LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

INSTITUT FÜR STATISTIK

Andreas Mayr, Nora Fenske, Benjamin Hofner, Thomas Kneib, Matthias Schmid

GAMLSS for high-dimensional data – a flexible approach based on boosting

Technical Report Number 098, 2010
Department of Statistics
University of Munich

http://www.stat.uni-muenchen.de

GAMLSS for high-dimensional data – a flexible approach based on boosting

Andreas Mayr*[1], Nora Fenske[2], Benjamin Hofner[1], Thomas Kneib[3], Matthias Schmid[1]

[1] Institut für Medizininformatik, Biometrie und Epidemiologie, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
[2] Institut für Statistik, Ludwig-Maximilians-Universität München, Germany
[3] Institut für Mathematik, Carl von Ossietzky Universität Oldenburg, Germany

Abstract

Generalized additive models for location, scale and shape (GAMLSS) are a popular semi-parametric modelling approach that, in contrast to conventional GAMs, regress not only the expected mean but every distribution parameter (e.g. location, scale and shape) to a set of covariates. Current fitting procedures for GAMLSS are infeasible for high-dimensional data setups and require variable selection based on (potentially problematic) information criteria. The present work describes a boosting algorithm for high-dimensional GAMLSS that was developed to overcome these limitations. Specifically, the new algorithm was designed to allow the simultaneous estimation of predictor effects and variable selection. The proposed algorithm was applied to data of the Munich Rental Guide, which is used by landlords and tenants as a reference for the average rent of a flat depending on its characteristics and spatial features. The net-rent predictions that resulted from the high-dimensional GAMLSS were found to be highly competitive while covariate-specific prediction intervals showed a major improvement over classical GAMs.

Keywords: GAMLSS, high-dimensional data, gradient boosting, variable selection, prediction inference, spatial information.

1 Introduction

Generalized additive models for location, scale and shape (GAMLSS) were introduced by Rigby and Stasinopoulos (2005) as a class of statistical models for regression problems with univariate response. GAMLSS can be seen as a flexible alternative to generalized additive models (GAMs, Hastie and Tibshirani, 1990) as they extend the traditional GAM framework

* *Address for correspondence:* Andreas Mayr, Institut für Medizininformatik, Biometrie und Epidemiologie, Friedrich-Alexander Universität Erlangen-Nürnberg, Waldstr. 6, 91054 Erlangen.
Email: andreas.mayr@imbe.med.uni-erlangen.de

by a variety of modelling options. For example, GAMLSS do not require the conditional distribution of the response variable, given a set of covariates, to be a member of the exponential family; instead, a wide variety of discrete, continuous and mixed discrete-continuous distributions is possible, including distributions based on Box-Cox transformations (such as the Box-Cox t-distribution, Rigby and Stasinopoulos, 2004, or the Box-Cox power exponential distribution, Rigby and Stasinopoulos, 2006) and zero adjusted-distributions (such as the zero-adjusted inverse Gaussian distribution, which is useful for insurance data, see Heller et al., 2006). A comprehensive list of optional distributions for GAMLSS is given in Stasinopoulos and Rigby (2007).

Another key feature of GAMLSS is that every parameter of the conditional response distribution is modelled by its own predictor and an associated link function. While traditional GAMs are typically restricted to modelling the conditional *mean* of the response variable (treating other distributional parameters as fixed), the GAMLSS approach allows for the regression of each distribution parameter on the covariates. Common distribution parameters are location, scale, skewness and kurtosis but degrees of freedom (of a t-distribution) and zero inflation probabilities can be modelled as well. Thus, in the GAMLSS approach, the full conditional distribution of a multi-parameter model is related to a set of predictor variables of interest.

In the same way as traditional GAMs, in GAMLSS the structure of each predictor is assumed to be additive, so that a wide variety of functional terms can be included in each predictor. Examples include non-parametric terms based on penalized splines, varying coefficient terms, spatial and subject-specific terms for repeated measurements. The estimation of GAMLSS coefficients is usually based on penalized likelihood maximization; for details on fitting procedures see Rigby and Stasinopoulos (2005).

In practical applications, GAMLSS have proved to be a convenient option when the response variable does not follow a distribution from the exponential family or when the shape of the response's distribution explicitly depends on covariates. Over the last several years, GAMLSS have been applied to many different areas, ranging from normalising cDNA microarray data (Khondoker et al., 2009) to the analysis of flood frequencies (Villarini et al., 2009), long-term rainfall data (Villarini et al., 2010), and the health impact of temperatures in dwellings (Rudge and Gilchrist, 2007). Clinical applications include long-term survival models for clinical studies (de Castro et al., 2010), while Beyerlein et al. (2008) and Fenske et al. (2008) used GAMLSS to investigate childhood obesity, in an approach closely related to another typical GAMLSS application: the construction of reference charts for child growth curves (see for example Cole et al., 2009).

In this paper, we address the problem of *variable selection*, i.e. the selection of a reasonably small subset of informative covariates to be included in a particular GAMLSS. The selection of informative covariates plays a key role in many practical applications and is often required in applications with high-dimensional data, i.e. data sets with a potentially large number of covariates.

Clearly, even in the traditional GAM setting, variable selection is a complicated issue – one that has been discussed extensively in the literature. With GAMLSS, problems related to variable selection become even more serious, as not only the location parameter (usually corresponding to the conditional mean) but also the scale and shape as well as other parameters of the response distribution are associated with a set of predictor variables. The high degree of flexibility offered by GAMLSS obviously implies that efficient strategies for variable

selection are needed to avoid overfitting of the data and to produce sparse models containing only the most relevant covariates for each distribution parameter. Rigby and Stasinopoulos (2005) proposed using the Generalized Akaike Information Criteria (GAIC) for variable selection in GAMLSS. This approach, however, has several shortcomings that are partially inherited from problems associated with the traditional AIC criterion (Ripley, 2004; Greven and Kneib, 2010, see Section 2.2 for a detailed discussion). In the traditional framework for GAMLSS estimation, it is impossible to avoid these shortcomings, especially if estimation is based on data with a large number of covariates. In these high-dimensional settings, variable selection procedures usually *must* be incorporated. In particular, the GAMLSS fitting procedures proposed by Rigby and Stasinopoulos (2005) are infeasible when there are more covariates than observations.

To address these issues, we developed and subsequently applied a boosting technique (denoted *gamboostLSS* in the following) for estimating and selecting the predictor effects in GAMLSS. Our algorithm is based on the classical gradient boosting approach that originated in the machine-learning field and has been successfully adapted to fit general types of GAMs (Bühlmann and Hothorn, 2007; Kneib et al., 2009). Making use of a recently suggested boosting algorithm for multi-dimensional predictor effects (Schmid et al., 2010), we present a method that can be used to adapt the classical boosting framework to the characteristics of GAMLSS. In addition, we exploit a key feature of classical gradient boosting: As shown by Bühlmann and Yu (2003), classical gradient boosting algorithms not only result in GAM fits but also can be modified to include an intrinsic mechanism for variable selection (component-wise gradient boosting). This approach can be fully integrated into the new gamboostLSS algorithm, producing a sparse solution with respect to all GAMLSS parameters (i.e. predictors for shape, scale, etc.). Consequently, gamboostLSS becomes an efficient technique to simultaneously estimate and select predictor effects in the GAMLSS framework, especially in settings involving high-dimensional data.

The GAMLSS application that motivated the development of the algorithm and which is considered in Section 4 of this paper, is the 2007 Munich Rental Guide, an official reference to determine and assess the net rent per square metre of flats in the German city of Munich (see also Kneib et al., 2010). We applied gamboostLSS to model and select the predictor effects of nearly 250 covariates describing flats in terms of their size, age and other characteristics related to the net rent per square metre. Also included was spatial information, such as the neighbourhood the flat is located in. For this high-dimensional data set, the usual GAMLSS fitting procedures were problematic with respect to variable selection due to the large number of covariates. Yet in order to include spatial information a new algorithm was needed, as with current fitting procedures inclusion was not possible. We show that GAMLSS can compete with traditional mean regression methods for this high-dimensional data set in terms of prediction accuracy for the net rent per square metre. At the same time, gamboostLSS can be adapted to compute covariate-specific prediction intervals, taking into account the effects of both the flat's characteristics and its spatial information on the shape and scale of the conditional response-distribution and therefore also on the size of these intervals. This cannot be accomplished by common modelling strategies that depend on a normally distributed response, as they implicitly assume homoscedasticity and yield equally sized intervals regardless of how expensive or cheap the flat is – clearly contradicting practical experience.

We show that there is a substantial benefit in the GAMLSS approach when not only the

expected mean but also the scale parameter and degrees of freedom of a three-parametric t-distribution are regressed to the covariates. The size of the resulting intervals is no longer fixed but adjusts flexibly to the given covariate values.
The paper is organised as follows. Section 2 starts with a detailed description of the proposed gamboostLSS algorithm and its characteristics. We then discuss classical approaches to variable selection for GAMLSS and compare them to the selection mechanisms incorporated in gamboostLSS. Section 3 contains the results of a simulation study using high-dimensional data with few informative predictor variables but a large number of non-informative covariates. We show that gamboostLSS is an efficient strategy to separate noise from information, i.e. to include only the informative predictor variables in the GAMLSS. In Section 4, we apply the new gamboostLSS algorithm to analyse the 2007 Munich Rental Guide. In addition to point predictions, gamboostLSS can be used to fit covariate-specific prediction intervals, for example for the net rent per square metre, as demonstrated here. A summary of gamboostLSS and its applications, as discussed herein, as well as further aspects regarding the gamboostLSS approach, are given in Section 5. This section also briefly describes the implementation of gamboostLSS, which is based on the R software for statistical computing (R Development Core Team, 2009). The implementation is available with the R add-on package **gamboostLSS** (Hofner et al., 2010).

2 Boosting GAMLSS

2.1 GAMLSS

Rigby and Stasinopoulos (2005) refer to GAMLSS as *semi-parametric* regression type models. While the term *parametric* refers to the fact that the response variable is assumed to follow a parametric distribution, these models are also *non-parametric* because modelling of the relation between covariates and the response may include non-linear effects. The model class assumes observations y_i for $i = 1, 2, ..., n$ that are conditionally independent given a set of covariates. The conditional density $f_{\text{dens}}(y_i|\boldsymbol{\theta}_i)$ may depend on up to four distribution parameters $\boldsymbol{\theta}_i = (\theta_{i1}, \theta_{i2}, \theta_{i3}, \theta_{i4})^\top$. These parameters are commonly referred to as location ("$\theta_{i1} = \mu_i$"), scale ("$\theta_{i2} = \sigma_i$"), skewness ("$\theta_{i3} = \nu_i$") and kurtosis ("$\theta_{i4} = \tau_i$"), although $\boldsymbol{\theta}$ may include any kind of distribution parameter. Each distribution parameter θ_k is modelled by its own additive predictor η_{θ_k} for $k = 1, ..., 4$ and depends additively on the covariates, including possible smooth predictor effects. Let $g_k(\cdot)$ be the known monotonic link functions for each predictor and $x_{k1}, ..., x_{kp_k}$ the p_k covariates in the submodel of parameter θ_k. Note that we allow each of the parameters θ_k to depend on possibly different sets of covariates. A GAMLSS is given by the set of equations

$$g_k(\theta_k) = \beta_{0\theta_k} + \sum_{j=1}^{p_k} f_{j\theta_k}(x_{kj}) = \eta_{\theta_k}, \quad k = 1, \ldots, 4, \tag{1}$$

where $\beta_{0\theta_k}$, $k = 1, \ldots, 4$ are the intercept values of the four submodels. The function $f_{j\theta_k}$ for $j = 1, ..., p_k$ represents the type of effect the covariate j has on the distribution parameter θ_k. As examples of $f_{j\theta_k}$, one can consider a classical linear effect $f_{\text{linear}}(x_{kj}) = x_{kj}\beta_{kj}$ or a smooth effect $f_{kj}(x_{kj}) = f_{\text{smooth}}(x_{kj})$ represented by regression splines. In addition, spatial

