Journal of the Royal Statistical Society
SERIES C Applied Statistics

Appl. Statist. (2012)
61, Part 3, pp. 403–427

Generalized additive models for location, scale and shape for high dimensional data—a flexible approach based on boosting

Andreas Mayr,

Friedrich-Alexander-Universität Erlangen–Nürnberg, Erlangen, Germany

Nora Fenske,

Ludwig-Maximilians-Universität München, Germany

Benjamin Hofner,

Friedrich-Alexander-Universität Erlangen–Nürnberg, Erlangen, Germany

Thomas Kneib

Georg-August-Universität Göttingen, Germany

and Matthias Schmid

Friedrich-Alexander-Universität Erlangen–Nürnberg, Erlangen, Germany

[Received December 2010. Revised August 2011]

Summary. Generalized additive models for location, scale and shape (GAMLSSs) are a popular semiparametric modelling approach that, in contrast with conventional generalized additive models, regress not only the expected mean but also every distribution parameter (e.g. location, scale and shape) to a set of covariates. Current fitting procedures for GAMLSSs are infeasible for high dimensional data set-ups and require variable selection based on (potentially problematic) information criteria. The present work describes a boosting algorithm for high dimensional GAMLSSs 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 algorithm proposed was applied to Munich rental guide data, which are 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 GAMLSSs were found to be highly competitive and covariate-specific prediction intervals showed a major improvement over classical generalized additive models.

Keywords: Generalized additive models for location, scale and shape; Gradient boosting; High dimensional data; Prediction inference; Spatial information; Variable selection

1. Introduction

Generalized additive models for location, scale and shape (GAMLSSs) were introduced by Rigby and Stasinopoulos (2005) as a class of statistical models for regression problems with univariate response. GAMLSSs can be seen as a flexible alternative to generalized additive mod-

Address for correspondence: Andreas Mayr, Institut für Medizininformatik, Biometrie und Epidemiologie, Friedrich-Alexander-Universität Erlangen–Nürnberg, Waldstraße 6, 91054 Erlangen, Germany.
E-mail: andreas.mayr@imbe.med.uni-erlangen.de

 0035–9254/12/61403

els (GAMs) (Hastie and Tibshirani, 1990) as they extend the traditional GAM framework by a variety of modelling options. For example, GAMLSSs 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 GAMLSSs is given in Stasinopoulos and Rigby (2007).

Another key feature of GAMLSSs is that every parameter of the conditional response distribution is modelled by its own predictor and an associated link function. Whereas 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 multiparameter model is related to a set of predictor variables of interest.

In the same way as traditional GAMs, in GAMLSSs 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 and 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, GAMLSSs 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 recent years, GAMLSSs have been applied to many different areas, ranging from normalizing complementary DNA 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 effect of temperatures in dwellings (Rudge and Gilchrist, 2007). Clinical applications include long-term survival models for clinical studies (de Castro *et al.*, 2010), whereas Beyerlein *et al.* (2008) and Fenske *et al.* (2008) used GAMLSSs to investigate childhood obesity, in an approach that was 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 GAMLSSs, 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 that is offered by GAMLSSs 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 the use of the generalized Akaike information criterion (GAIC) for variable selection in GAMLSSs. This approach, however, has several shortcomings that are partially inherited from problems that are associated with the traditional AIC (Ripley (2004) and 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 that were proposed by Rigby and Stasinopoulos (2005) are infeasible when there are more covariates than observations.

To address these issues, we developed a boosting technique (denoted *gamboostLSS* in what follows) for estimating and selecting the predictor effects in GAMLSSs. 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 GAMLSSs. 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 (componentwise 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 estimate and select predictor effects in the GAMLSS framework simultaneously, 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, which is 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 330 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 of the flat. For this high dimensional data set, the usual GAMLSS fitting procedures were problematic with respect to variable selection because of the large number of covariates. Yet to include spatial information a new algorithm was needed, as with current fitting procedures inclusion was not possible. We show that GAMLSSs 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.

The paper is organized 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 GAMLSSs and compare them with 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, e.g. 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, 2011). The implementation is available with the R add-on package `gamboostLSS` (Hofner *et al.*, 2011).

2. Boosting generalized additive models for location, scale and shape

2.1. Generalized additive models for location, scale and shape

Rigby and Stasinopoulos (2005) referred to GAMLSSs as *semiparametric* regression-type models. Although 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 the relationship between covariates and the response may be modelled via additive smooth effects represented by (penalized) regression splines. The model class assumes observations y_i for $i = 1, 2, \ldots, n$ that are conditionally independent given a set of covariates and after having accounted for spatiotemporal effects. The conditional density $f_{\text{dens}}(y_i|\boldsymbol{\theta}_i)$ may depend on up to four distribution parameters $\boldsymbol{\theta}_i = (\theta_{i1}, \theta_{i2}, \theta_{i3}, \theta_{i4})^{\mathrm{T}}$. 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 $k = 1, \ldots, 4$ and depends additively on the covariates, including possible smooth non-linear predictor effects. Let $g_k(\cdot)$ be the known monotonic link functions for each predictor and $x_{k1}, \ldots, x_{kp_k}$ 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

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

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

For parametric models, the unknown quantities of a GAMLSS can be estimated by maximizing the log-likelihood

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

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 (which are denoted by $\hat{\eta}_{\theta_{ik}}$, $k = 1, \ldots, 4$) via the inverse link functions:

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

To estimate the predictor functions in η_{θ_k}, Rigby and Stasinopoulos (2005) introduced a penalized likelihood approach based on modified versions of the backfitting 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, backfitting steps are successively applied to the four distribution parameters, with the submodel fits of previous iterations used as offset values for those parameters that are not involved in the current backfitting step. For details on the two algorithms, see Stasinopoulos and Rigby (2007).

2.2. Variable selection based on the Akaike information criterion

Rigby and Stasinopoulos (2005) discussed a variety of strategies to select relevant predictors and covariate effects in GAMLSSs. Specifically, they proposed to use a generalized version of the AIC, defined as

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

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 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 Bayesian information criterion has generally been criticized as having a large variance, i.e. as being highly unstable with respect to the set of predictor variables that are included in the 'optimal' statistical model (see for example Rawlings *et al.* (1998)). Second, information criteria often result in the inclusion of a large number of non-informative predictor variables, i.e. 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, e.g. linear *versus* non-linear effects (Greven and Kneib, 2010).

A particular problem that is 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) suggested setting a between 2 and 4 but this choice seems difficult to justify theoretically.

Finally, even with a moderate number of potential covariates, the number of candidate GAMLSSs 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 that is 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

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 stagewise additive modelling (Friedman *et al.*, 2000; Friedman, 2001; Bühlmann and Yu, 2003). Therefore, boosting can be used as a technique for fitting generalized additive regression models.

