LMU

LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

INSTITUT FÜR STATISTIK

Christian Lindenlaub

Concurvity in Geo-Additive Models

Master Thesis

Supervision: Prof. Dr. Fabian Scheipl
Department of Statistics – University of Munich

December 26, 2012

Abstract

If covariate and spatial effects are modeled at the same time in order to cover spatial autocorrelation and unobserved heterogeneity, it will lead to wrong or attenuated effects in the presence of "concurvity". This is caused because spatial autocorrelation cannot separate clearly between spatial and covariate effect. Flexible modeling of the spatial effect includes that it consists of enough degrees of freedom for absorbing the covariate effect partially. This falsification of the estimated covariate effects can be prevented or weakened by modifying the spatial effect.

The basic idea is the modification of the spatial effect in a way which can only reproduce the variability it cannot be explained by covariate information in principle. Technically, it can be done by making basis functions, used for spatial effect orthogonal to the basis functions, to the covariate effect. This idea is implemented as a new operator of the `mboost` package. Simulations are conducted to investigate the performance of the new %ll%-operator and its strengths and weaknesses. In addition, they identified certain situations where the %ll%-operator perform well.

Finally, the %ll%-operator is used for an ecological application to investigate the impacts of climate change of the tree population in Bavaria. That is the cause because there is hardly any other industry except for the forestry which is so dependent on the natural environment. Successful management of the forests is inextricably linked to the adaptation to natural climatic conditions. The model indicates a strong decline of the Spruce (*Picea abies*) in Bavaria.

Key Words: concurvity, boosting, pGAM, climate change, Picea abies

Acknowledgement

To Fabian Scheipl, for his excellent supervision during my further work on this thesis.

To Karl H. Mellert and the Bayerischen Forstverwaltung for providing the data.

To Florian, Ariane, Julia and others for spending with me long hours of discussions.

To my family, for supporting me.

To Nadja, for proof-reading.

To Valerie, for hugs, kisses and giggles.

Contents

1. Introduction

"Prediction is very difficult, especially about the future."

Niels Bohr (1885 – 1962)

This well known quotation by the Danish physicist Niels Bohr, who won the Nobel Prize in 1922, emphasis one important requirement on the modern statistics. However, the modern statistics has further goals besides prediction. The analysis of the relationship between given variables is equally important. Linear models and their extensions provide this possibility.
Nowadays, many statisticians have a broad experience of fitting linear models with uncorrelated errors.

"Everything is related to everything else, but near things are more related than distant things."

Waldo R. Tobler (1930 – today)

Tobler [1970]'s first law of geography as well as ecology [Legendre and Fortin, 1989; Fortin and Dale, 2005] violates the assumption of uncorrelated errors. The first law of geography states that everything in space is related but the relatedness of things decreases with distance. Thereby, Tobler [1970] forms the basis for spatial autocorrelation and geo-statistics. Spatial dependence implies that activities in one region effect activities in another region.

By considering this background, it becomes clear that the "classical" linear models with the central assumption of independent observations are no longer an adequate modeling tool. According to He [2004], generalized additive models [Hastie and Tibshirani, 1990] are mainly used for geo-statistical analysis because they allow nonparametric relationships between independent variables and their response. However, the generalized additive models do not solve the problem of spatial auto-correlation in the data either. Nevertheless, the models try an extension in order to solve the problem and also include an additional spatial effect.

Furthermore, wrong or attenuated effects are caused if covariate and spatial effects are modeled at the same time in order to cover spatial autocorrelation. This occurs because spatial autocorrelation cannot separate clearly between spatial and covariate effects. Thus, the covariate effect is partially absorbed by the spatial effect. Hastie and Tibshirani [1990] call it "multicollinearity in non-linear models". Nowadays, the term "concurvity" [Hastie and Tibshirani, 1990; Guisan et al., 2002; He, 2004] is more common. The impact of concurvity, e.g. on the parameter estimates, has not been investigated completely [He, 2004].

A theoretical overview over generalized additive models and the extension to geo-additive models is shown in 2. Furthermore, Boosting is presented as a powerful machine learning technique for model estimation. In addition, this chapter demonstrates the alternative method pGAM [Gu et al., 2010] to deal with concurvity.

This thesis gives an idea to solve the dilemma of concurvity. The spatial effect is modified so that it can only reproduce a variability which cannot be explained by covariate information in principle. This is done with the help of the new %ll%-operator. Chapter 3 presents this idea and the %ll%-operator in detail and shows an implementation for the R–mboost package [Hothorn et al., 2009].

Chapter 4 investigates with the help of three simulation studies the performance of the new %ll%-operator. Furthermore, the %ll%-operator is compared to a "standard" generalized additive boosting-model and the alternative pGAM-model.

Finally, the new %ll%-operator is used in an ecological application practically. Chapter 5 analyzes the impact of climate change on the tree population in Bavaria.

2. Theory

This chapter presents the basic statistical frameworks for flexible specification and the corresponding models.

2.1. Generalized Additive Models

The basic aim of additive models is the flexible modeling of the relation between dependent and independent variables. The additive model extends a simple linear model

$$\begin{aligned} y_i &= f_1(z_{i1}) + \ldots + f_q(z_{iq}) + \beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik} + \varepsilon_i \\ &= f_1(z_{i1}) + \ldots + f_q(z_{iq}) + \eta_i^{\text{lin}} + \varepsilon_i \\ &= \eta_i^{\text{add}} + \varepsilon_i \end{aligned}$$

where $f_1(z_{i1}), \ldots, f_q(z_{iq})$ are smooth functions of the covariates $z_1, \ldots, z_q$. These functions are estimated in a nonparametric fashion. A generalized additive model differs from an additive model. Its additive predictor is linked with the expected value by a known smooth monotonic link-function.