(represented by tensor product regression splines) or random effects can contribute to the additive predictors. Clearly, a GAMLSS reduces to a conventional GAM when the model family under consideration includes the location parameter $\theta_{i1} = \mu_i$ as the only distribution parameter to be regressed on the covariates.
The unknown quantities of a GAMLSS can be estimated by maximizing the log-likelihood

$$l = \sum_{i=1}^{n} \log \left[f_{\text{dens}}(y_i|\boldsymbol{\theta}_i) \right] = \sum_{i=1}^{n} \log \left[f_{\text{dens}}(y_i|\mu_i, \sigma_i, \nu_i, \tau_i) \right], \tag{2}$$

with respect to the distribution parameters $\boldsymbol{\theta}_i$. Estimates of the components of $\boldsymbol{\theta}_i$ are then obtained from back-transforming the estimates of the prediction functions (denoted by $\hat{\eta}_{\theta_{ik}}$, $k = 1, \ldots, 4$) via the inverse link functions:

$$\hat{\mu}_i = g_1^{-1}(\hat{\eta}_{\theta_{i1}}), \quad \hat{\sigma}_i = g_2^{-1}(\hat{\eta}_{\theta_{i2}}), \quad \hat{\nu}_i = g_3^{-1}(\hat{\eta}_{\theta_{i3}}), \quad \hat{\tau}_i = g_4^{-1}(\hat{\eta}_{\theta_{i4}}) \, . \tag{3}$$

To estimate the predictor functions in η_{θ_k}, Rigby and Stasinopoulos (2005) introduced a penalized likelihood approach based on modified versions of the back-fitting algorithm for conventional GAM estimation. They proposed two algorithms to obtain GAMLSS estimates, implemented in the R package `gamlss` (Stasinopoulos and Rigby, 2007). Both follow the same basic principle: in each iteration, back-fitting steps are successively applied to the four distribution parameters, with the submodel fits of previous iterations used as offset values for those parameters not involved in the current back-fitting step. For details on the two algorithms, see Stasinopoulos and Rigby (2007).

2.2 AIC-based variable selection

Rigby and Stasinopoulos (2005) discuss a variety of strategies to select relevant predictors and covariate effects in GAMLSS. Specifically, they propose a generalized version of the AIC, defined as

$$\text{GAIC}(a) = -2 \cdot \sum_{i=1}^{n} \log \left[f_{\text{dens}}(y_i|\hat{\boldsymbol{\theta}}_i) \right] + a \cdot \text{df}.$$

The GAIC consists of the negative log-likelihood and a fixed penalty factor a multiplied by the total effective degrees of freedom (df). Note that $a = 2$ or $a = \log(n)$ leads to the classical AIC or Bayesian information criterion (BIC) respectively. Despite being a convenient strategy, GAIC-based variable selection has several shortcomings:
First, variable selection based on information criteria such as the AIC and BIC has generally been criticized as having a large variance, i.e. as being highly instable with respect to the set of predictor variables included in the 'optimal' statistical model (see for example Rawlings et al., 1998). Second, information criteria often result in the inclusion a large number of non-informative predictor variables; that is, they tend to include too many predictors in the optimal model (Ripley, 2004). Also, these criteria may show a substantial bias if used to distinguish between modelling alternatives, for example linear vs. non-linear effects (Greven and Kneib, 2010).

A particular problem associated with the GAIC is the choice of the penalty parameter a. For $a = 2$, the criterion minimizes the Kullback Leibler discrepancy towards the optimal model. Rigby and Stasinopoulos (2005) suggest setting a between 2 and 4 but this choice seems hard to justify theoretically.
Finally, even with a moderate number of potential covariates, the number of candidate GAMLSS can become very large. Consequently, if the aim is to model high-dimensional data with a large number of predictor variables, as in our Munich Rental Guide application, then GAIC-based variable selection becomes computationally almost impossible. The gamboostLSS algorithm introduced in the next sections avoids these problems because it does not rely on the GAIC approach for variable selection; rather, it provides a strategy to estimate the GAMLSS prediction functions while simultaneously selecting appropriate sets of predictor variables.

2.3 Functional gradient descent

The general idea behind boosting algorithms is to consider an ensemble of different 'weak' statistical models that yield predictions for the response variable. These models, called *base-learners*, are subsequently combined to form an overall prediction of the response that is more accurate than any of the predictions obtained from a single base-learner alone (hence the term 'weak' base-learner). Generally, a base-learner can be any kind of statistical tool that fits into the regression framework, i.e. it must result in a prediction of the response variable that is based on the information contained in one or more covariates:

$$\text{covariate(s)} \xrightarrow{\text{base learner}} \text{prediction of the response}$$

Typical examples of base-learners are classification and regression trees, linear models or penalized regression splines (see Section 2.6 for examples).
Boosting was first introduced in the machine-learning field as an algorithm for the classification of binary outcomes (AdaBoost, see Freund and Schapire, 1996). Later it was shown that boosting can be interpreted as a gradient descent algorithm in function space (gradient boosting, Friedman, 2001) that is directly linked to forward stage-wise additive modelling (Friedman et al., 2000; Friedman, 2001; Bühlmann and Yu, 2003). Consequently, boosting can be used as a technique for fitting generalized additive regression models whose prediction function is determined by specification of the set of base-learners. An overview of state-of-the-art boosting algorithms can be found in Bühlmann and Hothorn (2007).
Here, we consider the gradient boosting approach introduced by Friedman (2001). The task is to derive a prediction η by minimizing the expectation of a loss function $\rho(\cdot)$ assumed to be differentiable with respect to η:

$$\hat{\eta} = \underset{\eta}{\operatorname{argmin}}\ \mathbb{E}_{Y,X}\left[\rho(Y, \eta(X))\right] \ ,$$

where Y and X are the random variables for response and covariate(s) respectively. In practice, for a sample of observations $(y_1, x_1), ..., (y_n, x_n)$, the algorithm minimizes the empirical risk

$$\frac{1}{n}\sum_{i=1}^{n} \rho(y_i, \eta(x_i))$$

with respect to η by a stepwise descent of the loss-function's gradient. Instead of fitting the original data points, for iterations $m = 1, ..., m_{\text{stop}}$ the boosting algorithm iteratively fits the gradient of the loss function to the covariate. In every step, the current version of η is updated additively by a step-length (denoted 'sl') in order to approximate a minimum.

The functional gradient descent algorithm is formally given as follows:

1. Initialize $\hat{\eta}^{[0]} = 0$.
2. Increase m by 1. Compute the negative gradient and evaluate at the current estimate:

$$u_i = -\frac{\partial}{\partial \eta}\rho(y_i, \eta_i)\Big|_{\eta_i = \hat{\eta}^{[m-1]}(x_i)}$$

3. Fit the negative gradient with the base-learners:

$$(x_i, u_i)_{i=1}^n \overset{\text{base learner}}{\longrightarrow} \hat{h}^{[m]}(\cdot)$$

4. Update the prediction function with a step-length $0 < \text{sl} \leq 1$:

$$\hat{\eta}^{[m]} = \hat{\eta}^{[m-1]} + \text{sl} \cdot \hat{h}^{[m]}(\cdot)$$

5. Iterate steps 2 – 4 until the stopping iteration m_{stop} is reached.

The additive structure of the resulting model fit is a direct effect of the gradient descent algorithm, as the final aggregation of the base-learners is strictly additive; in every iteration, small increments are added to the current prediction function $\hat{\eta}$. This is also the link between gradient descent boosting and stagewise additive modelling as provided by the LARS algorithm (see Efron et al., 2004).
For multi-dimensional X, the algorithm can be adapted to fit the covariates *component-wise*: For each base-learner, one component of X is fit to the gradient vector, and in each boosting step the algorithm updates only the component with the best-performing base-learner (Bühlmann and Yu, 2003). The main advantages of this strategy emerge when a small stopping iteration m_{stop} is chosen ('early stopping'): First, the algorithm includes a data-driven mechanism for variable selection, as only the best-performing covariate is updated in each boosting step. By stopping the algorithm early, less important covariates are not updated and are therefore effectively excluded from the final model. Second, the predictor functions of those covariates included in the model are shrunk towards zero, in part also due to the step-length $\text{sl} < 1$. Shrinkage of the effect estimates leads to a lower variance and therefore to more stable predictions (see Efron, 1975; Copas, 1983; Hastie et al., 2009). Furthermore, component-wise boosting allows the estimation of a greater number of effect coefficients than observations. As only one base-learner is fitted at a time, the curse of dimensionality becomes almost irrelevant for the estimation procedure. Also, problems with multi-collinearity, which arise often in high-dimensional data, do not have a negative effect on the estimation accuracy.
An implementation of gradient descent algorithms for a multitude of statistical modelling options is available with the R add-on package **mboost** (Hothorn et al., 2010a,b).

2.4 The gamboostLSS algorithm

With the gamboostLSS algorithm, we propose a component-wise gradient descent algorithm that rotates between the different prediction functions of the distribution parameters for GAMLSS. Analogous to the classical gradient descent algorithm presented in the previous subsection, gamboostLSS can handle high-dimensional data settings ($p > n$) and includes intrinsic variable selection.

To extend the classical gradient boosting approach to the GAMLSS framework, we adopted a strategy recently proposed by Schmid et al. (2010): In each iteration, gamboostLSS calculates the negative partial derivatives of the negative log-likelihood function of a GAMLSS with respect to each of the four predictors η_{θ_k}, $k = 1, \ldots, 4$. These four predictors are updated successively in each iteration, in which the current estimates of the other distribution parameters are used as offset values. A schematic overview of the updating process of gamboostLSS in iteration $m + 1$ is as follows:

$$
\begin{aligned}
(\hat{\mu}^{[m]}, \hat{\sigma}^{[m]}, \hat{\nu}^{[m]}, \hat{\tau}^{[m]}) &\xrightarrow{\text{update}} \hat{\eta}_{\mu}^{[m+1]} \longrightarrow \hat{\mu}^{[m+1]} \,, \\
(\hat{\mu}^{[m+1]}, \hat{\sigma}^{[m]}, \hat{\nu}^{[m]}, \hat{\tau}^{[m]}) &\xrightarrow{\text{update}} \hat{\eta}_{\sigma}^{[m+1]} \longrightarrow \hat{\sigma}^{[m+1]} \,, \\
(\hat{\mu}^{[m+1]}, \hat{\sigma}^{[m+1]}, \hat{\nu}^{[m]}, \hat{\tau}^{[m]}) &\xrightarrow{\text{update}} \hat{\eta}_{\nu}^{[m+1]} \longrightarrow \hat{\nu}^{[m+1]} \,, \\
(\hat{\mu}^{[m+1]}, \hat{\sigma}^{[m+1]}, \hat{\nu}^{[m+1]}, \hat{\tau}^{[m]}) &\xrightarrow{\text{update}} \hat{\eta}_{\tau}^{[m+1]} \longrightarrow \hat{\tau}^{[m+1]} \,.
\end{aligned}
$$

The prediction functions are updated for each additive predictor η_{θ_k} until the stopping iteration m_{stop} is reached. In some settings, it may be additionally convenient to allow m_{stop} to differ between the distribution parameters. $\boldsymbol{m}_{\text{stop}} = (m_{\text{stop},1}, ..., m_{\text{stop},4})^{\top}$ is therefore a vector of tuning parameters that can, for example, be determined using cross-validation (see Subsection 2.5 for details).