Although there are many different types of boosting algorithms (see Hastie *et al.* (2009)), in this work we shall concentrate on the gradient boosting approach that was introduced by Friedman (2001) which is the foundation of model-based boosting (see Bühlmann and Yu (2003) and Bühlmann and Hothorn (2007)). The task is to derive a general prediction η by minimizing the expectation of a loss function $\rho(\cdot)$ assumed to be differentiable with respect to η:

$$\hat{\eta} = \arg\min_{\eta} (\mathbb{E}_{Y,X}[\rho\{Y, \eta(X)\}]),$$

where Y and X are the random variables for response and covariate(s) respectively. η here denotes a general prediction and will in the case of a GAMLSS be replaced by a combination of different η_{θ_k} from expression (1). In practice, to fit a GAM based on a sample of observations $(y_1, x_1), \ldots, (y_n, x_n)$, 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 $m = 1, \ldots, m_{\text{stop}}$, the boosting algorithm iteratively fits the negative gradient of the loss function. In every step, the current version of η is updated additively by a step length sl to approximate a minimum.

The functional gradient descent algorithm is formally given as follows.

Step 1: initialize $\hat{\eta}^{[0]} = 0$ and $m = 0$.

Step 2: increase m by 1. Compute the negative gradient and evaluate at the current estimate:

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

Step 3: fit the negative gradient vector **u** by a so-called *base learner* $h(\cdot)$. A base learner can be any kind of statistical tool that fits into a regression framework, with the simplest case being a univariate linear regression model. The negative gradient vector is taken as response variable and is fitted to the covariate:

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

Typical examples for base learners are classification and regression trees, linear models or penalized regression splines.

Step 4: update the prediction function with a step length $0 < \text{sl} \leqslant 1$:

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

Step 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 algorithm LARS (see Efron *et al.* (2004)).

For multi-dimensional X, the algorithm can be adapted to fit the covariates *componentwise*: for each base learner, one component of X is fitted 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 because of 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) and Hastie *et al.* (2009)). Furthermore, componentwise 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 multicollinearity, which arise often in high dimensional data, do not have a negative effect on the accuracy of estimation.

2.4. gamboostLSS

With the gamboostLSS algorithm, we propose a componentwise gradient descent algorithm that rotates between the different prediction functions of the distribution parameters for GAMLSSs. Analogously to the classical gradient descent algorithm that was 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 that was 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 $\eta_{\theta_k}, k = 1, \ldots, 4$. 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

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

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. $\mathbf{m}_{\text{stop}} = (m_{\text{stop},1}, \ldots, m_{\text{stop},4})^{\mathrm{T}}$ is therefore a vector of tuning parameters that can, for example, be determined by using cross-validation (CV) (see Section 2.5 for details).

In the case of GAMLSSs, the *componentwise* base learning strategy that was 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 $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_4)^{\mathrm{T}}$, each distribution parameter of a GAMLSS

has its separate additive predictor η_{θ_k} that is updated componentwise 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 on a different set of covariates at the final iteration, 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 Section 2.6 for a detailed 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

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

with $\rho=-l$ the negative log-likelihood of the response distribution and (Y,X) the random variables for the response and 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 $\boldsymbol{\eta}_i=(\eta_{\mu i},\eta_{\sigma i},\eta_{\nu i},\eta_{\tau i})^{\mathrm{T}}$, with $\mathbf{y}=(y_1,\ldots,y_n)^{\mathrm{T}}$ denoting the response vector with observations being conditionally independent given a set of covariates, and after having accounted for spatiotemporal effects. In each iteration of gamboostLSS, prespecified 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).

Step 1: initialize the additive predictors $\hat{\eta}_{\mu i}^{[0]},\hat{\eta}_{\sigma i}^{[0]},\hat{\eta}_{\nu i}^{[0]}$ and $\hat{\eta}_{\tau i}^{[0]}$ with offset values, e.g. $\hat{\eta}_{\theta ki}^{[0]}=0$ for $k=1,\ldots,4$ and $i=1,\ldots,n$.
Step 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 $m=0$.
Step 3: increase m by 1.
Step 4:

(a) set $k=0$;
(b) increase k by 1; if $m>m_{\text{stop},k}$ proceed to step 4(f); otherwise compute the negative partial derivative $-\partial\rho(y_i,\boldsymbol{\eta}_i)/\partial\eta_{\theta_k}$ by plugging in the current estimates $\boldsymbol{\eta}_i=(\hat{\eta}_{\mu i}^{[m-1]},\hat{\eta}_{\sigma i}^{[m-1]},\hat{\eta}_{\nu i}^{[m-1]},\hat{\eta}_{\tau i}^{[m-1]})$, which yields the vector of partial derivatives

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

(c) fit the negative gradient vector $\mathbf{u}_k^{[m-1]}$ 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

$$j^* = \underset{1 \leqslant j \leqslant p_k}{\arg\min} \left[\sum_{i=1}^{n} \{u_{ik}^{[m-1]} - h_{kj}(\cdot)\}^2 \right];$$

(e) update the additive predictor η_{θ_k},

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

where sl is a small step length ($0 < \mathrm{sl} \ll 1$); 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)–4(f) for $k = 2, \ldots, 4$.

Step 5: iterate steps 3 and 4 until $m > m_{\mathrm{stop},k}$ for all $k = 1, \ldots, 4$.

Owing to the additive updates in each iteration step ($\hat{\eta}_{\theta_k}^{[m]} = \hat{\eta}_{\theta_k}^{[m-1]} + \mathrm{sl}\, h_{kj^*}(\cdot)$), every resulting predictor η_{θ_k} follows an additive structure as in expression (1). The type of the resulting predictor functions $f_{j\theta_k}$ (the 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 that is represented by the function $f_{j\theta_k}$ for covariate component j on parameter θ_k (for possible base learners see Section 2.6).

One of the main characteristics of gamboostLSS is its ability to handle high dimensional set-ups in which there are more effects to estimate than observations. These set-ups are more likely for GAMLSSs 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 consistency with 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 can handle distributions that are even more complex than those considered by Rigby and Stasinopoulos (2005), as gamboostLSS does not require the number of distribution parameters (which is denoted as K in what follows) 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 $\mathbf{m}_{\mathrm{stop}}$. Here, $\mathbf{m}_{\mathrm{stop}}$ 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 $m_{\mathrm{stop},k} \to \infty, \forall k$, gamboostLSS converges to the same solution as provided by the classical maximum likelihood estimation (based on the algorithms that were 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 that are 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 that is involved in the additive updates. The step length contributes to the shrinkage effect and guarantees the stability of gamboostLSS; therefore, sl should be a small positive number (much less than 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 dependence 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, which is a value that is 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 CV techniques. To avoid overfitting effectively, 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 by 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 GAMLSSs, CV is more complex, as K different stopping iterations can be chosen to allow for different levels of complexity in each submodel. In the following sections, we distinguish between one-dimensional early stopping ($m_{\text{stop},k} \equiv m_{\text{stop}}$ for $k = 1, \ldots, K$) and multi-dimensional early stopping in which the elements of $m_{\text{stop},k}$ differ for $k = 1, \ldots, K$. Although the choice of the same stopping iteration for all distribution parameters ($m_{\text{stop},k} \equiv m_{\text{stop}}$ for $k = 1, \ldots, K$) 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 by 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 $k = 1, \ldots, K$.