The smooth functions $f_1(z_{i1}), \ldots, f_q(z_{iq})$ are represented by a linear combination of basic-functions

$$f_j = \sum_{l=1}^{d_j} \gamma_{jl} B_l(z_j), \quad j = 1 \ldots q$$

There are different types of basic-functions for $B_l,\ l = 1 \ldots d_j$. Common examples are B-Splines or TP-Splines. Section 2.3 focuses on B-Splines.

A covariate can always be represented by

$$\boldsymbol{f}_j = \boldsymbol{Z}_j \boldsymbol{\gamma}_j$$

with the coefficient vector $\boldsymbol{\gamma}_j = (\gamma_1 \ldots \gamma_j)$ and the design matrix $\mathbf{Z}_j$. The additive model in matrix notation

$$\boldsymbol{y} = \boldsymbol{Z}_1 \gamma_1 + \ldots + \mathbf{Z}_q \boldsymbol{\gamma}_q + \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

The estimation occurs either with the penalized least squares criterion for normal distributed response

$$PKQ(\lambda) = (\boldsymbol{y} - \boldsymbol{Z}\boldsymbol{\gamma})^T (\boldsymbol{y} - \boldsymbol{Z}\boldsymbol{\gamma}) + \lambda \boldsymbol{\gamma}^T \boldsymbol{K} \gamma.$$

Thereby denotes $\boldsymbol{Z}$ a matrix whose entries are the basic-functions evaluated at the observations

$$\mathbf{Z} = \begin{pmatrix} B_1^l(z_1) & \ldots & B_d^l(z_1) \\ \vdots & & \vdots \\ B_1^l(z_n) & \ldots & B_d^l(z_n) \end{pmatrix}.$$

Simple GAMs are estimated with the penalized least-squares estimator

$$\hat{\gamma} = (\boldsymbol{Z}^T \boldsymbol{Z} + \lambda \boldsymbol{K})^{-1} \boldsymbol{Z}^T \boldsymbol{y}.$$

or with the Fisher Scoring algorithm [Fahrmeir et al., 2009]. Generalized additive models require more complex methods as the backfitting algorithm [Hastie and Tibshirani, 1990]. For a more detailed overview of generalized additive models see Hastie and Tibshirani [1990] and Fahrmeir et al. [2009].

2.2. Geo-Additive Models

Geo-additive models enlarge the predictor of additive models with an additional spatial effect $f_{\text{geo}}(s)$ [Fahrmeir et al., 2009]. A geo-additive model is represented by

$$\begin{aligned} y_i &= \eta_i^{\text{add}} + f_{\text{geo}}(s_i) + \varepsilon_i \\ &= f_1(z_{i1}) + \ldots + f_q(z_{iq}) + f_{\text{geo}}(s_i) + \mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i \end{aligned}$$

with $i = 1 \ldots n$ and $\boldsymbol{x}_i^T \boldsymbol{\beta} = \beta_0 + \beta_1 x_{i1} + \beta_k x_{ik}$. The given assumptions in the additive model are also valid for the covariates x_i, z_i and the error term ε_i. The spatial effect $f_{geo}(s)$ can be used as a surrogate for unobserved and undocumented covariates consisting of spatial information. Common estimation techniques are two-dimensional smoothing splines, for example tensorproduct P-Splines or Markov random-fields.

Especially, for discrete $s \in \{1, \ldots, d\}$ Markov random-fields are used for estimation. Thereby, rewrite the vector $\boldsymbol{f}_{\text{geo}}(s) = (f_{\text{geo}}(s_1), \ldots, f_{\text{geo}}(s_n))^T$ of the spatial effect by

$$\boldsymbol{f}_{\text{geo}} = \boldsymbol{Z}_{\text{geo}} \boldsymbol{\gamma}_{\text{geo}} \tag{2.1}$$

with $\boldsymbol{\gamma}_{\text{geo}} = (\gamma_{\text{geo},1}, \ldots, \gamma_{\text{geo},n})^T$ for the spatial effect and a $n \times d$ design matrix $\boldsymbol{Z}_{\text{geo}}$. The design matrix $\boldsymbol{Z}_{\text{geo}}$ (with $\boldsymbol{Z}_{\text{geo}}[i, s] = 1$ if $s_i = s$ and 0 else) is called incidence-matrix because of its special design. Further details on $\boldsymbol{Z}_{\text{geo}}$ and $\boldsymbol{\gamma}_{\text{geo}}$ are presented in Fahrmeir et al. [2009].

It is possible to rewrite $\boldsymbol{f}_{\text{geo}}(s)$ for continuous s, too. It can also be done in the way shown in the equation (2.1). Usually, the estimation occurs with tensorproduct P-Splines. The design of $\boldsymbol{Z}_{\text{geo}}$ and $\boldsymbol{\gamma}_{\text{geo}}$ is presented in Fahrmeir et al. [2009] and section 2.3.2.
The final geo-additive model can be written as

$$\boldsymbol{y} = \boldsymbol{Z}_1 \boldsymbol{\gamma}_1 + \ldots + \boldsymbol{Z}_q \boldsymbol{\gamma}_q + \boldsymbol{Z}_{\text{geo}} \boldsymbol{\gamma}_{\text{geo}} + \boldsymbol{X} \boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

The estimation of $\boldsymbol{\gamma}_{\text{geo}}$ is regularized the same way as the estimation of the coefficient vector $\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_q$ with a penalty $\lambda_{\text{geo}} \boldsymbol{\gamma}_{\text{geo}}^T \boldsymbol{K}_{\text{geo}} \boldsymbol{\gamma}_{\text{geo}}$ or equivalent Gauss priors.

2.3. Splines

This section outlines basic statistical methods for nonparametric modeling. Therefore, their key concepts will be revealed.