In the case of GAMLSS, the *component-wise* base-learning strategy presented above can be naturally extended. Since there is not only one (as in classical GAMs) but a set of up to four distribution parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_4)^{\top}$, each distribution parameter of a GAMLSS has its separate additive predictor η_{θ_k} that is updated component-wise by gamboostLSS. For example, if $\boldsymbol{\theta}$ is of length $K = 4$, we specify four sets of base-learners, with each set used to update one of the four additive predictors. Thus, in each iteration and for each distribution parameter, the base-learner that best fits the respective negative partial derivative is used to update the prediction function under consideration. A direct consequence of this strategy is that each of the prediction functions may depend at the final iteration on a different set of covariates, leading to variable selection in each predictor. In principle, any type of base-learner that can be used in classical gradient boosting can also be specified for the prediction functions in gamboostLSS (see Subsection 2.6 for an in-depth discussion).

A formal definition of gamboostLSS is as follows. Since the task is to model the distribution parameters of the conditional density $f_{\text{dens}}(y|\mu, \sigma, \nu, \tau)$, the optimization problem for gamboostLSS can be formulated as

$$
(\hat{\mu}, \hat{\sigma}, \hat{\nu}, \hat{\tau}) = \underset{\eta_\mu, \eta_\sigma, \eta_\nu, \eta_\tau}{\operatorname{argmin}} \, \mathbb{E}_{Y,X} \left[\rho(Y, \eta_\mu(X), \eta_\sigma(X), \eta_\nu(X), \eta_\tau(X)) \right] \tag{4}
$$

with $\rho = -l$ the negative log-likelihood of the response distribution and (Y, X) the random variables for the response or the covariates respectively. Given that the theoretical

expectation is, in practice, unknown, we follow the classical gradient boosting approach and minimize the empirical risk

$$\frac{1}{n}\sum_{i=1}^{n}\rho(y_i,\eta_{\mu_i},\eta_{\sigma_i},\eta_{\nu_i},\eta_{\tau_i}) = \frac{1}{n}\sum_{i=1}^{n}\rho(y_i,\boldsymbol{\eta}_i)$$

over $\boldsymbol{\eta}_i = (\eta_{\mu_i},\eta_{\sigma_i},\eta_{\nu_i},\eta_{\tau_i})^\top$, with $\boldsymbol{y} = (y_1,...,y_n)^\top$ denoting the response vector with observations being conditionally independent given a set of covariates. In each iteration of gamboostLSS, pre-specified sets of base-learners are used to fit the negative partial derivatives of the empirical risk (with respect to the elements of $\boldsymbol{\eta}_i$) as evaluated at the current prediction.
These considerations lead to the following gradient boosting algorithm for fitting GAMLSS (gamboostLSS):

1. Initialize the additive predictors $\hat{\eta}_{\mu i}^{[0]}, \hat{\eta}_{\sigma i}^{[0]}, \hat{\eta}_{\nu i}^{[0]}, \hat{\eta}_{\tau i}^{[0]}$ with offset values, e.g. $\hat{\eta}_{\theta_{ki}}^{[0]} = 0$ for $k = 1,...,4$ and $i = 1,...,n$.

2. For each distribution parameter θ_k, $k = 1,\ldots,4$, specify a set of base-learners, i.e. a set of regression-type estimators (trees, P-splines, etc.) depending on subsets of the covariates. Denote the set of base-learners for distribution parameter θ_k by $h_{k1}(\cdot),\ldots,h_{kp_k}(\cdot)$, $k = 1,\ldots,4$, where p_k is the cardinality of the set of base-learners specified for θ_k. The base-learners may be the same for each θ_k but may also differ.

   Set the iteration counter $m = 0$.

3. Increase m by 1.

4. (a) Set $k = 0$.

   (b) Increase k by 1. If $m > m_{\text{stop},k}$ proceed to step 4(f).
   Else compute the negative partial derivative $-\frac{\partial}{\partial\eta_{\theta_k}}\rho(y_i,\boldsymbol{\eta}_i)$ and plug in the current estimates $\boldsymbol{\eta}_i = (\hat{\eta}_{\mu_i}^{[m-1]}, \hat{\eta}_{\sigma_i}^{[m-1]}, \hat{\eta}_{\nu_i}^{[m-1]}, \hat{\eta}_{\tau_i}^{[m-1]})$.
   This yields the vector of partial derivatives

$$\boldsymbol{u}_k^{[m-1]} = \left(-\frac{\partial}{\partial\eta_{\theta_k}}\rho(y_i,\boldsymbol{\eta}_i)\right)_{i=1,\ldots,n} .$$

   (c) Fit the negative gradient vector $\boldsymbol{u}_k^{[m-1]}$ to each of the base-learners contained in the set of base-learners specified for the predictor η_{θ_k} in step 2.

   (d) Select the component j^* that best fits the negative partial-derivative vector according to the least-squares criterion, i.e. select the base-learner h_{kj^*} defined by

$$j^* = \underset{1\le j\le p_k}{\operatorname{argmin}}\sum_{i=1}^{n}(u_{ik}^{[m-1]} - h_{kj}(\cdot))^2 .$$

   (e) Update the additive predictor η_{θ_k} as follows:

$$\hat{\eta}_{\theta_k}^{[m-1]} = \hat{\eta}_{\theta_k}^{[m-1]} + \text{sl}\cdot h_{kj^*}(\cdot) ,$$

where sl is a small step-length ($0 < \text{sl} \ll 1$). Therefore, only the best-performing base-learner (and therefore the best-performing covariate) contributes to the update.

(f) Set $\hat{\eta}_{\theta_k}^{[m]} = \hat{\eta}_{\theta_k}^{[m-1]}$.

(g) Repeat steps 4(b) to 4(f) for $k = 2, ..., 4$.

5. Iterate steps 3 and 4 until $m > m_{\text{stop},k}$ for all $k = 1, ..., 4$.

Due to the additive updates in each iteration step ($\hat{\eta}_{\theta_k}^{[m]} = \hat{\eta}_{\theta_k}^{[m-1]} + \text{sl} \cdot h_{kj^*}(\cdot)$), every resulting predictor η_{θ_k} follows an additive structure as in (1). The type of the resulting predictor functions $f_{j\theta_k}$ (effect of covariate j on distribution parameter θ_k) corresponds to the base-learner h_{kj}. The selection of the base-learners is therefore, above all, a decision regarding the structure of the additive model. The base-learner defines the type of effect represented by the function $f_{j\theta_k}$ for covariate component j on parameter θ_k (for possible base-learners see Subsection 2.6).
One of the main characteristics of gamboostLSS is its ability to handle high-dimensional setups in which there are more effects to estimate than observations. Note that these setups are more likely for GAMLSS than for common GAMs, as different models may be fitted not only for the conditional mean but also for other parameters of the response distribution.
For the sake of consistency within classical GAMLSS theory (Rigby and Stasinopoulos, 2005), in the denotation of the algorithm in this section four distribution parameters are always considered. Yet it should be noted that gamboostLSS is able to handle distributions even more complex than those considered by Rigby and Stasinopoulos (2005), as gamboostLSS does not require the number of distribution parameters (denoted as K in the following) to be less than or equal to 4.

2.5 Tuning gamboostLSS

The most important tuning parameter of gamboostLSS is the vector of stopping iterations $\boldsymbol{m}_{\text{stop}}$. Here, $\boldsymbol{m}_{\text{stop}}$ is a K-dimensional vector that defines the stopping iteration for each distribution parameter θ_k, i.e. the iteration after which further update of η_{θ_k} is no longer necessary. By standard gradient descent arguments (see Rosset et al., 2004), for $m_{\text{stop},k} \to \infty \; \forall k$ gamboostLSS converges to the same solution as provided by the classical maximum likelihood estimation (based on the algorithms provided by Stasinopoulos and Rigby, 2007). This result is also supported by simulation studies concerning GAMLSS fitting to low-dimensional data (Section 3). For small(er) stopping iterations (early stopping), the effect estimates produced by gamboostLSS shrink towards zero as the additive updates are stopped before convergence. Shrinkage of the effect estimates has the advantage that predictions become more stable since the variance of the estimates is reduced. This feature is also one of the major advantages of classical gradient boosting (Hastie et al., 2009). Another advantage of early stopping is that gamboostLSS has an intrinsic mechanism for data-driven variable selection, as only the best-fitting covariates are updated in each boosting iteration. Early stopping of the algorithm reduces the chance that less important variables are selected for the distribution parameters. Hence, the stopping iteration $m_{\text{stop},k}$ not only controls the amount of shrinkage applied to the effect estimates but also the complexity of the model for the distribution parameter θ_k.

Another tuning parameter is the step-length (sl) involved in the additive updates. The step-length also contributes to the shrinkage effect and guarantees the stability of gamboostLSS; therefore, sl should be a small positive number ($\ll 1$). In early boosting algorithms the estimation of an 'optimal' value of sl in every iteration was proposed (see Friedman, 2001). However, recent results suggest that this (time-consuming) procedure is of relatively little importance for the prediction accuracy of boosting algorithms as there is a direct dependency between m_{stop} and the step-length (Schmid and Hothorn, 2008). We therefore used a fixed step-length for gamboostLSS (sl is set equal to 0.1, a value commonly used in practical applications) and concentrated on finding an optimal stopping iteration m_{stop}.
For GAMs ($K = 1$) estimated by classical gradient boosting, m_{stop} is usually selected with the help of cross-validation (CV) techniques. To effectively avoid overfitting, it is crucial that boosting algorithms are not run until convergence; they should be stopped considering the predictive risk in a separate test data set (see Bühlmann and Hothorn, 2007). With CV, m_{stop} is optimized by evaluating the predictive empirical risk in each iteration using different folds of learning and test data. The 'optimal' value of m_{stop} is then given by the iteration with smallest predictive risk (averaged over the folds, see Hothorn et al., 2005). In the case of GAMLSS, CV is more complex, as K different stopping iterations can be chosen to allow for different levels of complexity in each sub-model. In the following sections, we distinguish between one-dimensional early stopping ($m_{\text{stop},k} \equiv m_{\text{stop}}$ for $k = 1, ..., K$) and multi-dimensional early stopping in which the elements of $m_{\text{stop},k}$ differ for $k = 1, ..., K$. While the choice of the same stopping iteration for all distribution parameters ($m_{\text{stop},k} \equiv m_{\text{stop}}$ for $k = 1, ..., K$) requires only a one-dimensional CV (and therefore reduces the computational effort), multi-dimensional early stopping provides greater flexibility and more accurate estimation results. With multi-dimensional early stopping, CV is achieved using a K-dimensional grid of stopping iterations, in which the optimal vector of stopping iterations is given by a combination of iterations with the smallest predictive empirical risk. For details on the early stopping techniques used for simulation studies and analysis of the Munich Rental Guide, we refer the reader to Sections 3 and 4.
Other parameters that influence the resulting stopping iteration are the initial values $\hat{\eta}_{\theta_k}^{[0]}$. Offset values such as $\hat{\eta}_{\theta_k}^{[0]} = 0$ are a possible and easy solution, yet they typically result in longer run-times (more iterations needed) than are needed with more 'intelligent' initial values. In the implementation of our algorithm, we used a marginal optimization of the empirical risk with respect to constant offsets ($\hat{\eta}_{\theta_k}^{[0]} = c_k$) for $k = 1, ..., K$.