2.6. Base learners and distributions

Another characteristic of the algorithm presented 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 that are available in the classical boosting framework, e.g. those provided by the R add-on package `mboost` for mean regression boosting (Hothorn *et al.*, 2010a, b; Bühlmann and Hothorn, 2007) can be also used for gamboostLSS. We consider the following types of base learners.

(a) Linear effects are represented by simple linear models estimated by the classical least squares method.
(b) Non-linear effects are modelled by using penalized regression splines (P-splines), i.e. 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).

(c) Spatial effects can be incorporated in 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 co-ordinates 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 for those effects with a neighbourhood structure. The covariate of the corresponding Markov random field is therefore given by an indicator specifying both a particular region and information on neighbouring regions (Sobotka and Kneib, 2011). We applied this base learner to model the spatial structure of the Munich rental guide data (see Section 4).

(d) 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 of modelling spatial and random-coefficient effects for GAMLSSs must be emphasized, since until now this has not been feasible, at least not with the currently available implementation of the classical algorithms that were provided by Stasinopoulos and Rigby (2007). The gamboostLSS algorithm proposed therefore not only extends the possibilities for fitting GAMLSSs to high dimensional data but also offers greater flexibility for modelling different types of effects in low dimensional settings.

Rigby and Stasinopoulos (2005) considered a large set of different GAMLSS distributions, all of which can be fitted by the boosting algorithm proposed. In this paper, we applied 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 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. 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 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 cannot estimate more coefficient effects than observations. Yet, in the smaller simulated settings and for the data sets that are 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 backfitting procedures (which are not presented here).

Our simulation study was aimed at answering the following questions.

(a) Can the gamboostLSS algorithm proposed correctly model the corresponding distribution parameters of the GAMLSS families in high dimensional settings?
(b) Can the algorithm identify the small subset of informative covariates?

(c) 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 by using the R software for statistical computing (R Development Core Team, 2011). The gamboostLSS implementation that is applied in this study is available with the R add-on package `gamboostLSS` (Hofner *et al.*, 2011).

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, 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

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

where the underlying additive linear predictors are given by

$$\log(\mu_i) = \eta_{\mu_i} = 1.5 + 1x_{1i} + 0.5x_{2i} - 0.5x_{3i} - 1x_{4i},$$

$$\log(\sigma_i) = \eta_{\sigma_i} = -0.4x_{3i} - 0.2x_{4i} + 0.2x_{5i} + 0.4x_{6i},$$

and where the covariates $\mathbf{x}_1, \ldots, \mathbf{x}_{1000}$ are $1 \times n$ vectors of independent identically distributed realizations of random variables $X_1, \ldots, X_{1000}$ following a multivariate normal distribution with a mean of 0 and a standard deviation sd equal to 1. The covariates are pairwise correlated with correlation coefficient $\rho = 0.5$. Thus 1000 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 gamboostLSS algorithm proposed. We considered 2×1000 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 identically distributed data set with 1000 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 are presented in Table 1: with two-dimensional early stopping the algorithm used some additional iterations in the fitting of the predictor for μ compared with the one-dimensional stopping. Nevertheless, the resulting empirical risk on the test data differs only slightly between the two stopping methods. This result suggests that, in this particular simulation setting, a one-dimensional search for the optimal m_{stop} can yield satisfying results.

Fig. 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 boxplots correspond to the empirical distribution of estimates from 100 independent samples of size $n = 800$, each generated from the negative binomial distribution that was 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, owing to the regularization property of the algorithm presented, all coefficient estimates are substantially shrunk towards zero. Note that, for low dimensional settings, it is possible to avoid shrinkage of effect estimates by letting the algorithm

Table 1. Results from the simulation studies based on 100 simulation runs†

	Results for the following settings:	
	Linear	*Non-linear*
Stopping iterations		
One dimensional	412.4 (67.9)	82.2 (11.7)
Two dimensional, η_μ	501.9 (96.9)	87.2 (16.8)
Two dimensional, η_σ	419.3 (124.0)	98.9 (22.8)
Selection rates (non-informative)		
η_μ	3.5%	1.3%
η_σ	1.8%	0.8%
Empirical risk		
One dimensional, early stopping	2670.2 (47.3)	704.4 (52.8)
Two dimensional, early stopping	2667.5 (47.9)	688.5 (51.7)

†The numbers presented are mean values with standard deviation in parentheses.

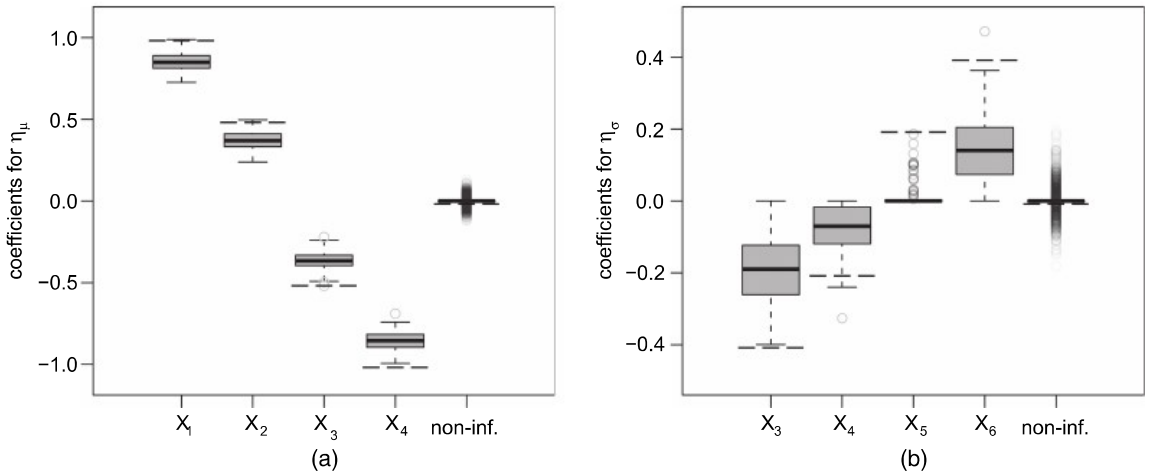

Fig. 1. Results from the simulation study, linear setting: the boxplots display the empirical distribution of the estimated coefficients for (a) the location parameter and (b) the scale parameter of the negative binomial distribution, obtained from running gamboostLSS in a high dimensional setting (100 simulation runs) (– – –, underlying true coefficients without shrinkage)

run until convergence. In this case, results of effect estimates converge to the results of the original backfitting algorithms of `gamlss`.