2.3.1. B(asic)-Splines Basis functions

B(asic)-Splines are a flexibly modeling strategy to describe the influence of a continuous variable with good numerical properties [Fahrmeir et al., 2009]. The function $f(z)$ is approximated by piecewise polynomials. There are additional smoothness requirements at the knots of the function $f(z)$. B-Splines basis functions are constructed in a way that the polynomial pieces with the favored degree are sufficiently smooth at a desired knot. A B-Spline basis function consists of $(l + 1)$ polynomial pieces with degree l, which are composed of $l - 1$ times continuously differentiable. Figure 2.1 illustrates single B-Spline basis function of degree $l = 0, 1, 2, 3$ on equidistant knots as the results from

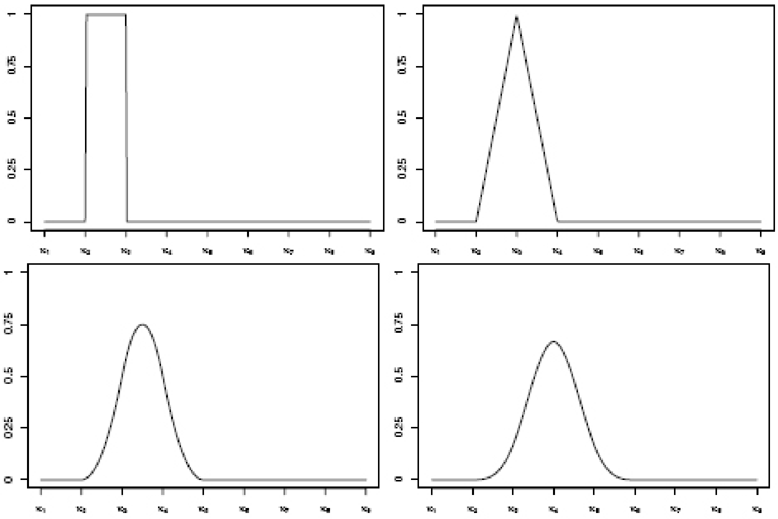

Figure 2.1.: One single B(asic)-Spline basis function of degree $l = 0, 1, 2, 3$ at equidistant knots illustrated by [Fahrmeir et al., 2009].

these considerations.

All B-Spline basis functions are built for visualization of polynomial splines based on the underlying knots. The complete B-Spline basis of degree $l = 0, 1, 2, 3$ are depicted at equidistant knots in figure 2.2.

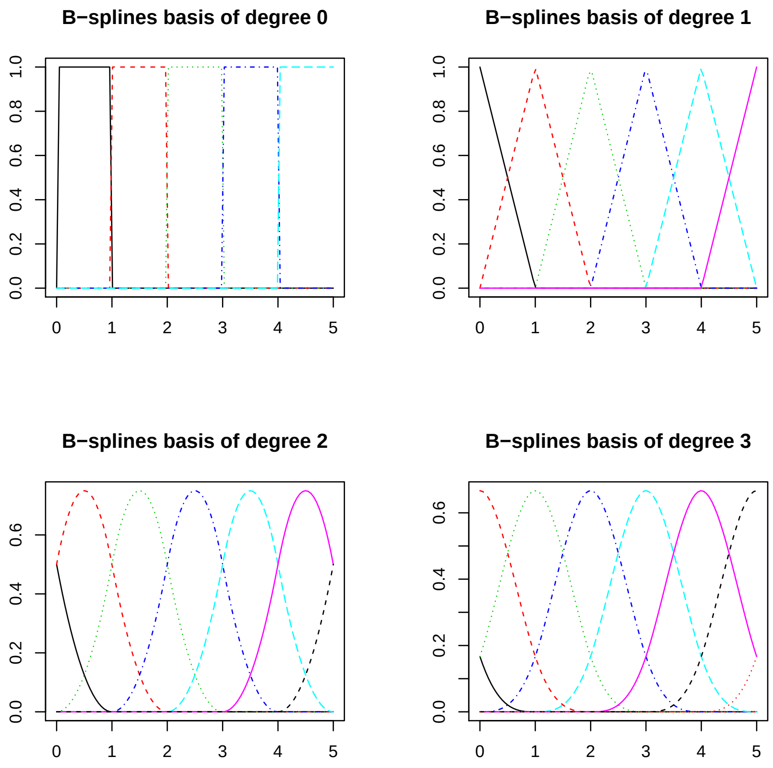

Figure 2.2.: B(asic)-Spline basis function of degree $l = 0, 1, 2, 3$ at equidistant knots.

With the help of this basis it is possible to represent $f(z)$ by a linear combination with $d = m + l - 1$ basis functions

$$f(z) = \sum_{j=1}^{d} \gamma_j B_j(z)$$

A great benefit of the B-Spline basis is based on its local precision. Contrary to other basis functions, B-Spline basis functions are only over $l + 2$ adjacent knots different from zero. Additionally they are bounded above in order to antagonize numerical problems.

For B-Splines of degree $l = 0$ the definition could be derived directly from figure 2.1

$$B_j^0(z) = \mathbb{1}_{[\kappa_j,\kappa_{j+1})}(z) = \begin{cases} 1 & \kappa_j \leq \kappa_{j+1}, \\ 0 & else, \end{cases} \qquad j = 1, \ldots, d-1.$$

In the shown case the equivalence to the spline representation with Truncated Power Series Basis (TP-Basis) is easy to see. Building up successive differences of the TP-Basis function of degree $l = 0$ leads to the B-Spline-Basis which is constantly over an interval defined by two adjacent knots. B-Splines of degree $l \geq 1$ are recursively represented by

$$B_j^l(z) = \frac{z - \kappa_j}{\kappa_{j+l} - \kappa_j} B_j^{l-1}(z) + \frac{\kappa_{j+l+1} - z}{\kappa_{j+l+1} - \kappa_{j+1}} B_{j+1}^{l-1}(z)$$

and means that the basis function consists of two linear pieces on the intervals $[\kappa_j, \kappa_{j+l+1})$ and $[\kappa_{j+l+1}, \kappa_{j+2})$.