2.6 Base-learners and distributions

Another characteristic of the presented algorithm is its flexibility with respect to the selection of base-learners and therefore of the type of effect(s) that covariates will have on the predictors of the GAMLSS distribution parameters. Generally, all base-learners available in the classical boosting framework can be used, e.g. those provided by the R add-on package **mboost** (Hothorn et al., 2010a,b). It is important to keep in mind that, due to the additive update ($\hat{\eta}_{\theta_k}^{[m]} = \hat{\eta}_{\theta_k}^{[m-1]} + \text{sl} \cdot h_{kj^*}(\cdot)$) of gamboostLSS, the final boosting estimate of a predictor effect for a particular covariate has the same structure as the base-learner specified for this covariate at the beginning of the gamboostLSS algorithm. For example, the predictor effect of a covariate is linear in this covariate if its base-learner is a simple linear model (see Bühlmann and Hothorn, 2007). Similarly, the predictor effect of a particular covariate is a

smooth non-linear function of this covariate if the corresponding base-learner is a smooth non-linear function as well.
There are several types of base-learners that can be used for gamboostLSS. (i) Linear effects are represented by simple linear models estimated by the classical least-squares method. (ii) Non-linear effects include those considered herein, which are modelled using penalized regression splines (P-splines); that is, a smooth effect of a predictor variable is modelled as a linear combination of B-spline functions on a fixed set of equidistant knots. Additionally, a roughness penalty based on the squared second-order differences of effect coefficients belonging to adjacent basis functions is included (Eilers and Marx, 1996). (iii) Spatial effects can be incorporated into gamboostLSS by setting up a bivariate tensor product extension of penalized B-splines for a two-dimensional continuous variable representing geographic information (Kneib et al., 2009). Consequently, this 'tensor product P-spline' becomes a base-learner relying on two covariates, namely the coordinates of a spatial location on a two-dimensional grid (or map). Another possible base-learner for spatial effects is the adaptation of Markov random fields (MRF) for those effects with a neighbourhood structure. The covariate of the corresponding MRF is therefore given by an indicator specifying both a particular region and information on neighbouring regions (Sobotka and Kneib, 2010). We applied this base-learner to model the spatial structure of the Munich Rental Guide Data (see Section 4). (iv) Random effects are taken into account by modelling subject-specific effects or the categorical grouping variables contained in a data set using random intercepts or slopes for each level or subject. Following the approach of Kneib et al. (2009) (supplementary material), we used ridge-penalized base-learners to incorporate random effects into gamboostLSS.
The possibility to model spatial and random coefficient effects for GAMLSS must be emphasized, since until now this has not been feasible, at least not with the currently available implementation of the classical algorithms provided by Stasinopoulos and Rigby (2007). The proposed gamboostLSS algorithm therefore not only extends the possibilities for fitting GAMLSS to high-dimensional data but also offers greater flexibility for modelling different types of effects in low-dimensional settings.
Rigby and Stasinopoulos (2005) consider a large set of different GAMLSS distributions, all of which can be fitted by the proposed boosting algorithm. In this paper, we apply the negative binomial distribution for count data and the log-logistic distribution for accelerated failure time models in simulation studies (Section 3). For the analysis of the Munich Rental Guide (Section 4), we have applied a three-parametric t-distribution.

3 Simulation study with high-dimensional data

We carried out a simulation study with different data settings including linear and non-linear effects for two different response distributions. In this study, two GAMLSS families were considered: the negative binomial distribution for count data and the log-logistic distribution for accelerated failure time models for time-to-event data. Both settings included high-dimensional data with more covariates than observations ($p > n$). Since most of the covariates were strictly non-informative, appropriate selection of the model's informative predictors was considered crucial.
For the presented settings, it was not possible to compare the results of gamboostLSS with those of the original algorithms by Rigby and Stasinopoulos (2005), as the latter are unable

to estimate more coefficient effects than observations. Yet, in the smaller simulated settings and data sets provided in the R add-on package `gamlss` (Stasinopoulos and Rigby, 2007), we confirmed that in the low-dimensional case our algorithm converged to the results of the original back-fitting procedures (not presented here).

Our simulation study was aimed at answering the following questions:

1. Is the proposed gamboostLSS algorithm able to correctly model the corresponding distribution parameters of the GAMLSS families in high-dimensional settings?
2. Is the algorithm able to identify the small subset of informative covariates?
3. What is the effect of early stopping? Is there a difference if one-dimensional rather than multi-dimensional early stopping is applied?

All calculations and simulations were carried out using the R software for statistical computing (R Development Core Team, 2009). The gamboostLSS implementation applied in this study is available with the R add-on package **gamboostLSS** (Hofner et al., 2010).

3.1 Linear setting

For linear settings, we considered the negative binomial distribution for count data with distribution parameters μ (location) and σ (accounting for overdispersion). With the chosen setting, both parameters are regressed to the covariates such that both location and dispersion depend on the covariates. We simulated $n = 800$ observations arising from the negative binomial distribution with density

$$f_{\text{dens}}(y_i|\mu_i, \sigma_i) = \frac{\Gamma(y_i + \sigma_i)}{\Gamma(y_i + 1)\Gamma(\sigma_i)} \frac{\left(\frac{\mu_i}{\sigma_i}\right)^{y_i}}{\left(\frac{\mu_i}{\sigma_i} + 1\right)^{(y_i+\sigma_i)}},$$

where the underlying additive linear predictors are given by

$$\begin{aligned}
\log(\mu_i) &= \eta_{\mu_i} = 1.5 + 1 \cdot x_{1i} + 0.5 \cdot x_{2i} - 0.5 \cdot x_{3i} - 1 \cdot x_{4i} + \sum_{j=5}^{1000} 0 \cdot x_{ji} \quad , \\
\log(\sigma_i) &= \eta_{\sigma_i} = 0 \cdot (x_{1i} + x_{2i}) - 0.4 \cdot x_{3i} - 0.2 \cdot x_{4i} + 0.2 \cdot x_{5i} + 0.4 \cdot x_{6i} + \sum_{j=7}^{1000} 0 \cdot x_{ji} \ ,
\end{aligned}$$

and where the covariates $\boldsymbol{x}_1, ..., \boldsymbol{x}_{1000}$ are $1 \times n$ vectors of *iid* realizations of random variables $X_1, ..., X_{1000}$ following a multivariate normal distribution with a mean of zero and a standard deviation (sd) equal to 1. The covariates are pairwise correlated with the correlation coefficient $\rho = 0.5$. Thus, 1,000 covariates were included of which only six were informative for any of the distribution parameters (two for both, two only for the location parameter and two only for the dispersion parameter).
Since the predictors are linear, simple linear regression models were used as base-learners in the proposed gamboostLSS algorithm. We considered $2 \times 1,000$ simple linear models as

base-learners for each of the two distribution parameters; hence, one model was used as a base-learner for each covariate and distribution parameter. The step length was fixed as 0.1 and the stopping iteration m_{stop} determined by evaluating the empirical risk on an additional independent *iid* data set with 1,000 observations, following the same distribution as the original data set. Both one- and two-dimensional early stopping were carried out, evaluating a grid of different stopping iterations for μ and σ. The resulting stopping iterations for the two parameters differed: With one-dimensional early stopping the average m_{stop} was 412.4 (sd = 67.9) whereas for two-dimensional stopping the average stopping iteration for $\hat{\eta}_\mu$ (501.9, sd = 96.9) was higher than that for $\hat{\eta}_\sigma$ (419.3, sd = 124.0). The resulting average predictive risk at the 'optimal' two-dimensional stopping iteration (2667.5, sd = 47.9) was only slightly smaller than that for one-dimensional early stopping (2670.2, sd = 47.3) at an average of 412.4 iterations for both parameters. This result suggests that, in this particular simulation setting, a one-dimensional search for the optimal m_{stop} can yield satisfying results. Figure 1 presents the coefficient estimates resulting from the algorithm with multi-dimensional early stopping, allowing for different complexities in the models for μ and σ. The box-plots correspond to the empirical distribution of estimates from 100 independent samples of size $n = 800$, each generated from the negative binomial distribution specified above. The signs of the coefficient estimates and their magnitudes both for μ and for σ clearly reflect the true structures of η_μ and η_σ. As expected, due to the regularization property of the presented algorithm, all coefficient estimates shrunk towards zero. The estimates for all non-informative covariates are presented together in the last box-plot, which shows that the variable selection carried out by gamboostLSS works remarkably well. This view is further supported by the selection rates, i.e. the proportion of simulation runs in which a particular base-learner was chosen at least once before gamboostLSS was stopped: The non-informative variables for the location parameter μ had an average selection rate of 3.5% in the estimation of $\hat{\eta}_\mu$. For the estimation of $\hat{\eta}_\sigma$, the non-informative variables (including X_1 and X_2) were selected in 1.8% of the simulation runs. The average number of variables selected from the 1,000 available covariates was 39.2 (sd = 14.2) for the location model and 20.5 (sd = 7.9) for the scale model, which highlights the ability of gamboostLSS to generate sparse models in high-dimensional data settings.

3.2 Non-linear setting

After evaluating the performance of gamboostLSS in high-dimensional data setups with a linear additive structure, we considered additive predictors including non-linear effects. For those non-linear predictors, we chose the log-logistic distribution for accelerated failure time models as the GAMLSS example. These models are an alternative to Cox proportional hazard models and are a popular choice for modelling survival data parametrically (Klein and Moeschberger, 2003). They are based on the model equation

$$\log(y) = \mu + \sigma \cdot W,$$

where y is the survival time, μ the location and σ the scale parameter. W is the noise variable, which in the case of a log-logistic response follows a standard logistic distribution. We simulated 800 observations following a log-logistic distribution with density

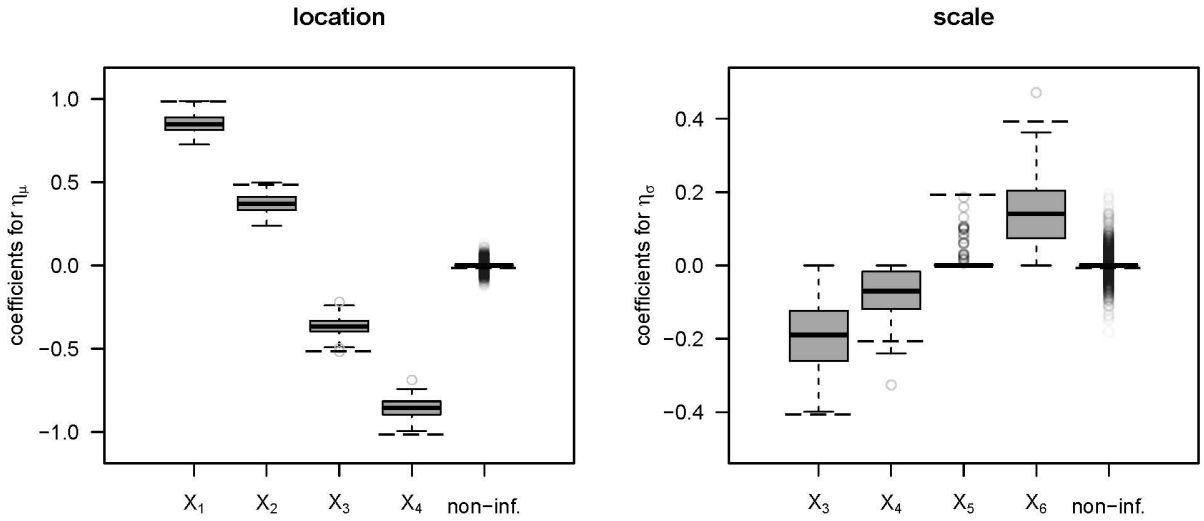

Figure 1: Results from the simulation study, linear setting: Box-plots display the empirical distribution of the estimated coefficients for the location parameter (left) and the scale parameter (right) of the negative binomial distribution, obtained from running gamboostLSS in a high-dimensional setting (100 simulation runs). The dashed lines represent the underlying true coefficients without shrinkage.