Estimates for all non-informative covariates are presented together in the last boxplot, which shows that the variable selection that is carried out by gamboostLSS works remarkably well. This view is further supported by the rates of selection, i.e. the proportion of simulation runs in which a particular base learner was chosen at least once before gamboostLSS was stopped (presented in Table 1).

The average number of variables selected from the 1000 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 set-ups 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 outcome distribution. 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

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

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

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

The underlying additive predictors were specified as follows:

$$\mu_i = \eta_{\mu_i} = 1 + 8\sin(x_{1i}) + 3\log(x_{2i}),$$

$$\log(\sigma_i) = \eta_{\sigma_i} = -0.8(x_{3i}^4 - x_{3i}^3 - 5x_{3i}^2) - 3x_{4i}.$$

All covariates were simulated through a multivariate normal distribution with pairwise correlation ($\rho = 0.5$). By sampling from the normal cumulative distribution function with subsequent rescaling, the resulting random variables $X_1, \ldots, X_{1000}$ are uniformly distributed on a grid from 0 to 3 and are pairwise correlated (correlation 0.5). The setting thus includes four informative (two for each distribution parameter) and 996 non-informative covariates. In addition to the survival times $\mathbf{y}_{\text{surv}}$, we simulated independent identically distributed censoring times $\mathbf{y}_{\text{cens}}$ following the same distribution as $\mathbf{y}_{\text{surv}}$. 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 right censored.

As base learners cubic P-splines (20 equidistant knots with a second-order difference penalty) were used, with 4 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 parameters. Hence, the learning algorithm could select from 2000 different base learners to update the GAMLSS fit. We again performed one- and two-dimensional early stopping, with m_{stop} selected by using an additional independent identically distributed data set consisting of 1000 observations following the same distribution as the original data. The average value of m_{stop} obtained from one-dimensional early stopping and two-dimensional early stopping as well as the predictive empirical risk are presented in Table 1.

Fig. 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. Although it seems to be problematic to detect the true function at the left-hand border region of X_2 (small values), effects of X_1, X_3 and X_4 are overall 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.

The informative covariates $X_1, \ldots, X_4$ were selected in every simulation run (selection rates of 100% for both parameters), whereas $X_5, \ldots, X_{1000}$ were selected on average in 1.7% of the

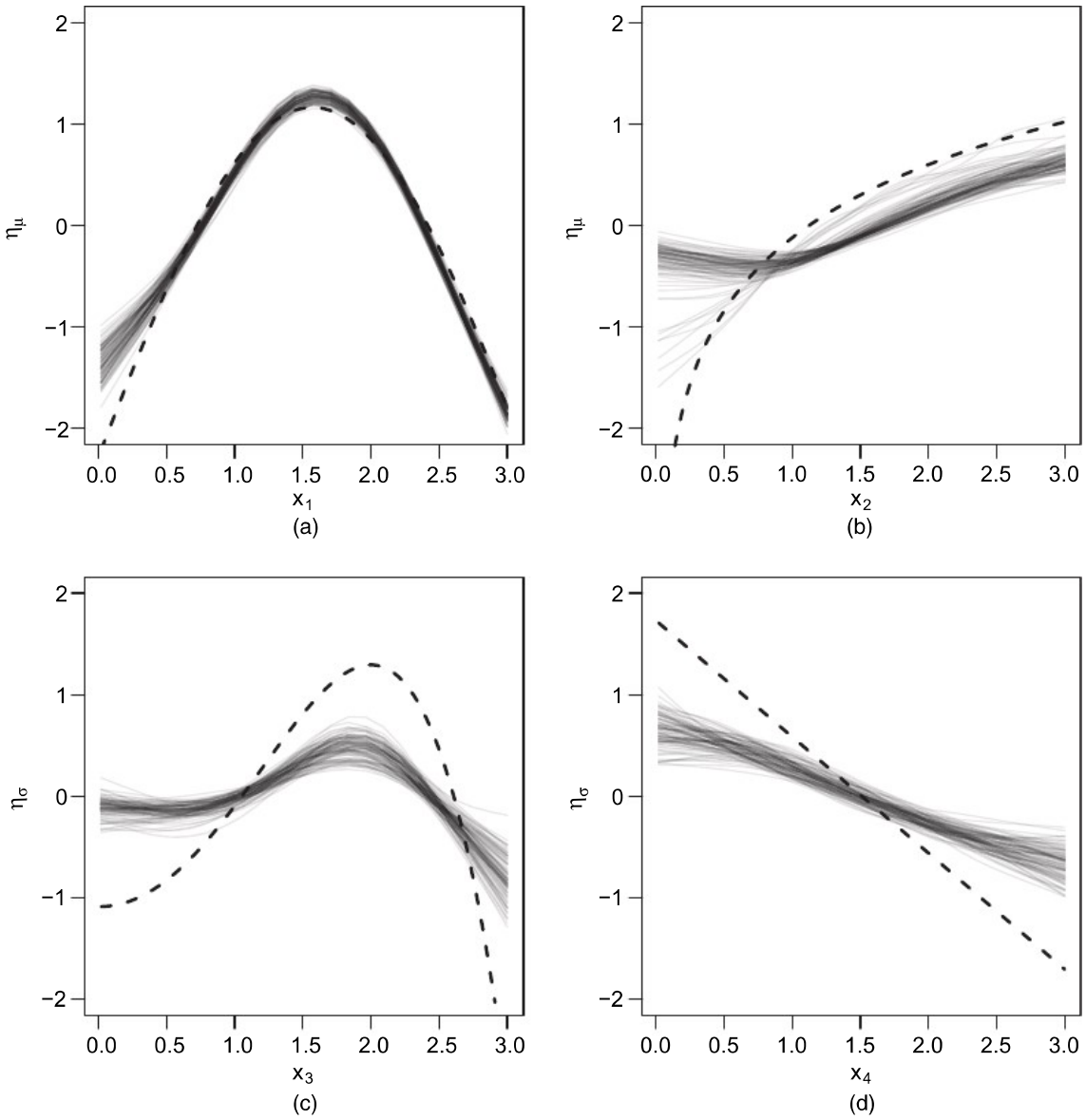

Fig. 2. Results from the simulation study, non-linear setting: estimated predictor functions (——) for (a), (b) the location parameter and (c), (d) the scale parameter of the log-logistic distribution obtained from running gamboostLSS at a high dimensional setting (100 simulation runs) (– – –, underlying true functions without shrinkage)