2.3.2. P(enalized)-Splines

Univariate P(enalized)-Splines

The performance of a non-parametrical function estimation based on polynomial splines depends strongly on the number and location of the used knots [Fahrmeir et al., 2009]. A common solution for this problem is to work with penalty approaches [Eilers and Marx, 1996]. The fundamental idea of penalty approaches is to approximate the function $f(z)$ in order to be estimated by a polynomial spline with an adequate great number of knots. In addition, a penalty term is introduced penalizing large estimation variability. The penalty term is regularly based on first or second differences and is represented by

$$\begin{aligned} \lambda \sum_{j=k+1}^{d} (\delta^k \gamma_j)^2 &= \lambda \boldsymbol{\gamma}^T \boldsymbol{D}_k^T \boldsymbol{D}_k \boldsymbol{\gamma} \\ &= \lambda \boldsymbol{\gamma}^T \boldsymbol{K}_k \boldsymbol{\gamma} \end{aligned}$$

with $\boldsymbol{K}_1$ the penalty matrix for the first differences

$$\boldsymbol{K}_1 = \begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{pmatrix},$$

respectively $\boldsymbol{K}_2$ the penalty matrix for the second differences

$$\boldsymbol{K}_2 = \begin{pmatrix} 1 & -2 & 1 & & & & \\ -2 & 5 & -4 & 1 & & & \\ 1 & -4 & 6 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 6 & -4 & 1 \\ & & & 1 & -4 & 5 & -2 \\ & & & & 1 & -2 & 1 \end{pmatrix}.$$

Bivariate P(enalized)-Splines

Fahrmeir et al. [2009] propose to use bivariate P(enalized)-Splines to model spatial effects. Suppose z_1 and z_2 are coordinates of a two-dimensional surface $f(z_1, z_2)$ in a spatial model. Firstly, the univariate basis for z_1 and z_2 are built and provide the basis functions $B_j^{(1)}(z_1), j = 1, \dots, d_1$ and $B_j^{(2)}(z_2), j = 1, \dots, d_2$. Finally, the tensorproduct-basis is built on all these basis functions

$$B_{jk}(z_1, z_2) = B_j^{(1)}(z_1) \cdot B_k^{(2)}(z_2), \qquad j = 1, \dots, d_1,\ k = 1, \dots, d_2$$

It leads to the following representation for $f(z_1, z_2)$:

$$f(z_1, z_2) = \sum_{j=1}^{d_1} \sum_{k=1}^{d_2} \gamma_{jk} B_{jk}(z_1, z_2)$$

Tensorproduct-B-Splines consist of good numerical properties. They are displayed in figure 2.3 for different spline degrees $l = 0, 1, 2, 3$. Figure 2.3 shows that a greater smoothness causes a higher spline degree. Figure 2.4 focuses on

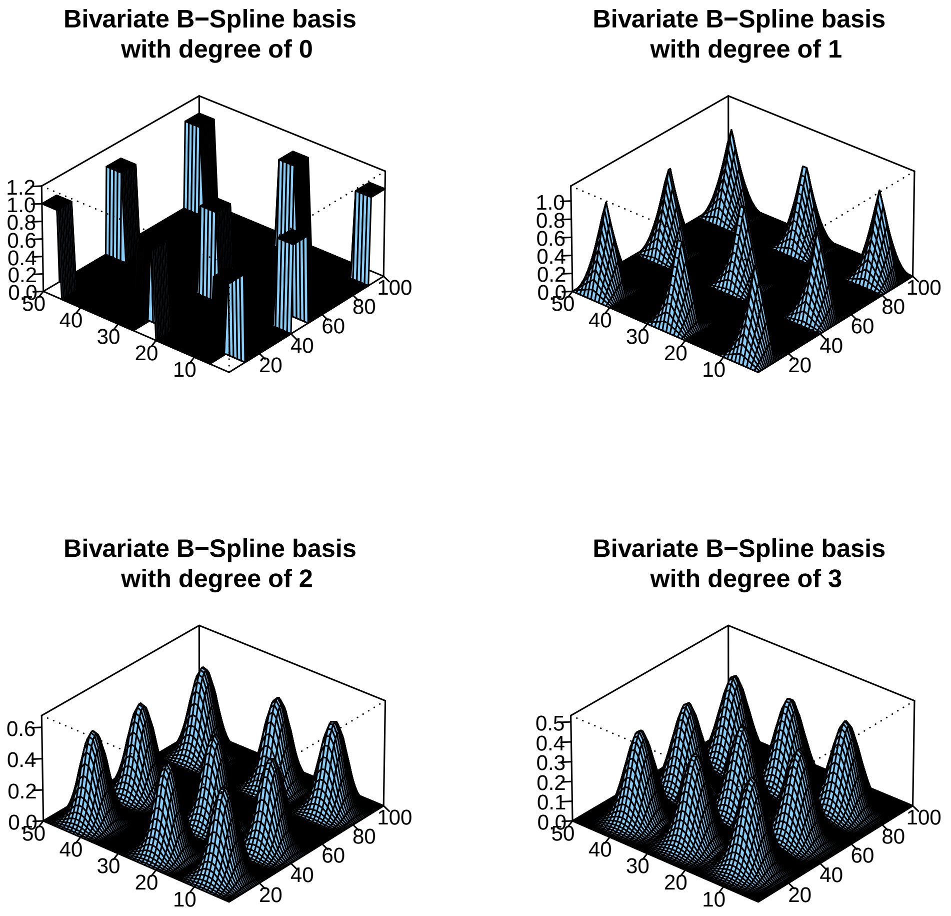

Figure 2.3.: Tensorproduct basis functions based on univariate B-Splines of degree $l = 0, 1, 2, 3$.