$$f_{\text{dens}}(y_i|\mu_i, \sigma_i) = \frac{\exp\left(\frac{y_i-\mu_i}{\sigma_i}\right)}{\sigma_i\left(1+\exp\left(\frac{y_i-\mu_i}{\sigma_i}\right)\right)^2} \ .$$

The underlying additive predictors were specified as follows:

$$\begin{aligned} \mu_i &= \eta_{\mu_i} = 1 + 8 \cdot \sin(x_{1i}) + 3 \cdot \log(x_{2i}) + \sum_{j=3}^{1000} 0 \cdot x_{ji} \quad , \\ \log(\sigma_i) &= \eta_{\sigma_i} = 0 \cdot (x_{1i} + x_{2i}) - 0.8 \cdot (x_{3i}^4 - x_{3i}^3 - 5 \cdot x_{3i}^2) - 3 \cdot x_{4i} + \sum_{j=5}^{1000} 0 \cdot x_{ji}. \end{aligned}$$

All covariates were sampled from uniformly distributed random variables $X_1, ..., X_{1000}$ on a grid from 0 to 3 and were pairwise independent, thus yielding four informative (two for each distribution parameter) and 996 non-informative covariates. In addition to the survival times $\boldsymbol{y}_{\text{surv}}$, we simulated *iid* censoring times $\boldsymbol{y}_{\text{cens}}$ following the same distribution as $\boldsymbol{y}_{\text{surv}}$. Censoring took place when the sampled censoring time was smaller than the survival time. The observed survival times were then given by $y_i = \min(y_{\text{surv}i}, y_{\text{cens}i})$. As a result, about half of the observed survival times were independently right-censored.
As base-learners cubic P-splines (20 equidistant knots with a second-order difference penalty) were used, with four degrees of freedom assigned to each P-spline base-learner. One P-spline base learner was used for each available covariate and for each of the distribution

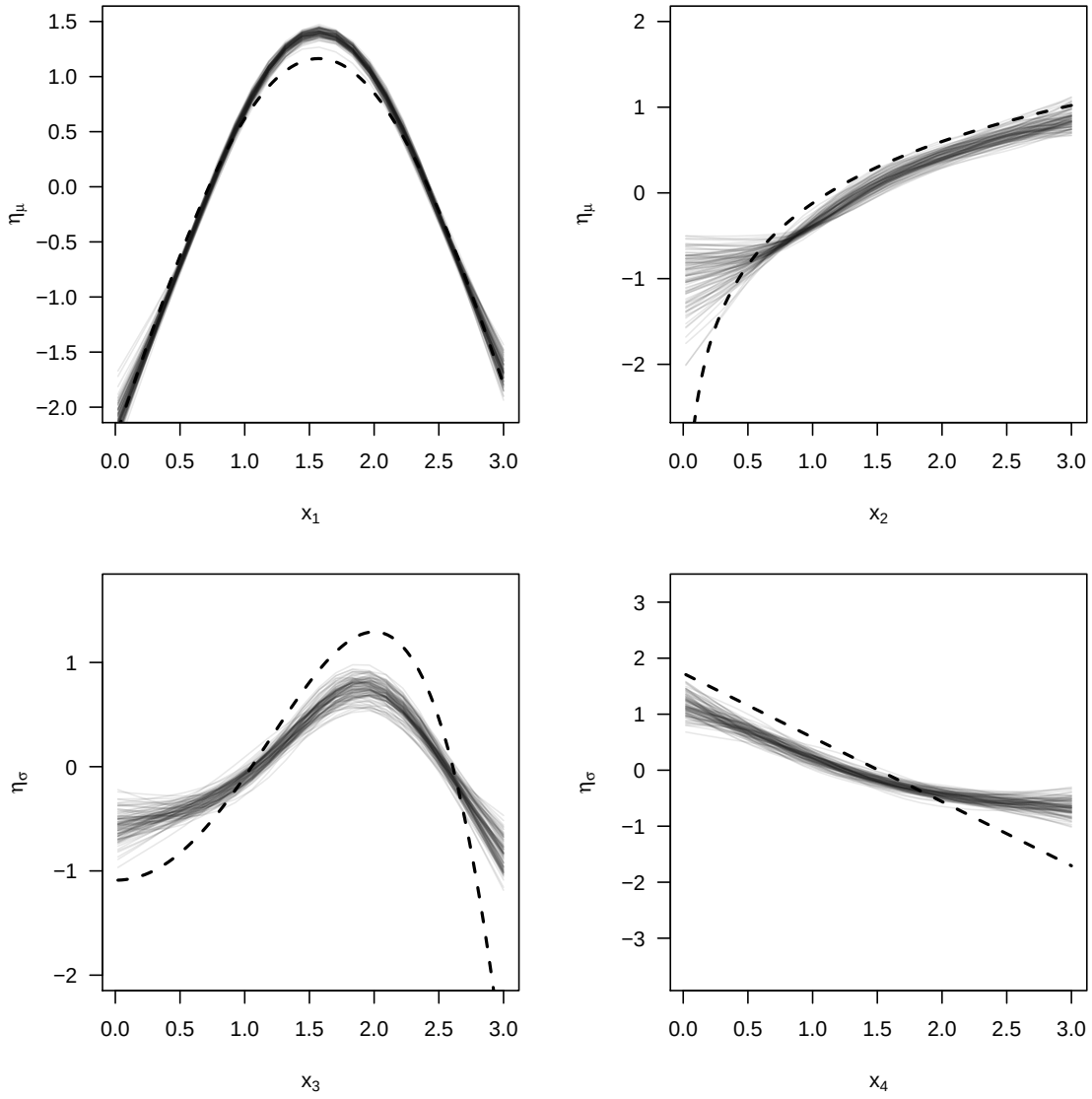

Figure 2: Results from the simulation study, non-linear setting. Solid grey lines display estimated predictor functions for the location parameter (top) and the scale parameter (bottom) of the log-logistic distribution obtained from running gamboostLSS at a high-dimensional setting (100 simulation runs). Dashed lines represent the underlying true functions without shrinkage.

parameters. Hence, the learning algorithm was able to select from 2,000 different base-learners to update the GAMLSS fit. We again performed one- and two-dimensional early stopping, with m_{stop} selected using an additional independent *iid* data set consisting of 1,000 observations following the same distribution as the original data. The average value of m_{stop} obtained from one-dimensional stopping was 113.4 (sd = 8.3). The iterations resulting from the two-dimensional early stopping differed only slightly for $\hat{\eta}_\mu$ (110.1, sd = 13.3) and $\hat{\eta}_\sigma$ (117.1, sd = 9.8). Also, the resulting average predictive risk from two-dimensional early stopping (626.7, sd = 48.2) was only slightly smaller than the empirical predictive risk from the one-dimensional strategy (634.7, sd = 45.4).

Figure 2 presents the effect estimates from the models with two-dimensional early stopping. The resulting function estimates from 100 simulation runs are plotted along with the respective true functions. The effects of $X_1, ..., X_4$ are well-approximated by their corresponding estimates, taking into account that, as in the linear setting, the effect estimates shrink towards zero as a result of the regularization property of gamboostLSS. Only the predictor function of X_1 for $\hat{\eta}_\mu$ seems to be somewhat problematic. In the centre of the X_1 grid, no shrinkage effect is observed, as the estimated functions appear to be larger than the true effect of X_1. This result may be explained by the fact that the slope of the sine function is largest near 0 and π, and that boundary effects may occur in these regions if P-splines are used to approximate the sine function. Since all function estimates in Figure 2 are vertically centred around the zero line, these boundary effects may in turn lead to a 'vertical lift' of the estimates and therefore to a spurious positive bias of estimates near 0.
The informative covariates $X_1, ..., X_4$ were selected in every simulation run (selection rates = 100% for both parameters), while $X_5, ..., X_{1000}$ were selected on average in 1.7% of the simulations for $\hat{\eta}_\mu$ and in 0.12% of the simulations for $\hat{\eta}_\sigma$. These selection rates further emphasize that the intrinsic variable selection carried out by gamboostLSS works remarkably well, providing sparse solutions in high-dimensional settings.

4 Munich Rental Guide

4.1 Data and models

Most larger German cities publish rental guides as a reference on 'average rents' for both landlords and tenants. These guides offer point predictions for the net rent based on a flat's characteristics together with spans (or prediction intervals) indicating the range of usual rents. Although earlier rental guides were tabular-based, nowadays most are derived from regression models with a flat's characteristics as covariates and the net rent or net rent per square metre as response variable.
In this section, we use GAMLSS to analyse data collected for the 2007 rental guide for the German city of Munich. The main objective of the analysis is to obtain point predictions for the net rent per square metre and to construct prediction intervals holding a pre-specified coverage probability for this variable. Our sample comprises data obtained from $n = 3016$ flats within the city of Munich, with detailed information on these flats in terms of 238 categorical covariates describing characteristics such as the quality of bathroom equipment, whether the flat is a first-time rental, or whether a garden or a balcony is included. In addition, the 2007 Munich rent data contain two continuous covariates, the size of the flat and the year of the building's construction, as well as spatial information regarding in which of the 411 neighbourhoods the particular flat is located (see http://www.muenchen.de/mietspiegel for the official documentation of the rental guide).
Previous analyses of rent data collected in the city of Munich revealed that both the size of the flat and the year of the building's construction have non-linear predictor effects on the net rent. Also, spatial heterogeneity remained even after a number of further covariate effects were accounted for (Fahrmeir et al., 2004). Specifically, Kneib et al. (2010) demonstrated the beneficial use of the complete covariate information contained in the 238 categorical covariates. Moreover, Stasinopoulos et al. (2000) identified variance heteroscedasticity when

modelling the net rent from an earlier version of the Munich Rental Guide. To address this problem, Stasinopoulos et al. (2000) fitted a gamma distribution model in which both the mean and the dispersion were explicitly modelled. Additionally, Fahrmeir et al. (2004) considered a two-step estimation approach in which the squared residuals obtained from an ordinary least-squares estimation were successively used as weights in a weighted least-squares estimation. Instead of considering these approaches, we use GAMLSS to model heteroscedasticity directly. This is accomplished by including covariate effects on both the location and the variance parameters of the response distribution.