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 that is 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 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 GAMLSSs 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 prespecified coverage probability. Our sample comprises data obtained from $n = 3016$ flats within the city of Munich, with detailed information on these flats in terms of 328 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 which of the 411 neighbourhoods the particular flat is in (see `http://www.muenchen.de.uaccess.univie.ac.at/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 some 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 328 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 GAMLSSs 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 $\theta_3 = \exp(\eta_{\mathrm{df}}) =: \mathrm{df}$. The probability density function of the net rent per square metre conditional on a set of predictor variables is thus given by

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

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

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

$i = 1, \ldots, n$, where $\beta_{0\theta_k}$ and $\boldsymbol{\beta}_{\theta_k}$ correspond to the intercept and parametric effects of the 328 categorical covariates (denoted by $\mathbf{x}_i^{\mathrm{T}}$), $f_{1\theta_k}(\mathrm{size})$ and $f_{2\theta_k}(\mathrm{year})$ 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 328 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 are 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 by using cubic P-spline base learners each with 20 inner knots, a second-order difference penalty and 4 degrees of freedom. A Gaussian Markov random-field base learner with 6 degrees of freedom is assigned to the spatial effect (Sobotka and Kneib, 2011). Optimal boosting iterations are determined separately for each of the three model parameters by using three-dimensional tenfold CV. 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 a 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 GAM framework and are therefore denoted as GAMs. Componentwise 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 these different models in terms of their predictive performance.

4.2. Results from high dimensional generalized additive models for location, scale and shape

Figs 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 either in old buildings (constructed before 1900) or in quite new buildings. Similarly, small flats are more expensive (per square metre) than larger flats. The spatial effect indicates increased net rent per square metre in the centre of Munich and also along the Isar River, which crosses Munich from south to north.

Despite the similarity between GAMLSSs and conventional mean regression models, the former offer 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 or lower net rent per square metre are mostly associated respectively with greater or less variability. This intuitively makes sense and corresponds to the form of heteroscedasticity that is 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 η_{df}.

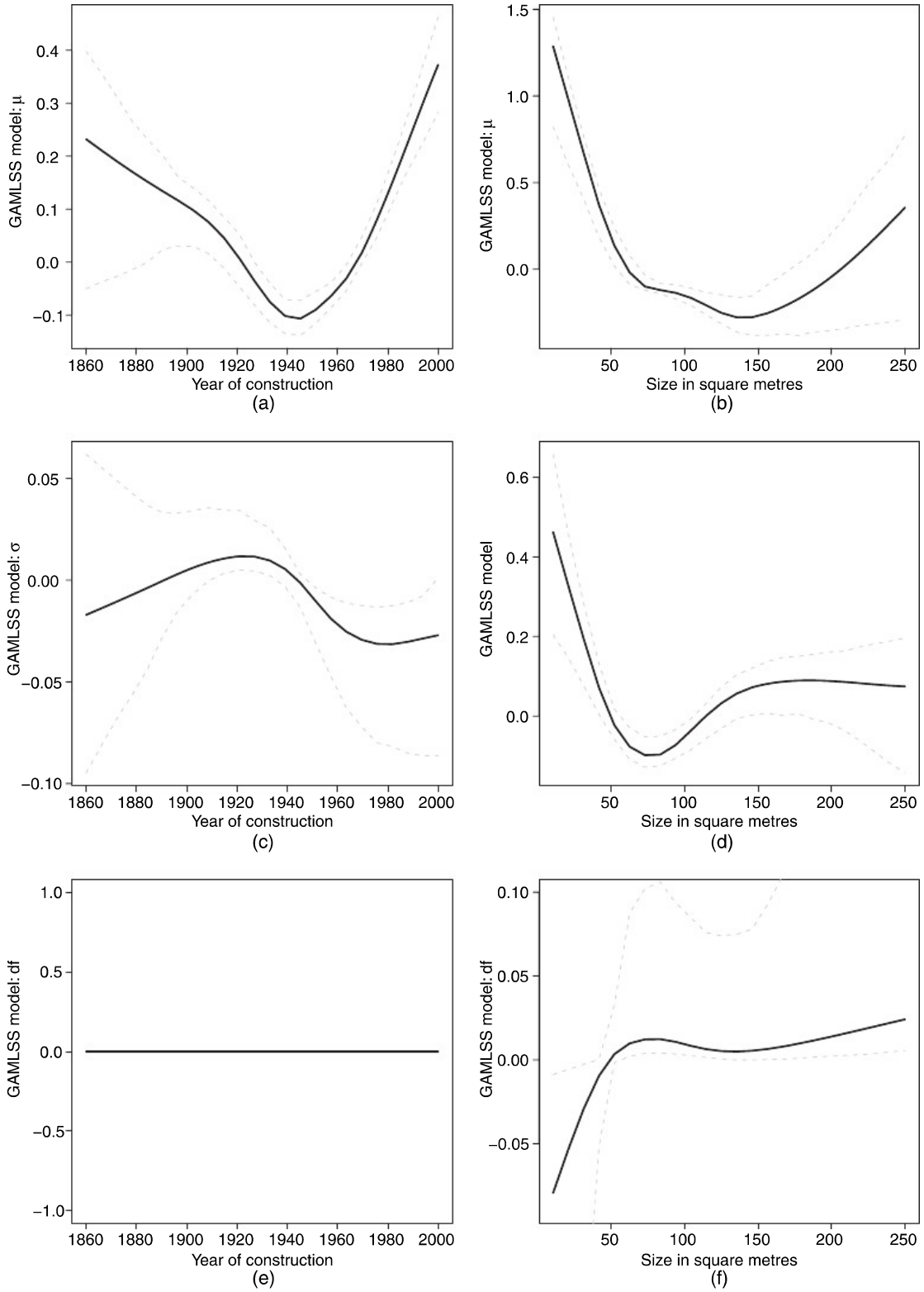

Fig. 3. Munich rental guide: estimated non-linear effects for (a), (b) the location parameter, (c), (d) the scale parameter and (e), (f) the degrees of freedom obtained in a high dimensional GAMLSS (-------, 95% confidence bands, estimated from 100 bootstrap samples)