the contourplots of the Tensorproduct-B-Splines. The level curves differ clearly from a circle. Therefore, Tensorproduct-B-Splines are not radial. Beyond that,

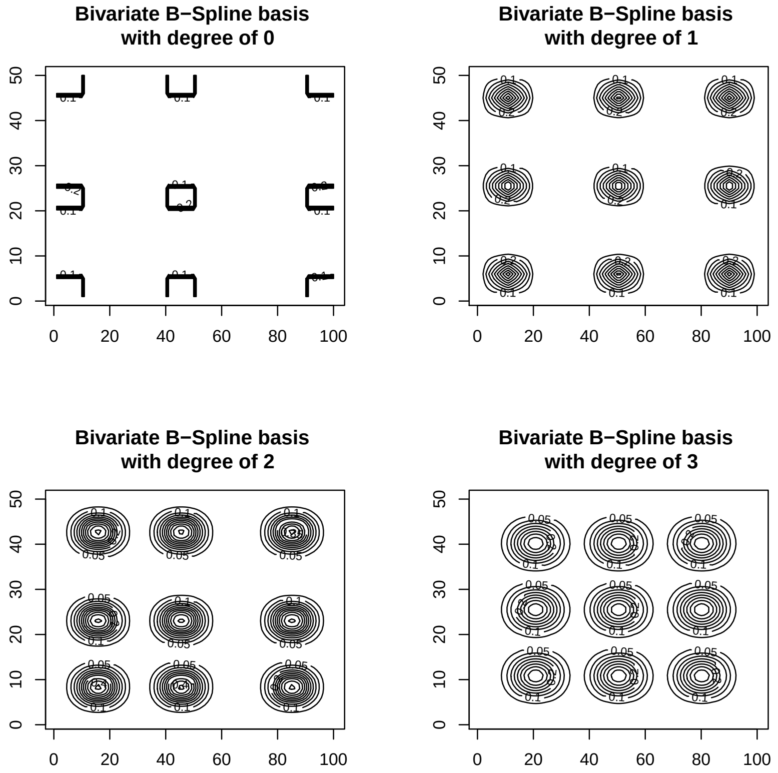

Figure 2.4.: Contourplots to Tensorproduct-B-Splines basis functions of degree $l = 0, 1, 2, 3$.

the choice of the optimal number and location of knots is as important as in the univariate case. In the bivariate case admittedly it is often the problem that there are certain regions without any observations. Consequently, it is not possible to estimate basis functions lying in this area. These problems are solved with the help of regularization with a penalty term.

Firstly, there is the introduction of an adequate penalty term. It makes sense to use the spatial design of the basis functions and the regression coefficients. In the univariate case the penalty term is based on squared differences. To assign this concept to the two-dimensional case, the proper spatial neighbors must be defined first.

Thus figure 2.5 shows possible spatial neighbors for four, eight and twelve neighbors. Firstly, assume a simple neighborhood with four nearest neighbors. This situation is shown in the left part of figure 2.5. It makes sense to use a penalty term based on the squared differences between γ_{jk} and its neighbors. $\boldsymbol{D_1}$ and $\boldsymbol{D_2}$ are the univariate difference matrices in z_1 and z_2 direction. The row wise first differences can be calculated by applying the difference matrix $\boldsymbol{I}_{d_2} \otimes \boldsymbol{D_1}$ on the vector $\boldsymbol{\gamma}$. Thereby, $\boldsymbol{I}_d$ denotes the d-dimensional identity matrix and $\otimes$ the Kronecker product. Hence,

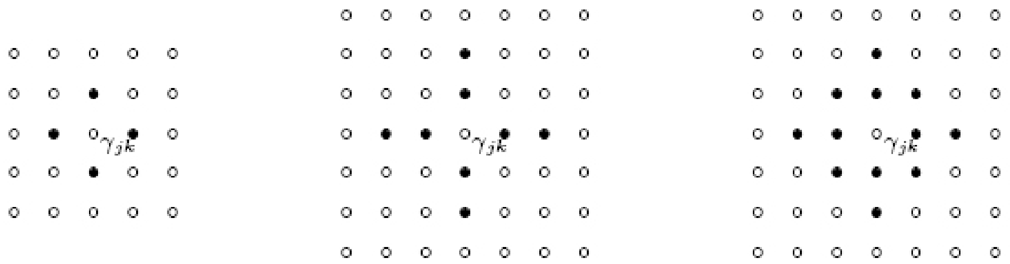

Figure 2.5.: Fahrmeir et al. [2009] illustrate the spatial neighborhood on a regular grid: Neighbors to coefficient γ_{jk} are marked by a black point.

$$\boldsymbol{\gamma}^T(\boldsymbol{I}_{d_2} \otimes \boldsymbol{D}_1)^T(\boldsymbol{I}_{d_2} \otimes \boldsymbol{D}_1)\boldsymbol{\gamma} = \sum_{k=1}^{d_2}\sum_{j=2}^{d_1}(\gamma_{jk} - \gamma_{j-1,k})^2$$

the sum is formed of all row wise squared differences. The column-wise squared differences are formed analogue

$$\boldsymbol{\gamma}^T(\boldsymbol{D}_2 \otimes \boldsymbol{I}_{d_2})^T(\boldsymbol{D}_2 \otimes \boldsymbol{I}_{d_2})\boldsymbol{\gamma} = \sum_{j=1}^{d_1}\sum_{k=2}^{d_2}(\gamma_{jk} - \gamma_{j,k-1})^2.$$