As a response distribution for the net rent per square metre, we consider the three-parameter t-distribution with location parameter $\theta_1 = \eta_\mu =: \mu$, scale parameter $\theta_2 = \exp(\eta_\sigma) =: \sigma$ and degrees of freedom $\theta_3 = \exp(\eta_{\texttt{df}}) =: \texttt{df}$. The probability density function of the net rent per square metre conditional on a set of predictor variables is thus given by

$$f(y_i|\mu_i, \sigma_i, \texttt{df}_i) = \frac{\Gamma(\frac{\texttt{df}_i+1}{2})}{\sigma_i \Gamma(\frac{1}{2})\Gamma(\frac{\texttt{df}_i}{2})\sqrt{\texttt{df}_i}} \left(1 + \frac{(y_i - \mu_i)^2}{(\sigma_i^2 \cdot \texttt{df}_i)}\right)^{(-(\texttt{df}_i+1)/2)} ,$$

(see Rigby and Stasinopoulos, 2005). The mean of the t-distribution is equal to μ, and its variance is given by $\sigma^2 \cdot \frac{\texttt{df}}{\texttt{df}-2}$. For each of the parameters μ, σ^2, and $\texttt{df}$, we consider the predictors

$$\begin{aligned}
\eta_{\mu_i} &= \beta_{0\mu} + \boldsymbol{x}_i^\top \boldsymbol{\beta}_\mu + f_{1\mu}(\text{size}_i) + f_{2\mu}(\text{year}_i) + f_{\text{spat}\mu}(s_i) , \\
\eta_{\sigma_i} &= \beta_{0\sigma} + \boldsymbol{x}_i^\top \boldsymbol{\beta}_\sigma + f_{1\sigma}(\text{size}_i) + f_{2\sigma}(\text{year}_i) + f_{\text{spat}\sigma}(s_i) , \\
\eta_{\texttt{df}_i} &= \beta_{0\texttt{df}} + \boldsymbol{x}_i^\top \boldsymbol{\beta}_{\texttt{df}} + f_{1\texttt{df}}(\text{size}_i) + f_{2\texttt{df}}(\text{year}_i) + f_{\text{spat}\texttt{df}}(s_i) ,
\end{aligned}$$

$i = 1, \ldots, n$, where $\beta_{0\theta_k}$ and $\boldsymbol{\beta}_{\theta_k}$ correspond to the intercept and parametric effects of the 238 categorical covariates (denoted by $\boldsymbol{x}_i^\top$), $f_{1\theta_k}(\text{size})$ and $f_{2\theta_k}(\text{year})$ are non-linear effects of the size of the flat and the year of construction respectively and $f_{\text{spat}\theta_k}(s)$ is a spatial effect based on the neighbourhood $s = 1, \ldots, 411$ within the city of Munich.

To sum up, this section presents a GAMLSS using the complete set of 238 categorical covariates in addition to the size of the flat, its year of construction and spatial information. Estimation and variable selection for this high-dimensional GAMLSS is accomplished by using gamboostLSS with linear base-learners for the effects corresponding to categorical predictor variables. Non-linear effects for size and year of construction of the flats are modelled using cubic P-spline base-learners each with 20 inner knots, a second-order difference penalty and four degrees of freedom. A Gaussian MRF base-learner with six degrees of freedom is assigned to the spatial effect (Sobotka and Kneib, 2010). Optimal boosting iterations are determined separately for each of the three model parameters using three-dimensional 10-fold cross validation. This strategy is computationally more expensive than using the same stopping iteration for all three predictors, yet it enables gamboostLSS to select models with very different complexities for each parameter.

To evaluate the predictive performance of the high-dimensional GAMLSS, we consider an alternative model based on the t-distribution with the same predictor structure as above but with a reduced set of categorical covariates, including only an expert selection of 28 effects. This expert set of covariates was used in the last official Munich Rental Guide and was also considered as a benchmark model in Kneib et al. (2010). In fact, the expert selection is not merely a subset of the original covariates but also involves transformation and combinations of the original covariates. Models based on this expert selection are referred to as 'expert

models' in the remainder of this section. In addition to GAMLSS with a t-distribution for the response, we estimate additive models based on squared error loss for both the high-dimensional and the expert sets of covariates. Those Gaussian additive models are part of the generalized additive models framework and are therefore denoted as GAMs. Component-wise gradient boosting with the squared error loss (see Bühlmann and Hothorn, 2007) is used to fit the GAMs. The same base-learners as those specified above are used to estimate the predictor $\theta_1 = \eta_\mu$ (i.e. the location parameter) of these models. Below, we evaluate the predictive performance of these different models in terms of their point predictions and prediction intervals.

4.2 Results from high-dimensional GAMLSS

Figures 3 and 4 show the estimated non-linear and spatial effects for the high-dimensional GAMLSS. Regarding the location parameter, the results are mostly consistent with previous findings from mean regression models (e.g. Kneib et al., 2010): Increased net rents per square metre are associated with flats that are located either in old buildings (constructed before 1900) or in rather new buildings. Similarly, small flats are more expensive (per square metre) than larger ones. The spatial effect indicates increased net rent per square metre in the center of Munich and also along the Isar River, which crosses Munich from south to north.
Despite the similarity between GAMLSS and conventional mean regression models, the former offers much richer, additional information in terms of the covariate effects on the scale parameter and the degrees of freedom. In our example, the size effects and the spatial effect on the scale parameter indicate that areas with a higher (lower) net rent per square metre are mostly associated with greater (less) variability. This intuitively makes sense and corresponds to the form of heteroscedasticity most frequently associated with applications involving housing data. The effect of the year of construction is much less prominent than the effect of size on the degrees of freedom. Neither the year of construction nor the spatial effect was selected by gamboostLSS for the predictor $\eta_{\texttt{df}}$.
For comparison, Figure 4 shows the spatial effects obtained from fitting a high-dimensional and an expert model GAM. In principle, the same areas identified by the location part η_{μ_i} of the high-dimensional GAMLSS were identified by the GAM, with a few difference in the absolute size of the estimated effects. Similarly, non-linear effects on the location parameters adopt basically the same forms (not displayed here) for GAM as for GAMLSS (Figure 3), with the range of effects being somewhat larger for the latter. This effect is most probably caused by the additional impact of covariates on the scale and degrees of freedom.
Among the 238 categorical covariates, only a small subset has non-negligible impact on the parameters of the response distribution. In the high-dimensional GAMLSS, lower values of the location parameter are, for example, associated with flats located in company houses or in the basement. The presence of a roof terrace, on the other hand, implies a surcharge on the location parameter. Larger uncertainty, i.e. a positive effect on the standard deviation, is associated with company housing, special kitchen equipment and the absence of facilities for warm-water generation. Negative effects on the degrees of freedom were identified for flats located in the basement, flats with a bathroom niche and those with special kitchen equipment.

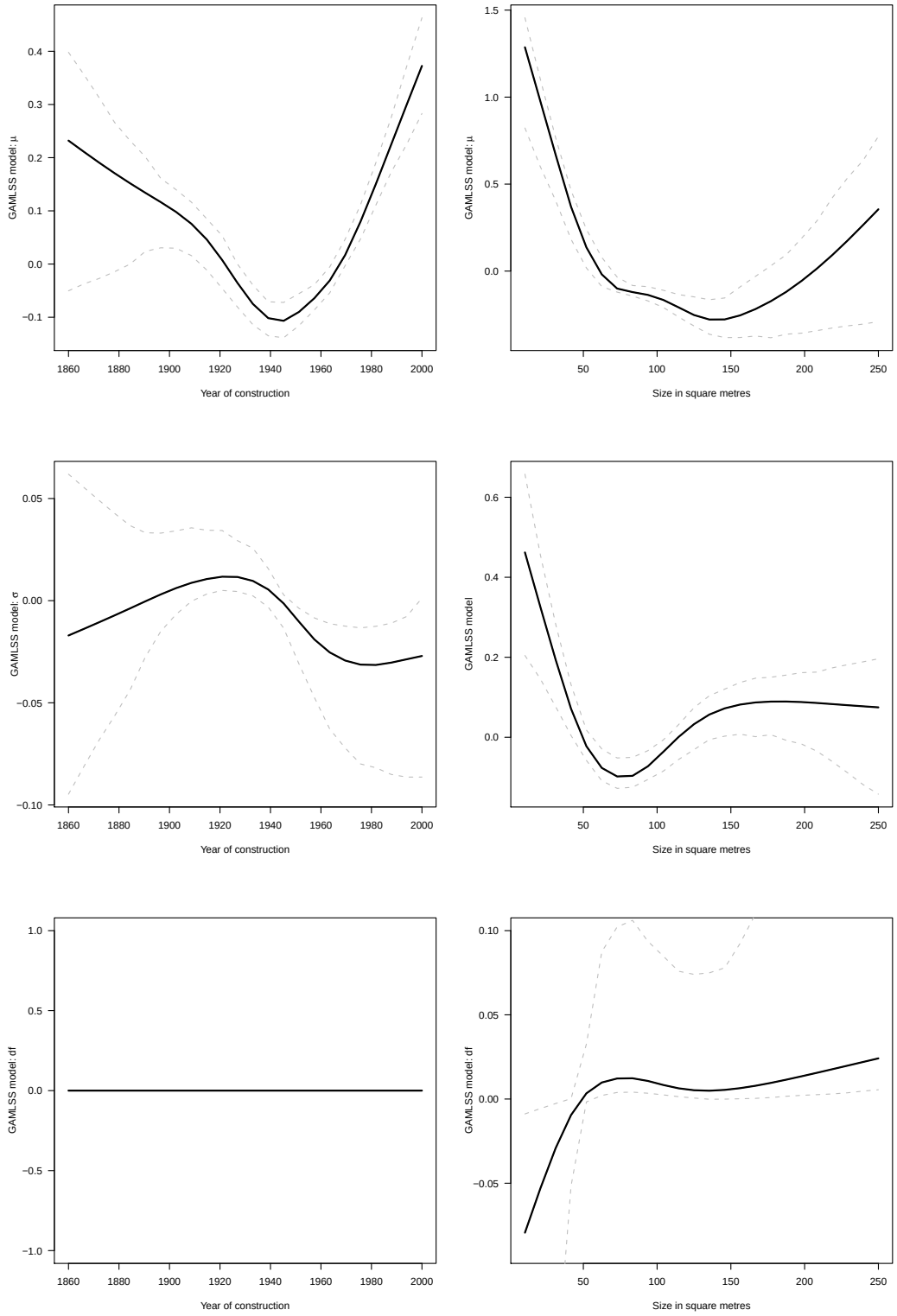

Figure 3: Munich Rental Guide: Estimated non-linear effects for the location parameter (top row), scale parameter (middle row) and the degrees of freedom (bottom row) obtained in a high-dimensional GAMLSS. Dashed lines represent 95% confidence bands, estimated from 100 bootstrap samples.