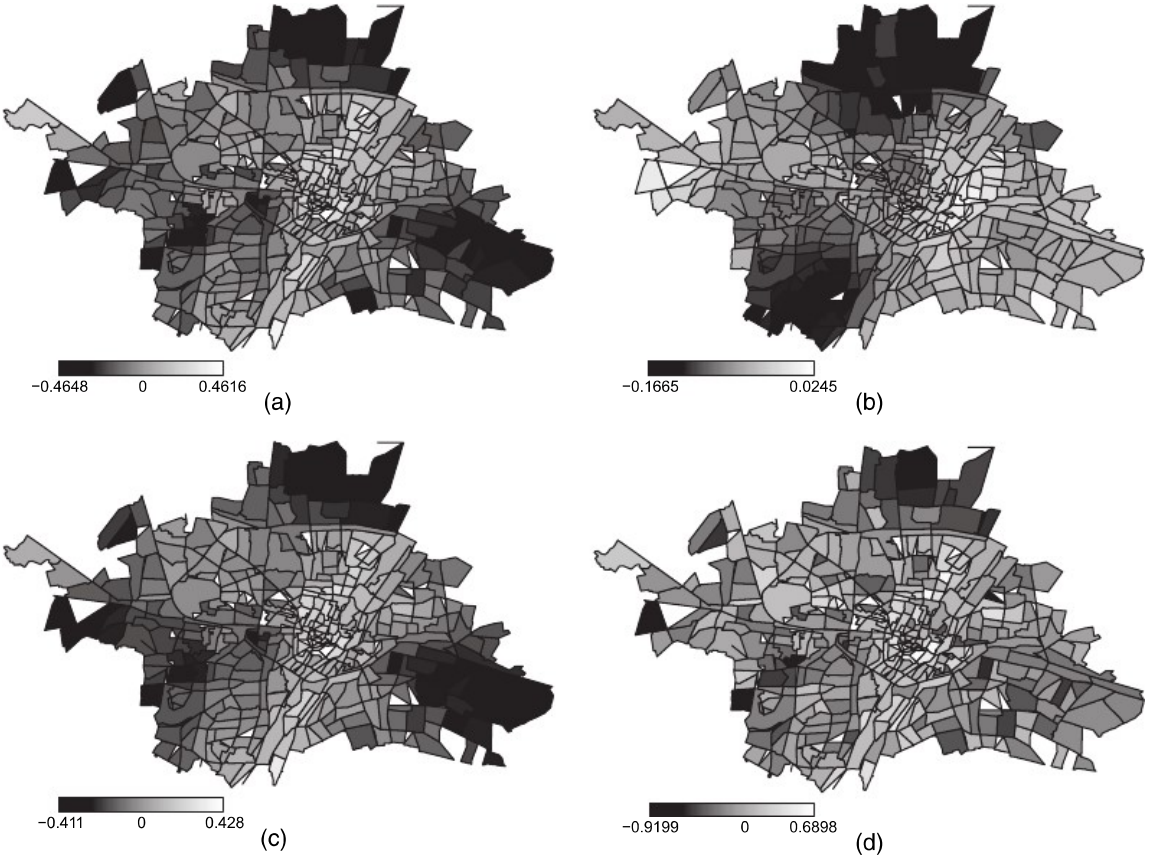

Fig. 4. Munich rental guide: estimated spatial effects obtained for (a) (μ) and (b) (σ) the high dimensional GAMLSS and (c) (μ) and (d) (μ, expert model) the high dimensional GAM

For comparison, Fig. 4 shows the spatial effects that were 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 differences in the absolute size of the estimated effects. Similarly, non-linear effects on the location parameters adopt basically the same forms (which are not displayed here) for GAMs as for GAMLSSs (Fig. 3), with the range of effects being somewhat larger for the latter. This effect is most probably caused by the additional effect of covariates on the scale and degrees of freedom.

Among the 328 categorical covariates, only a small subset has a non-negligible influence on the parameters of the response distribution. In the high dimensional GAMLSS, lower values of the location parameter are, for example, associated with flats in company houses or in the basement. The presence of a roof terrace, in contrast, 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 in the basement, flats with a bathroom niche and those with special kitchen equipment.

4.3. Predictive performance and prediction intervals

To analyse the accuracy of prediction of the high dimensional GAMLSS *versus* that of the expert models and GAMs, we carried out a tenfold CV. In each of the CV samples, the optimal boosting iterations were determined by using an additional split-off data set. Hence, from

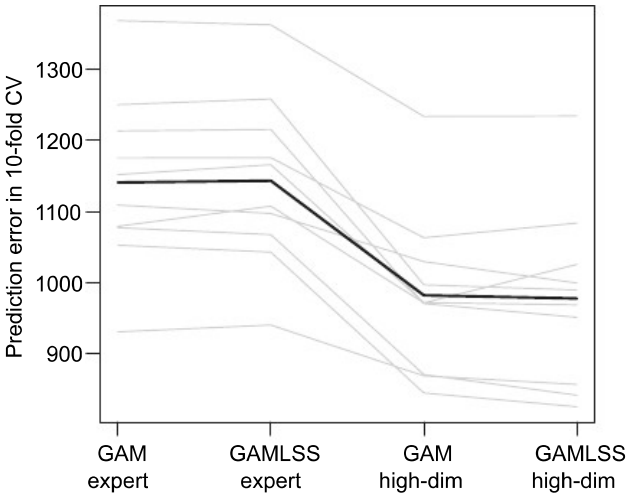

Fig. 5. Munich rental guide: mean-squared error in predictions compared for various models (——, mean-squared errors for the different cross-validation runs; ——, average values)

every training set of the CV circle, a fifth of the flats were excluded to find the optimal stopping iteration without touching the test data.

Fig. 5 shows a parallel co-ordinate plot containing the average mean-squared prediction errors obtained from the four models (high dimensional GAMLSS, expert GAMLSS, high dimensional GAM and 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 GAMLSSs. In fact, the point predictions are only marginally better for GAMLSSs than for GAMs.

