

Breiman L (2001) Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science*, **16**, 199–231.

Brockhaus S, Melcher M, Leisch F and Greven S (2017) Boosting flexible functional regression models with a high number of functional historical effects. *Statistics and Computing*, **27**, 913–26.

Bühlmann P and Hothorn T (2007) Boosting algorithms: Regularization, prediction and model fitting (with discussion). *Statistical Science*, **22**, 477–522.

Bühlmann P, Gertheiss J, Hieke S, Kneib T, Ma S, Schumacher M, Tutz G, Wang C-Y, Wang Z and Ziegler A (2014) Discussion of 'The evolution of boosting algorithms' and 'Extending statistical boosting'. *Methods of Information in Medicine*, **53**, 436–45.

de Onis M, Monteiro C, Akre J and Clugston G (1993) The worldwide magnitude of protein-energy malnutrition: An overview from the WHO global database on child growth. *Bulletin of the World Health Organization*, **71**, 703–12.

Fenske N, Burns J, Hothorn T and Rehfuess EA (2013) Understanding child stunting in India: A comprehensive analysis of socio-economic, nutritional and environmental determinants using additive quantile regression. *PloS ONE*, **8**, e78692.

Fenske N, Kneib T and Hothorn T (2011) Identifying risk factors for severe childhood malnutrition by boosting additive quantile regression. *Journal of the American Statistical Association*, **106**, 494–510.

Freund Y (1990) Boosting a weak learning algorithm by majority. In Fulk MA and Case J eds, *Proceedings of the Third Annual Workshop on Computational Learning Theory, COLT 1990, University of Rochester, Rochester, NY, USA, August 6–8, 1990*, pages 202–16.

Gravier E, Pierron G, Vincent-Salomon A, Gruel N, Raynal V, Savignoni A, De Rycke Y, Pierga J-Y, Lucchesi C, Reyal F, Fourquet A, Roman-Roman S, Radvanyi F, Sastre-Garau X, Asselain B, and Delattre O (2010) A prognostic DNA signature for T1T2 node-negative breast cancer patients. *Genes, Chromosomes and Cancer*, **49**, 1125–34.

Harrell FE, Califf RM, Pryor DB, Lee KL and Rosati RA (1982) Evaluating the yield of medical tests. *Journal of the American Medical Association*, **247**, 2543–46.

Harrell FE, Lee KL, and Califf RM, Pryor DB, Rosati RA (1984) Regression modeling

strategies for improved prognostic prediction. *Statistics in Medicine*, **3**, 143–52.

Hepp T, Schmid M, Gefeller O, Waldmann E and Mayr A (2016) Approaches to regularized regression—A comparison between gradient boosting and the lasso. *Methods of Information in Medicine*, **55**, 422–30.

Hofner B, Hothorn T, Kneib T and Schmid M (2011) A framework for unbiased model selection based on boosting. *Journal of Computational and Graphical Statistics*, **20**, 956–71.

Hofner B, Kneib T and Hothorn T (2014a) A unified framework of constrained regression. *Statistics and Computing*, **26**, 1–14. doi:10.1007/s11222-014-9520-y

Hofner B, Mayr A, Robinzonov N and Schmid M (2014b) Model-based boosting in R: A hands-on tutorial using the R package mboost. *Computational Statistics*, **29**, 3–35. doi:/10.1007/s00180-012-0382-5

Hofner B, Mayr A and Schmid M (2016) gamboostLSS: An R package for model building and variable selection in the GAMLSS framework. *Journal of Statistical Software*, **74**, 1–31. doi:10.18637/jss.v074.i01

Hothorn T (2018) Top-down transformation choice. *Statistical Modelling*, **18**, 274–98.

Hothorn T, Bühlmann P, Kneib T, Schmid M and Hofner B (2016) *mboost: Model-Based Boosting*. R package version 2.8-0. URL https://CRAN.R-project.org/package=mboost

Kneib T, Hothorn T and Tutz G (2009) Variable selection and model choice in geoadditive regression models. *Biometrics*, **65**, 626–34.

Koenker R, Ng P and Portnoy S (1994) Quantile smoothing splines. *Biometrika*, **81**, 673–80.

Mayr A, Binder H, Gefeller O and Schmid M (2014a) The evolution of boosting algorithms. *Methods of Information in Medicine*, **53**, 419–27.

——— (2014b) Extending statistical boosting. *Methods of Information in Medicine*, **53**, 428–35.

Mayr A, Fenske N, Hofner B, Kneib T and Schmid M (2012a) Generalized additive models for location, scale and shape for high-dimensional data—A flexible aproach based on boosting. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, **61**, 403–27.

Mayr A, Hofner B and Schmid M (2012b) The importance of knowing when to stop—A sequential stopping rule for component-wise gradient boosting. *Methods of Information in Medicine*, **51**, 178–86.

Mayr A, Hofner B and Schmid M (2016) Boosting the discriminatory power of sparse survival models via optimization of the concordance index and stability selection. *BMC Bioinformatics*, **17**, 288.

Mayr A, Hofner B, Waldmann E, Hepp T, Meyer S and Gefeller O (2017a) An update on statistical boosting in Biomedicine. *Computational and Mathematical Methods in Medicine*. doi:10.1155/2017/6083072

Mayr A, Hothorn T and Fenske N (2012c) Prediction intervals for future BMI values of individual children—A non-parametric approach by quantile boosting. *BMC Medical Research Methodology*, **12**. doi:10.1186/1471-2288-12-6

Mayr A and Schmid M (2014) Boosting the concordance index for survival data—A unified framework to derive and evaluate biomarker combinations. *PloS ONE*, **9**, e84483.

Mayr A, Schmid M, Pfahlberg A, Uter W and Gefeller O (2017b) A permutation test to analyse systematic bias and random measurement errors of medical devices via boosting location and scale models. *Statistical Methods in Medical Research*, **26**, 1443–60.

R Development Core Team (2016) *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0. URL https://www.R-project.org

Ramey JA (2016) *datamicroarray: Collection of data sets for classification*. URL https://github.com/boost-R/datamicroarray

Rigby RA and Stasinopoulos D (2005) Generalized additive models for location, scale and shape (with discussion). *Applied Statistics*, **54**, 507–54.

Schmid M and Hothorn T (2008) Boosting additive models using component-wise P-splines. *Computational Statistics & Data Analysis*, **53**, 298–311.

Sobotka F and Kneib T (2012) Geoadditive expectile regression. *Computational Statistics and Data Analysis*, **56**, 755–67. doi:10.1016/j.csda.2010.11.015

Stasinopoulos M, Rigby RA and de Bastiani F (2018) A distributional regression approach using GAMLSS. *Statistical Modelling*, **18**, 248–73.

Tibshirani R (1996) Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society - Series B*, **58**, 267–88.

Tutz G and Binder H (2006) Generalized additive modeling with implicit variable selection by likelihood-based boosting. *Biometrics*, **62**, 961–71.

Umlauf N and Kneib T (2018) A primer on Bayesian distributional regression. *Statistical Modelling*, **18**, 219–47.

Waldmann E (2018) Quantile regression—A short story on the how and why. *Statistical Modelling*, **18**, 203–18.

Waldmann E, Taylor-Robinson D, Klein N, Kneib T, Pressler T, Schmid M and Mayr A (2017) Boosting joint models for longitudinal and time-to-event data. *Biometrical Journal*. doi:10.1002/bimj.201600158