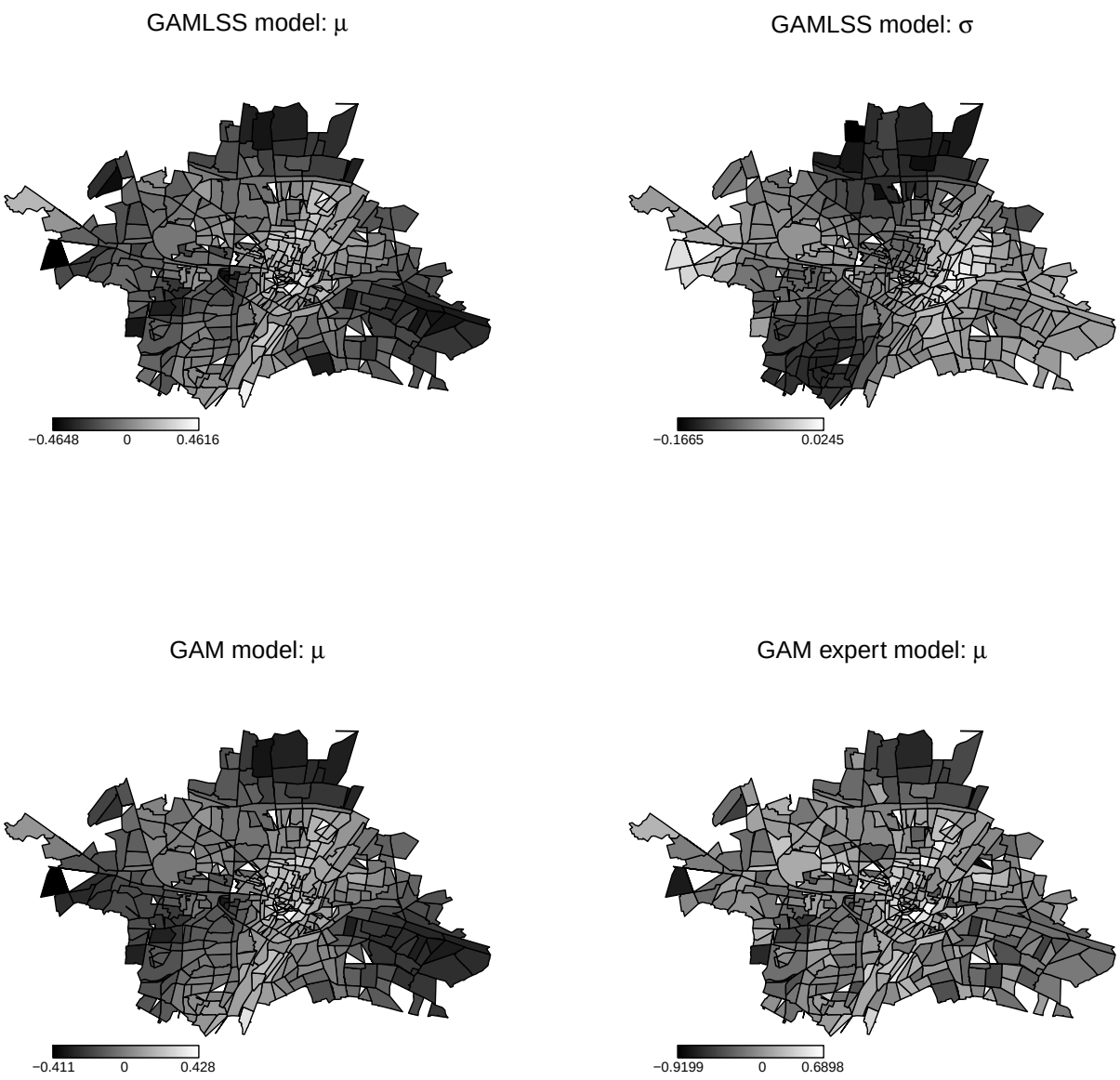

Figure 4: Munich Rental Guide: Estimated spatial effects obtained for the high-dimensional GAMLSS (top) and for GAMs (bottom). Estimates of the high-dimensional GAM (bottom left) are compared with those of the expert GAM (bottom right).

4.3 Predictive performance and prediction intervals

To analyse the prediction accuracy of the high-dimensional GAMLSS vs. that of the expert models and GAMs, we carried out a 10-fold cross validation. In each of the CV samples, the optimal boosting iterations were determined using an additional split-off data set. Hence, from every training set of the CV circle, one-fifth of the flats were excluded to find the optimal stopping iteration without touching the test data.
Figure 5 shows a parallel coordinate plot containing the average mean squared prediction errors obtained from the four models (high-dimensional GAMLSS/expert GAMLSS/ high-dimensional GAM/expert GAM). In accordance with the results of Kneib et al. (2010), the inclusion of all available covariates in high-dimensional models pays off with respect to increasing prediction accuracy. This is true for both GAMs and GAMLSS. In fact, the point predictions are only marginally better for GAMLSS than for GAM.

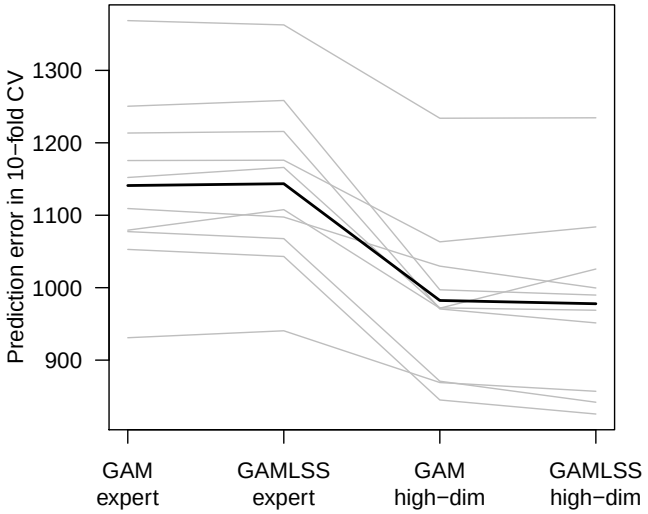

Figure 5: Munich Rental Guide: Mean squared error (MSE) in predictions compared for different models. Grey lines represent MSE for the different cross-validation runs and the dark line the average values.

While Figure 5 clearly suggests that the accuracy of point predictions obtained from classical GAMs carries over to those obtained from GAMLSS, the inclusion of covariate effects on parameters such as σ^2 and `df` additionally allows for an improved accuracy of the prediction intervals. Indeed, both GAMs and GAMLSS can be used to compute covariate-specific prediction intervals PI(X) for the net rent per square metre in Munich. The practical relevance of this approach is obvious: By setting lower and upper bounds for the expected net rent (conditional on the values of the covariates), PIs provide information on the level of variance in the net rent per square metre that tenants can expect.
We therefore use the conditional distributions of the four models to calculate the quantiles needed for the corresponding PI. By definition, the $\alpha \cdot 100\%$ of observations from a continuous

distribution should be smaller than the associated α-quantile (denoted by Q_α). A 95% PI is therefore given by

$$\text{PI}_{0.95}(X) = [Q_{0.025}(X), Q_{0.975}(X)]$$

(Meinshausen, 2006). It is clear that the three-parameter GAMLSS approach for the Munich Rental Guide allows for the construction of more flexible PIs than obtained with common GAMs, which rely on modelling the conditional mean of the net rent per square metre and therefore may not reflect other covariate-specific effects on the variance or the shape of the conditional distribution. As GAMLSS additionally regress scale and degrees of freedom to the covariates, the size of the resulting PI – and not only their centre – explicitly depends on a flat's characteristics. This effect is evident in Figure 6, in which the PIs resulting from high-dimensional GAMLSS and Gaussian models are compared. While the centres of the intervals (i.e. the conditional means μ_i) are relatively similar, there is a noticeable impact of the covariates on the quantiles of the conditional distribution(s) obtained with the GAMLSS. Clearly, the normality assumption implies homoscedasticity for the GAMs and therefore a constant width of all PIs obtained from these models. With GAMLSS, the sizes of the PIs are much more flexible and they take into account the impact of the covariates on the conditional variance of the net rent per square metre. This approach not only avoids the assumption of homoscedasticity, already identified as a problem regarding the rental guide (Stasinopoulos et al., 2000; Fahrmeir et al., 2004), but takes into account heteroscedasticity to obtain better predictions.

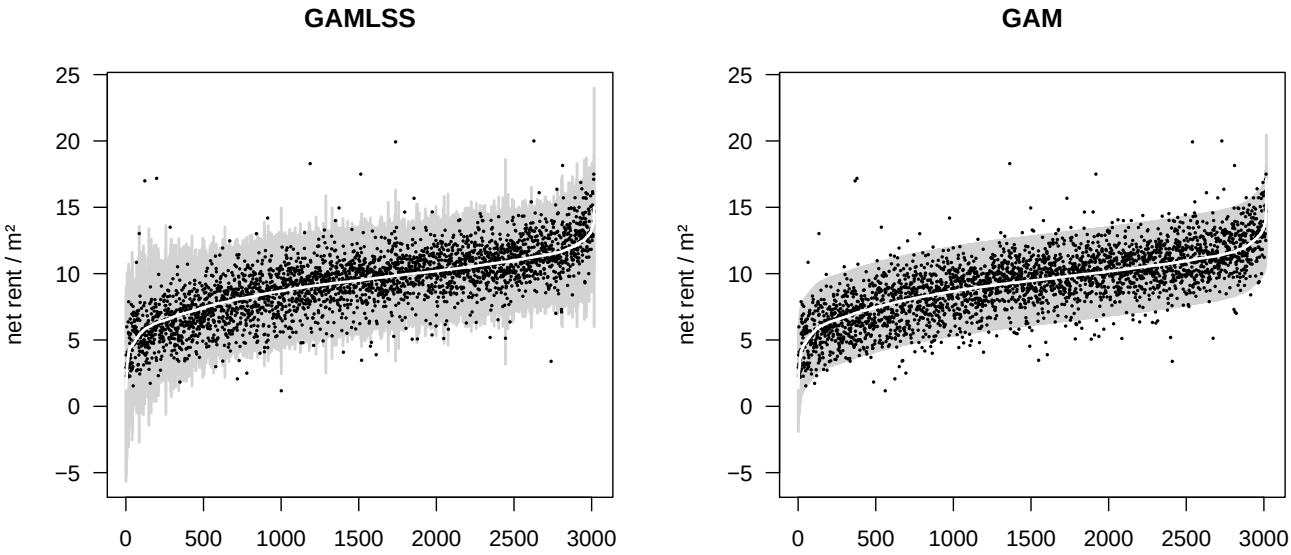

Figure 6: Munich Rental Guide: 95% prediction intervals based on the quantiles of the modelled conditional distribution from GAMLSS (left) and GAM (right). The solid white line represents point predictions (by which the values on the x axes were ordered); prediction intervals are shaded grey. The dark points correspond to the observed net rents per square metre contained in the sample.

This additional flexibility pays off for the Munich Rental Guide, as demonstrated here in our estimation of the coverage probability of the PIs from GAMLSS and GAMs. To evaluate the prediction accuracy of the PIs, we first draw 100 bootstrap samples from the complete data set and then fit both high-dimensional GAMs and GAMLSS models to the bootstrap samples. The covariates of the out-of-bootstrap flats are used to compute the PIs for the net rent per square metre of these flats. The average number of net rents lying within the intervals (sample coverage) is then compared for the two methods. As can be seen in Table 1, average sample coverage is closer to the expected coverage with the intervals obtained using GAMLSS than with those derived from the GAM approach.

α level	GAM	GAMLSS
99%	**97.49** (96.21–98.63)	**98.60** (97.72–99.36)
97.5%	**95.32** (92.40–97.30)	**96.80** (95.17–98.10)
95%	**92.23** (89.45–94.32)	**93.93** (92.07–95.80)
90%	**87.07** (83.86–90.44)	**88.52** (85.23–91.32)

Table 1: Munich Rental Guide: Average sample coverage (percent) of the prediction intervals obtained with high-dimensional GAM vs. GAMLSS. The range observed in 100 bootstrap samples is presented in brackets.

5 Conclusion

As a natural extension of the well-established GAM framework, GAMLSS have gained increasing popularity in recent years and their use has expanded to include many different fields of application (see for example the references in Section 1 or the information provided at http://gamlss.org). We applied GAMLSS to the Munich Rental Guide in order to adjust for heteroscedasticity in regression models predicting the net rent of Munich flats. Building on earlier approaches to address the problem of heteroscedasticity in this type of data (Stasinopoulos et al., 2000; Fahrmeir et al., 2004), we showed that the point predictions for the net rent per square metre obtained from GAMLSS are highly competitive with those obtained from mean regression methods. A substantial improvement of GAMLSS over traditional mean regression methods becomes evident when flat-specific covariates are used to derive prediction intervals for net rents per square metre. In this case, the coverage probabilities of intervals derived from GAMLSS are better than those obtained using Gaussian methods.