Although Fig. 5 clearly suggests that the accuracy of point predictions obtained from classical GAMs carries over to those obtained from GAMLSSs, the inclusion of covariate effects on parameters such as σ and df additionally allows for an improved accuracy of the prediction intervals (PIs). Indeed, both GAMs and GAMLSSs can be used to compute covariate-specific PIs 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 that are needed for the corresponding PI. By definition, the $100\alpha\%$ of observations from a continuous distribution should be smaller than the associated α-quantile (which is denoted by Q_α). A 95% PI is therefore given by

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

(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 GAMLSSs additionally regress scale and degrees of freedom to the covariates, the size of the resulting PIs—and not only their centre—explicitly depends on a flat's characteristics. This effect is evident in Fig. 6, in which the PIs resulting from high dimensional GAMLSS and Gaussian models are compared. Although the centres of the intervals (i.e. the conditional means μ_i) are relatively similar, there is a noticeable effect of the covariates on the

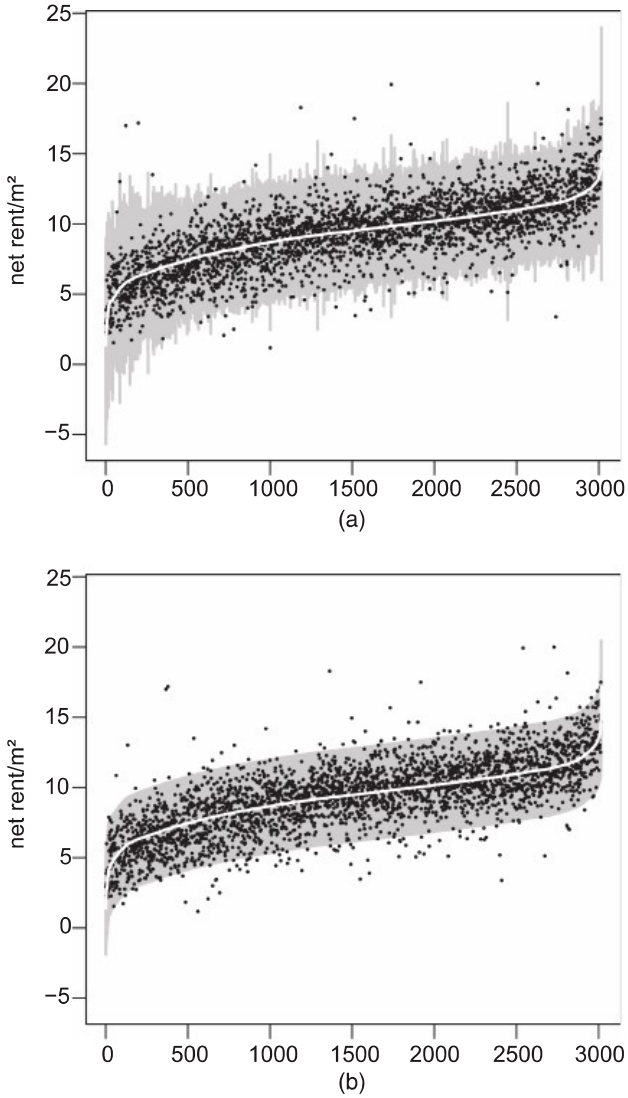

Fig. 6. Munich rental guide: 95% PIs based on the quantiles of the modelled conditional distribution from (a) GAMLSSs and (b) GAMs (═, point predictions (by which the values on the x-axes were ordered); ■, PIs; ●, observed net rents per square metre contained in the sample)

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 GAMLSSs, the sizes of the PIs are much more flexible and they take into account the effect of the covariates on the conditional variance of the net rent per square metre. This approach not only avoids the assumption of homoscedasticity, which has already been identified as a problem regarding the rental guide (Stasinopoulos *et al.*, 2000; Fahrmeir *et al.*, 2004), but also takes into account heteroscedasticity to obtain better predictions.

This additional flexibility pays off for the Munich rental guide, as demonstrated here in our estimation of the coverage probability of the PIs from GAMLSSs and GAMs. To evaluate the prediction accuracy of the PIs, we first draw 100 bootstrap samples from the complete data set

Table 2. Munich rental guide: average sample coverage of the PIs obtained with high dimensional GAMs versus GAMLSSs†

α-level (%)	Coverage (%) of PIs by the following methods:	
	GAMs	*GAMLSSs*
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)

†The range observed in 100 bootstrap samples is presented in parentheses.

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 2, the average sample coverage is closer to the expected coverage with the intervals obtained by using GAMLSSs than with those derived from the GAM approach.

5. Conclusion

As a natural extension of the GAM framework, GAMLSSs 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 GAMLSSs to the Munich rental guide 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 GAMLSSs are highly competitive with those obtained from mean regression methods, although compared with classical GAMs the improvement arising from the usage of GAMLSSs for the point prediction can be seen as only marginal. A substantial improvement of GAMLSSs over traditional mean regression methods, however, becomes evident when flat-specific covariates are used to derive PIs for net rents per square metre. In this case, the coverage probabilities of intervals derived from GAMLSSs are better than those obtained by 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 that are currently available in R package `gamlss` (Stasinopoulos and Rigby, 2007).

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 that was

presented in this paper naturally adapts to the structure of the GAMLSS that was 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 ability 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). It is therefore also possible to use gamboostLSS in high dimensional settings to perform variable selection and to use the selected variables to refit a low dimensional model yielding the same unshrunk effect estimates as the original fitting algorithms.

A limitation of gamboostLSS is its computationally expensive tuning procedure based on multi-dimensional CV. 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 CV). In simulations, we did not find strong evidence to support the necessity of multi-dimensional CV, yet we noted in the analysis of the Munich rental guide that multi-dimensional stopping is more convenient for adjusting GAMLSSs to different levels of complexity 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 distributions of residuals 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 that are not included in estimates of the GAMLSS parameters. Accordingly, in this study we relied on a prediction-based framework to validate our method. The current lack of appropriate model diagnostics is not a limitation that is restricted to gamboostLSS but is inherent to all boosting methods and related regularization techniques such as the lasso (Tibshirani, 1996).

In summary, the advantages that are offered by gamboostLSS are as follows.

(a) 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.
(b) The gamboostLSS algorithm proposed 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 that were proposed by Stasinopoulos and Rigby (2007).
(c) 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 accuracy of prediction of the GAMLSS solution obtained.

In view of these considerations, gamboostLSS offers a framework for a fully data-driven mechanism to select variables and predictor effects in GAMLSSs.

6. Implementation

The gamboostLSS algorithm that is developed in this paper is implemented in the R (R Development Core Team, 2011) add-on package `gamboostLSS` (Hofner *et al.*, 2011). Models can be

fitted by using the function `gamboostLSS()`, which is based on the gradient boosting framework that is 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, e.g. those of linear, smooth, spatial and random effects. In addition to making this infrastructure available for GAMLSS models, `mboost` constitutes 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 at the University Hospital of the Friedrich-Alexander-Universität Erlangen–Nürnberg (project J11). Nora Fenske received support from the Munich Center of Health Sciences. Benjamin Hofner and Thomas Kneib received support from the German Research Foundation, grant HO 3242/1-3 and grant KN 922/4-1.

References

Beyerlein, A., Fahrmeir, L., Mansmann, U. and Toschke, A. (2008) Alternative regression models to assess increase in childhood BMI. *BMC Med. Res. Methodol.*, **8**, article 59.

Breiman, L. (2001) Random forests. *Mach. Learn.*, **45**, 5–32.

Bühlmann, P. and Hothorn, T. (2007) Boosting algorithms: regularization, prediction and model fitting (with discussion). *Statist. Sci.*, **22**, 477–522.