Finally, the penalty term consists of the added up and squared column-wise and row wise differences

$$\lambda\boldsymbol{\gamma}^T\boldsymbol{K}\boldsymbol{\gamma} = \lambda\boldsymbol{\gamma}^T[(\boldsymbol{I}_{d_2} \otimes \boldsymbol{D}_1)^T(\boldsymbol{I}_{d_2} \otimes \boldsymbol{D}_1)\boldsymbol{\gamma} + \boldsymbol{\gamma}^T(\boldsymbol{D}_2 \otimes \boldsymbol{I}_{d_2})^T(\boldsymbol{D}_2 \otimes \boldsymbol{I}_{d_2})]\boldsymbol{\gamma}.$$

Generally, the penalty term can be built with the help of univariate penalty matrices and the Kronecker product

$$\boldsymbol{K} = \boldsymbol{I}_{d_2} \otimes \boldsymbol{K}_1 + \boldsymbol{K}_2 \otimes \boldsymbol{I}_{d_1}.$$

This leads to a quadratic penalty term

$$\lambda \boldsymbol{\gamma}^T \boldsymbol{K} \boldsymbol{\gamma} \quad = \quad \lambda \boldsymbol{\gamma}^T [\boldsymbol{I}_{d_2} \otimes \boldsymbol{K}_1 + \boldsymbol{K}_2 \otimes \boldsymbol{I}_{d_1}] \gamma$$

where $\boldsymbol{K}_1 = \boldsymbol{D}_1^T \boldsymbol{D}_1$ and $\boldsymbol{K}_2 = \boldsymbol{D}_2^T \boldsymbol{D}_2$ are univariate penalty matrices.

Optimal choice of the smoothness parameter λ

The optimal choice of the smoothness parameter λ is an important aspect. The smoothness parameter λ controls the smoothness of estimated functions and ensures a suitable compromise between bias and variability of an estimator. For $\lambda \rightarrow \infty$ exists a widely linear estimation of the function $f(z)$. Contrary to $\lambda \rightarrow 0$ exists a quite rough estimation of the function $f(z)$.

The problem occurs that bias and variability of a smoothness method are simultaneously depended on the smoothness parameter λ and both cannot be minimized at the same time. Therefore, a suitable equalization must be found.

On the one hand, the Mean Squared Error (MSE) is a good possibility:

$$\begin{aligned} MSE(\hat{f}(z)) &= \mathbb{E}\left[\left(\hat{f}(z) - f(z)\right)^2\right] \\ &= \underbrace{\left(\mathbb{E}\left[\hat{f}(z) - f(z)\right]\right)^2}_{\text{bias}} + \underbrace{Var(\hat{f}(z))}_{\text{variability}}. \end{aligned}$$

The MSE is added additively by the squared bias and the variance. Finally, the λ is taken where the MSE is minimal.

On the other hand, there is the Cross-Validation (CV) to find the optimal smoothness parameter λ. Respectively one observation is deleted in cross validation. Within the next step the smoothness parameter λ is estimated with the remaining $n-1$ observations. Finally, $f(z_i)$ is predicted for the deleted observation. Denoted by $\hat{f}^{(-i)}(z_i)$ is the estimation which occurs without the observation (z_i, y_i) and receives Cross-Validation criterion [Stone, 1974]:

$$CV = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}^{(-i)}(z_i)\right)^2 .$$

The minimization of the CV criterion leads in the sense of prediction error to an optimal λ.

A further alternative method to achieve the optimal smoothness parameter λ is by the Akaikes Information Criterion (AIC) [Akaike, 1974]:

$$AIC = n \cdot \log(\hat{\sigma}^2) + 2(\mathrm{df} + 1)$$

where $\hat{\sigma}^2 = \sum(y_i - \hat{f}(z_i))^2/n$. The AIC has to be minimized concerning the smoothness parameter.

2.4. Boosting

"A horse-racing gambler, hoping to maximize his winnings, decides to create a computer program that will accurately predict the winner of a horse race based on the usual information (number of races recently won by each horse, betting odds for each horse, etc.). To create such a program, he asks a highly successful expert gambler to explain his betting strategy.
Not surprisingly, the expert is unable to articulate a grand set of rules for selecting a horse. On the other hand, when presented with the data for a specific set of races, the expert has no trouble coming up with a "rule of thumb" for that set of races (such as, "Bet on the horse that has recently won the most races" or "Bet on the horse with the most favored odds").
Although such a rule of thumb, by itself, is obviously very rough and inaccurate, it is not unreasonable to expect it to provide predictions that are at least a little bit better than random guessing.
Furthermore, by repeatedly asking the expert's opinion on different collections

of races, the gambler is able to extract many rules of thumb.
In order to use these rules of thumb to maximum advantage, there are two problems faced by the gambler:
First, how should he choose the collections of races presented to the expert so as to extract rules of thumb from the expert that will be the most useful?
Second, once he has collected many rules of thumb, how can they be combined into a single, highly accurate prediction rule?[1]
An answer to this question is given by Schapire [1990] and Bühlmann and Yu [2003] with their boosting algorithm. Due to disadvantages in spatial application, an extension of this algorithm is presented in section 3.3.3. Firstly, an important variation of the boosting algorithm is introduced in section 2.4.1.

2.4.1. Boosting Algorithm

The expression "Boosting" signifies a series of algorithms that improve the power of several "weak" learners (called in the following "base-learner") by combining them to an ensemble ("to boost"). The benefit of such an ensemble was shown by Kearns and Valiant [1994] for the first time. The corner stone was laid by Schapire [1990] with his paper "the strength of weak learnability". The first step towards practical application was done by Breiman [1998, 1999] with his today well known AdaBoost algorithm. These first Boosting-Algorithms were one of the most powerful machine learning technique used in the last twenty years for binary outcomes [Schapire, 1990; Freund and Schapire, 1995]. Breiman [1998, 1999] was able to imbed this algorithm in statistical framework by considering AdaBoost as a steepest descent algorithm in function space.
Friedman et al. [2000] and Bühlmann and Yu [2003] derived the general statistical framework which yields a direct interpretation of boosting as a method for function estimation. Nowadays, Boosting is a method to optimize prediction accuracy and to obtain statistical model estimates via gradient descent techniques.