For the analysis of the Munich Rental Guide data, which particularly include also a spatial covariate, we developed the gamboostLSS algorithm, thereby extending the GAMLSS methodology to the analysis of high-dimensional data with potentially large numbers of informative covariates. Since estimation and selection of predictor effects are carried out simultaneously in gamboostLSS, the new algorithm addresses one of the remaining problems of the classical fitting methods currently available in R package **gamlss** (Stasinopoulos and Rigby, 2007). In contrast to gamboostLSS, the latter techniques have not been designed to handle high-dimensional data but instead rely on (partially biased) information criteria for variable selection.

Conversely, gamboostLSS can be considered as a natural extension of the gradient boosting framework (Friedman, 2001) to include regression models with multiple predictors. Consequently, the classical features of gradient boosting, such as shrinkage, variable selection and additive prediction functions (and thus the interpretability of estimates), carry over to each of the distribution parameters of a GAMLSS. In addition, the gamboostLSS algorithm presented in this paper naturally adapts to the structure of GAMLSS specified in Rigby and Stasinopoulos (2005). This cannot be accomplished with related machine-learning techniques such as support vector machines (Vapnik, 1996) or random forests (Breiman, 2001).
Our simulation study demonstrates the capability of gamboostLSS to produce sparse models, identifying the correct predictors in cases in which there are more covariates than observations ($p > n$). In low-dimensional settings, the algorithm converged to the same solution as obtained with the fitting methods of Stasinopoulos and Rigby (2007).
A limitation of gamboostLSS is its computationally expensive tuning procedure based on multi-dimensional cross-validation. Clearly, multi-dimensional stopping tends to become infeasible as the number of distribution parameters of a GAMLSS increases. A computationally less burdensome alternative to multi-dimensional stopping would be to use the same stopping iteration for all predictors (resulting in one-dimensional cross-validation). In simulations, we did not find strong evidence to support the necessity of multi-dimensional cross-validation, yet we noticed in the analysis of the Munich Rental Guide that multi-dimensional stopping is more convenient for adjusting GAMLSS to different complexity levels in parameter submodels. Further research is warranted on the topic of stopping procedures for this class of models. Another limitation of gamboostLSS is that classical tools for model diagnostics become invalid if applied to boosting estimates. Specifically, assessing residuals for normality may not be appropriate for gamboostLSS because boosting estimates shrink towards zero and residuals may therefore contain some of the remaining structure of the predictor effects not included in estimates of the GAMLSS parameters. Accordingly, in this study we relied on a prediction-based framework to validate our method. It should be noted that the current lack of appropriate model diagnostics is not a limitation restricted to gamboostLSS but is inherent to all boosting methods.
In summary, the advantages offered by gamboostLSS are the following: (i) Variable selection is accomplished automatically when gamboostLSS is applied. Gradient boosting produces a sparse solution with respect to all distribution parameters of a GAMLSS, implying that it is not necessary to rely on strategies based on information criteria. (ii) The proposed gamboostLSS algorithm can be applied to high-dimensional data sets in which the number of predictor variables exceeds the number of observations. This is currently not possible with the classical fitting techniques proposed by Stasinopoulos and Rigby (2007) (iii) By relying on an early stopping strategy, gamboostLSS has a built-in mechanism for the regularization of estimates. This essentially means that effect estimates shrink towards zero, thereby decreasing the variability of predictor effects and improving the prediction accuracy of the obtained GAMLSS solution. In view of these considerations, gamboostLSS offers a framework for a fully data-driven mechanism to select variables and predictor effects in GAMLSS.

Implementation

The gamboostLSS algorithm developed in this paper is implemented in the R (R Development Core Team, 2009) add-on package **gamboostLSS** (Hofner et al., 2010, available at http://R-forge.R-project.org/projects/gamboostlss). Models can be fitted using the function `gamboostLSS()`, which is based on the gradient boosting framework implemented in the R package **mboost** (Hothorn et al., 2010a,b). By relying on the **mboost** package, **gamboostLSS** incorporates a wide range of base-learners, for example, those of linear, smooth, spatial and random effects. In addition to making this infrastructure available for GAMLSS models, **mboost** constitutes a well-tested, mature software in the back end. Convenience functions to extract coefficients, plot the effects, make predictions or manipulate the model are available in **gamboostLSS**.

Acknowledgements

The authors thank Ludwig Fahrmeir for sharing the Munich Rental Guide data and Wendy Ran for the linguistic revision of the manuscript. The work of Andreas Mayr and Matthias Schmid was supported by the Interdisciplinary Center for Clinical Research (IZKF) at the University Hospital of the Friedrich-Alexander-Universität Erlangen-Nürnberg (Project J11).

References

Beyerlein, A., L. Fahrmeir, U. Mansmann, and A. Toschke (2008). Alternative regression models to assess increase in childhood BMI. *BMC Medical Research Methodology 8*(59).

Breiman, L. (2001). Random forests. *Machine Learning 45*, 5–32.

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

Bühlmann, P. and B. Yu (2003). Boosting with the L_2 loss: Regression and classification. *Journal of the American Statistical Association 98*(462), 324–338.

Cole, T. J., S. Stanojevic, J. Stocks, A. L. Coates, J. L. Hankinson, and A. M. Wade (2009). Age- and size-related reference ranges: A case study of spirometry through childhood and adulthood. *Statistics in Medicine 28*(5), 880–898.

Copas, J. B. (1983). Regression, prediction and shrinkage. *Royal Statistical Society, Series B 45*, 311–354.

de Castro, M., V. Cancho, and J. Rodrigues (2010). A hands-on approach for fitting long-term survival models under the GAMLSS framework. *Computer Methods and Programs in Biomedicine 97*, 168–177.

Efron, B. (1975). Biased versus unbiased estimation. *Advances in Mathematics 16*, 259–277.

Efron, B., I. Johnston, T. Hastie, and R. Tibshirani (2004). Least angle regression. *The Annals of Statistics 32*(2), 407–499.

Eilers, P. and B. Marx (1996). Flexible smoothing with b-splines and penalties. *Journal for the History of Astronomy 2*.

Fahrmeir, L., T. Kneib, and S. Lang (2004). Penalized structured additive regression for space-time data: a Bayesian perspective. *Statistica Sinica 14*, 731–761.

Fenske, N., L. Fahrmeir, P. Rzehak, and M. Höhle (2008). Detection of risk factors for obesity in early childhood with quantile regression methods for longitudinal data. Technical Report 38, Department of Statistics, Ludwig-Maximilians-Universität München.

Freund, Y. and R. Schapire (1996). Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning Theory*. San Francisco: Morgan Kaufmann Publishers Inc.

Friedman, J., T. Hastie, and R. Tibshirani (2000). Additive logistic regression: a statistical view of boosting. *Annals of Statistics 28*.

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

Greven, S. and T. Kneib (2010). On the behaviour of marginal and conditional akaike information criteria in linear mixed models. *Biometrika*. To appear.

Hastie, T. and R. Tibshirani (1990). *Generalized Additive Models*. London: Chapman & Hall.

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The elements of statistical learning: data mining, inference and prediction* (2 ed.). Springer.

Heller, G., D. M. Stasinopoulos, and R. Rigby (2006). The zero-adjusted Inverse Gaussian distribution as a model for insurance claims. In J. Hinde, J. Einbeck, and J.Newell (Eds.), *Proceedings of the 21th International Workshop on Statistical Modelling*, Galway, Ireland, pp. 226–233.

Hofner, B., A. Mayr, N. Fenske, and M. Schmid (2010). *gamboostLSS: Boosting Methods for GAMLSS models*. R package version 0.5-0.

Hothorn, T., P. Bühlmann, T. Kneib, M. Schmid, and B. Hofner (2010a). *mboost: Model-Based Boosting*. R package version 2.1-0.

Hothorn, T., P. Bühlmann, T. Kneib, M. Schmid, and B. Hofner (2010b). Model-based boosting 2.0. *Journal of Machine Learning Research 11*, 2109–2113.

Hothorn, T., F. Leisch, K. Hornik, and A. Zeileis (2005). The design and analysis of benchmark experiments. *Journal of Computational and Graphical Statistics 14*, 675–699.

Khondoker, M., C. Glasbey, and B. Worton (2009). A comparison of parametric and nonparametric methods for normalising cDNA microarray data. *Biometrical Journal 49*(6), 815–823.

Klein, J. P. and M. L. Moeschberger (2003). *Survival Analysis: Techniques for Censored and Truncated Data* (Second ed.). Springer.

Kneib, T., T. Hothorn, and G. Tutz (2009). Variable selection and model choice in geoadditive regression models. *Biometrics 65*, 626–634.

Kneib, T., S. Konrath, and L. Fahrmeir (2010). High-dimensional structured additive regression models: Bayesian regularisation, smoothing and predictive performance. *Applied Statistics*. to appear.

Meinshausen, N. (2006). Quantile regression forests. *Journal Machine Learning Research 7*, 983–999.

R Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Rawlings, J. O., S. G. Pantula, and D. A. Dickey (1998). *Applied Regression Analysis: A Research Tool* (2 ed.). New York: Springer.

Rigby, R. A. and D. M. Stasinopoulos (2004). Smooth centile curves for skew and kurtotic data modelled using the Box-Cox power exponential distribution. *Statistics in Medicine 23*, 3053–3076.

Rigby, R. A. and D. M. Stasinopoulos (2005). Generalized additive models for location, scale and shape (with discussion). *Applied Statistics 54*, 507–554.

Rigby, R. A. and D. M. Stasinopoulos (2006). Using the Box-Cox t distribution in GAMLSS to model skewness and kurtosis. *Statistical Modelling 6*, 209–229.

Ripley, B. D. (2004). Selecting amongst large classes of models. In N. Adams, M. Crowder, D. J. Hand, and D. Stephens (Eds.), *Methods and Models in Statistics*, pp. 155–170. London: Imperial College Press.

Rosset, S., J. Zhu, T. Hastie, and R. Schapire (2004). Boosting as a regularized path to a maximum margin classifier. *Journal of Machine Learning Research 5*, 941–973.

Rudge, J. and R. Gilchrist (2007). Measuring the health impact of temperatures in dwellings: Investigating excess winter morbidity and cold homes in the London Borough of Newham. *Energy and Buildings 39*, 847–858.

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

Schmid, M., S. Potapov, A. Pfahlberg, and T. Hothorn (2010). Estimation and regularization techniques for regression models with multidimensional prediction functions. *Statistics and Computing 20*(2), 139–150.

Sobotka, F. and T. Kneib (2010). Geoadditive expectile regression. *Computational Statistics and Data Analysis (submitted)*.

Stasinopoulos, D. M. and R. A. Rigby (2007). Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software 23(7)*.

Stasinopoulos, D. M., R. A. Rigby, and L. Fahrmeir (2000). Modelling rental guide data using mean and dispersion additive models. *The Statistician 49*, 479–493.

Vapnik, V. (1996). *The Nature of Statistical Learning Theory (Information Science and Statistics)*. Springer.

Villarini, G., J. Smith, and F. Napolitano (2010). Nonstationary modeling of a long record of rainfall and temperature over Rome. *Advances in Water Resources 33*, 1256–1267.

Villarini, G., J. Smith, F. Serinaldi, J. Bales, P. Bates, and W. Krajewski (2009). Flood frequency analysis for nonstationary annual peak records in an urban drainage basin. *Advances in Water Resources 32*, 1255–1266.