Bühlmann, P. and Yu, B. (2003) Boosting with the L_2 loss: regression and classification. *J. Am. Statist. Ass.*, **98**, 324–338.

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

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

Copas, J. B. (1983) Regression, prediction and shrinkage (with discussion). *J. R. Statist. Soc.* B, **45**, 311–354.

Efron, B. (1975) Biased versus unbiased estimation. *Adv. Math.*, **16**, 259–277.

Efron, B., Johnstone, I., Hastie, T. and Tibshirani, R. (2004) Least angle regression. *Ann. Statist.*, **32**, 407–499.

Eilers, P. and Marx, B. (1996) Flexible smoothing with B-splines and penalties. *Statist. Sci.*, **2**, 89–121.

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

Fenske, N., Fahrmeir, L., Rzehak, P. and Höhle, M. (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, Munich.

Freund, Y. and Schapire, R. (1996) Experiments with a new boosting algorithm. In *Proc. 13th Int. Conf. Machine Learning Theory*. San Francisco: Morgan Kaufmann.

Friedman, J. H. (2001) Greedy function approximation: a gradient boosting machine. *Ann. Statist.*, **29**, 1189–1232.

Friedman, J., Hastie, T. and Tibshirani, R. (2000) Additive logistic regression: a statistical view of boosting. *Ann. Statist.*, **28**, 337–374.

Greven, S. and Kneib, T. (2010) On the behaviour of marginal and conditional Akaike information criteria in linear mixed models. *Biometrika*, **97**, 773–789.

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

Hastie, T., Tibshirani, R. and Friedman, J. (2009) *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, 2nd edn. New York: Springer.

Heller, G., Stasinopoulos, D. M. and Rigby, R. (2006) The zero-adjusted Inverse Gaussian distribution as a model for insurance claims. In *Proc. 21st Int. Wrkshp Statistical Modelling, Galway* (eds J. Hinde, J. Einbeck and J. Newell), pp. 226–233.

Hofner, B., Mayr, A., Fenske, N. and Schmid, M. (2011) gamboostLSS: boosting methods for GAMLSS models. *R Package Version 1.0-0*. (Available from `http://cran.r-project.org.uaccess.univie.ac.at/package=gamboostLSS`.)

Hothorn, T., Bühlmann, P., Kneib, T., Schmid, M. and Hofner, B. (2010a) mboost: model-based boosting. *R Package Version 2.0-12*.

Hothorn, T., Bühlmann, P., Kneib, T., Schmid, M. and Hofner, B. (2010b) Model-based boosting 2.0. *J. Mach. Learn. Res.*, **11**, 2109–2113.
Hothorn, T., Leisch, F., Hornik, K. and Zeileis, A. (2005) The design and analysis of benchmark experiments. *J. Computnl Graph. Statist.*, **14**, 675–699.
Khondoker, M., Glasbey, C. and Worton, B. (2009) A comparison of parametric and nonparametric methods for normalising cDNA microarray data. *Biometr. J.*, **49**, 815–823.
Klein, J. P. and Moeschberger, M. L. (2003) *Survival Analysis: Techniques for Censored and Truncated Data*, 2nd edn. Berlin: Springer.
Kneib, T., Hothorn, T. and Tutz, G. (2009) Variable selection and model choice in geoadditive regression models. *Biometrics*, **65**, 626–634.
Kneib, T., Konrath, S. and Fahrmeir, L. (2010) High dimensional structured additive regression models: Bayesian regularization, smoothing and predictive performance. *Appl. Statist.*, **60**, 51–70.
Meinshausen, N. (2006) Quantile regression forests. *J. Mach. Learn. Res.*, **7**, 983–999.
Rawlings, J. O., Pantula, S. G. and Dickey, D. A. (1998) *Applied Regression Analysis: a Research Tool*, 2nd edn. New York: Springer.
R Development Core Team (2011) *R: a Language and Environment for Statistical Computing*. Vienna: R Foundation for Statistical Computing.
Rigby, R. A. and Stasinopoulos, D. M. (2004) Smooth centile curves for skew and kurtotic data modelled using the Box-Cox power exponential distribution. *Statist. Med.*, **23**, 3053–3076.
Rigby, R. A. and Stasinopoulos, D. M. (2005) Generalized additive models for location, scale and shape (with discussion). *Appl. Statist.*, **54**, 507–554.
Rigby, R. A. and Stasinopoulos, D. M. (2006) Using the Box-Cox t distribution in GAMLSS to model skewness and kurtosis. *Statist. Modlling*, **6**, 209–229.
Ripley, B. D. (2004) Selecting amongst large classes of models. In *Methods and Models in Statistics* (eds N. Adams, M. Crowder, D. J. Hand and D. Stephens), pp. 155–170. London: Imperial College Press.
Rosset, S., Zhu, J., Hastie, T. and Schapire, R. (2004) Boosting as a regularized path to a maximum margin classifier. *J. Mach. Learn. Res.*, **5**, 941–973.
Rudge, J. and Gilchrist, R. (2007) Measuring the health impact of temperatures in dwellings: investigating excess winter morbidity and cold homes in the London Borough of Newham. *En. Build.*, **39**, 847–858.
Schmid, M. and Hothorn, T. (2008) Boosting additive models using component-wise P-splines. *Computnl Statist. Data Anal.*, **53**, 298–311.
Schmid, M., Potapov, S., Pfahlberg, A. and Hothorn, T. (2010) Estimation and regularization techniques for regression models with multidimensional prediction functions. *Statist. Comput.*, **20**, 139–150.
Sobotka, F. and Kneib, T. (2011) Geoadditive expectile regression. *Computnl Statist. Data Anal.*, to be published, doi 10.1016/j.csda.2010.11.015.
Stasinopoulos, D. M. and Rigby, R. A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *J. Statist. Softwr.*, **23**, no. 7.
Stasinopoulos, D. M., Rigby, R. A. and Fahrmeir, L. (2000) Modelling rental guide data using mean and dispersion additive models. *Statistician*, **49**, 479–493.
Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. *J. R. Statist. Soc.* B, **58**, 267–288.
Vapnik, V. (1996) *The Nature of Statistical Learning Theory*. Berlin: Springer.
Villarini, G., Smith, J. and Napolitano, F. (2010) Nonstationary modeling of a long record of rainfall and temperature over Rome. *Adv. Wat. Resour.*, **33**, 1256–1267.
Villarini, G., Smith, J., Serinaldi, F., Bales, J., Bates, P. and Krajewski, W. (2009) Flood frequency analysis for nonstationary annual peak records in an urban drainage basin. *Adv. Wat. Resour.*, **32**, 1255–1266.