[1] Freund et al. [1999], page 771

An optimal prediction of $\boldsymbol{y}$ with the help of the covariates $\boldsymbol{x}_i = (x_{i1}, \dots, x_{ip})^T$, $i = 1, \dots, n$ is the major aim. The covariates are linked to response variable $\boldsymbol{y}$ as described in section 2.1 or 2.2.

$$g(\mathbb{E}[y_i \mid \boldsymbol{x}_i^T]) = \eta_i = \sum_{j=1}^{p} f_j(x_j) = \sum_{l=1}^{L} b_l(\cdot) \tag{2.2}$$

Thereby, every smooth effect $f_j(x_j)$ is represented by a base-learner $b_l(\cdot)$. The major boosting challenge is to minimize an expected loss function $\mathbb{E}\left[\rho(\cdot,\cdot)\right]$ relating to a real-valued function η.

$$\hat{\boldsymbol{\eta}} := \underset{\boldsymbol{\eta}}{\operatorname{argmin}}\, \mathbb{E}_{Y,X}\left[\rho(\boldsymbol{y}, \eta(\boldsymbol{x}))\right] \tag{2.3}$$

A typical assumption to the loss function $\rho(\cdot,\cdot)$ is to be differentiable and convex with respect to $\eta(\cdot)$. Different loss functions are described in detail in section 2.4.4. Usually, the L_2-loss is used.

In general, the expected mean in equation (2.3) is unknown in practice. Thus, replace $\mathbb{E}\left[\rho(\cdot,\cdot)\right]$ with the empirical risk

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

for estimation of $\hat{\boldsymbol{\eta}}$ and apply iterative steepest descent in function space. The minimization of $\mathcal{R}$ as a function of $\eta(\cdot)$ maximizes the empirical log-likelihood corresponds to $\eta(\cdot)$ [Bühlmann and Hothorn, 2007]. The minimization is done step by step in direction towards the steepest descent of the loss function. Instead of using the original data, the boosting algorithm uses the derivative of the loss function on the covariates in every iteration $m = 1, \dots m_{stop}$. This results that poorly predicted values get a very high weight in the following iteration. Reaching the minimum is done by adding up by a step-length factor ν compressed value to the previous value of $\eta(\cdot)$ in each iteration. The following algorithm was developed by Friedman [2001]. The illustration of the algorithm is based on Hofner [2011].

Component-wise Gradient Boosting Algorithm

Initialization:

Set $m = 0$. Initialize the function estimate $\hat{\boldsymbol{\eta}}^{[0]}(\cdot)$ with an offset value. Usual choices are

$$\hat{\boldsymbol{\eta}}^{[0]} = \underset{c}{\operatorname{argmin}}\, n^{-1} \sum_{i=1}^{n} \rho(y_i, c)$$

or

$$\hat{\boldsymbol{\eta}}^{[0]} \equiv 0$$

Iterate:

(1) **Negative gradient vector:**

First increase m by 1. Then compute the negative gradient of the loss function $\rho(y_i, \eta(\boldsymbol{x}_i^T))$ and evaluate the function values of the previous iteration $\hat{\boldsymbol{\eta}}^{[m-1]}(\boldsymbol{x}_i^T)$. This leads to the negative gradient vector:

$$u_i^{[m]} = -\frac{\partial}{\partial \eta} \rho(y_i, \eta) \mid_{\eta = \hat{\eta}^{[m-1]}(x_i^T)}, \quad i = 1, \ldots, n$$

(2) **Estimation:**

Fit the negative gradient vector $\boldsymbol{u}^{[m]} = \left(u_1^{[m]}, \ldots, u_n^{[m]}\right)$ to $x_1, \ldots, x_n$ by regressing the L base-learners b_{l^*} separately on $u^{[m]}$:

$$(x_i, u_i)_{i=1}^n \overset{\text{base procedure}}{\rightarrow} \hat{b}_l^{[m]}(\cdot).$$

After the evaluation of all base-learner choose those with the highest goodness of fit. That means, choosing the base-leaner b_{l^*} which minimizes the residual sum of squares:

$$l^* = \underset{1 \leq l \leq L}{\operatorname{argmin}} \sum_{i=1}^{n} \left(u_i^{[m]} - \hat{b}_l^{[m]}(x_i^T)\right)^2.$$

(3) **Update:**

Update the function estimate

$$\hat{\boldsymbol{\eta}}^{[m]}(\cdot) = \hat{\boldsymbol{\eta}}^{[m-1]}(\cdot) + \nu \cdot \hat{b}_{l^*}^{[m]}(\cdot)$$

and the actual partial effect j^*, containing the base-learner l^*:

$$\hat{f}_{j^*}^{[m]}(\cdot) = \hat{f}_{j^*}^{[m-1]}(\cdot) + \nu \cdot \hat{b}_{l^*}^{[m]}(\cdot)$$

where $0 < \nu \leq 1$ is a step-length factor. The estimates of all other functions $\hat{f}_j$, $j \neq j^*$ remain unchanged.

Stopping rule: Iterate steps (2) to (4) until $m = m_{\text{stop}}$ for a given stopping iteration m_{stop}.

The algorithm above consists of two important tuning parameters. A detail description of them is given in section 2.4.3.

2.4.2. Choosing Base-Learners

The structural assumption of the model, especially the types of effects that are used can be specified in terms of base-learners. Therefore, boosting is a component-wise iterative process which selects just one base-learner (one component) in each iteration. However, each base-learner can be selected more often and results in a related type of effect. The fit of the data is improved by attempting the vector of the residuals $\boldsymbol{u}^{[m]}$ by the most appropriate base-learner in every iteration. For example, a base-learner can be either a linear or a smooth effect. The estimation of the base-learners occurs with penalized least squares

$$\hat{b}_j = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{K})^{-1}\boldsymbol{X}^T\boldsymbol{u}$$

with the design matrix X, the penalty matrix K and the smoothness parameter λ for example presented by Fahrmeir et al. [2009]. The smoothing parameter λ and degrees of freedom (`df`) have a one-to-one relationship and regulate the amount of penalization. Using the component-wise boosting algorithm naturally leads to variable and model selection. Nevertheless, the selection of base-learners in each iteration can be biased if the competing base-learners have different degrees of flexibility [Hofner, 2011; Hofner et al., 2011]. Consequently, boosting (almost) always prefers smooth base-learner over linear base-learner no matter of the true effect. The smooth base-learner offers much more flexibility and typically incorporates a linear effect as a special case [Hofner et al., 2011]. Hence, Hofner [2011] proposes to specify equal degrees of freedom for all base-learners if unbiased model and variable selection are the goal.

Different modeling variations are determined by representing each partial effect of the equation (2.2) by one or several base-learners $l = 1, \ldots, L_j$:

$$\hat{f}_j^{[m_{\text{stop}}]} = \sum_{l=1}^{L_j} \sum_{m=1}^{m_{\text{stop}}} \eta \cdot \hat{b}_{j,l}^{[m]}.$$

The additive structure of equation (2.2) is preserved. The appendix of Maloney et al. [2011] presents extensive explanations. As base-learner, linear and categorical effects, interactions, one-and two-dimensional splines, random effects and much more can be used. Hofner [2011] gives an overview in general. Section 3.3.3 shows the relevant base-learner for this thesis.

2.4.3. Tuning Parameter in the Boosting Algorithm

The stopping iteration m_{stop} is the main tuning parameter of boosting algorithm. In order to avoid overfitting, boosting algorithms should not run until complete convergence [Bühlmann and Hothorn, 2007]. Therefore, an optimal choice of the stopping iteration m_{stop} is very important. The optimal m_{stop} is usually chosen by an information criterion, for example AIC [Akaike, 1974], cross-validation [Stone, 1974] or bootstrap [Efron, 1979].

The step-length factor ν is of minor importance, as long as it is "small", for example $\nu = 0.1$. Bühlmann and Hothorn [2007] suppose that a small value of ν typically requires a larger number of boosting iterations and more computing time. Thus, the step-length factor ν and the optimal stopping iteration m_{stop} influence each other. As long as the shrinkage effect of ν is used, the overfitting proceeds relatively slowly [Bühlmann and Hothorn, 2007]. Friedman [2001] proofed empirically that predictive accuracy is potentially better and almost never worse when choosing ν "sufficiently small" because the estimate of $\hat{\eta}(\cdot)$ are shrunken towards zero [Friedman, 2001]. Small values ensure that the boosting algorithm does not fail the minimum of the empirical risk $\mathcal{R}$. In addition, shrinkage generally stabilizes the effect estimations and avoids multicollinearity problems [Hofner, 2011; Friedman, 2001].

2.4.4. Loss Functions and Boosting Algorithm

As mentioned before, the structural component of the boosting models is determined by the base-learners. The stochastic component of the model is defined by the loss function. Large numbers of boosting algorithms can be defined by specifying different loss functions $\rho(\cdot,\cdot)$. There are different options for the regression setting with response $\boldsymbol{y}$. Usually, for GAMs the loss function is simply the negative log-likelihood function of the outcome distribution. Therefore, in the following section several options are briefly discussed for choosing the loss.

The use of the normal distribution leads to the special case L_2Boosting. Most often the squared error loss, also called L_2-loss

$$\rho_{L_2}(\boldsymbol{y}, \eta(\boldsymbol{x})) = \frac{1}{2} \left| \boldsymbol{y} - \eta(\boldsymbol{x}) \right|^2$$

is used. The loss function is scaled by the factor $\frac{1}{2}$ to confirm a helpful representation of its first derivative, namely simply the residuals. By modeling the residuals, the boosting algorithm focuses on the "difficult" observations which were previously estimated poorly [Hofner, 2011].

A loss function with some robustness properties is the absolute-error-loss or L_1-loss and is represented by

$$\rho_{L_1}(\boldsymbol{y}, \eta(\boldsymbol{x})) = \left| \boldsymbol{y} - \eta(\boldsymbol{x}) \right| .$$

The L_1-loss is not differentiable at the point $y = \eta$. However, it is possible to compute the partial derivatives because the single point $y = \eta$ has the probability zero to be realized by the data.

The Huber-loss function is a compromise between the L_1 and the L_2 loss

$$\rho_{Huber}(\boldsymbol{y}, \eta(\boldsymbol{x})) = \begin{cases} \left|\boldsymbol{y} - \eta(\boldsymbol{x})\right|^2 /2, & \text{if } \ |y - \eta(\boldsymbol{x})| \leq \delta \\ \delta(|\boldsymbol{y} - \eta(\boldsymbol{x})| - \delta/2), & \text{if } \ |y - \eta(\boldsymbol{x})| > \delta \end{cases}$$

δ is chosen adaptively. A strategy for choosing δ is proposed by Friedman [2001]

$$\delta_m = \text{median}\left(\{|y_i - \hat{f}^{[m-1]}(\boldsymbol{x}_i^T)\ |; i = 1, \ldots, n|\}\right)$$

where the previous fit $\hat{\eta}^{[m-1]}(\cdot)$ is used.

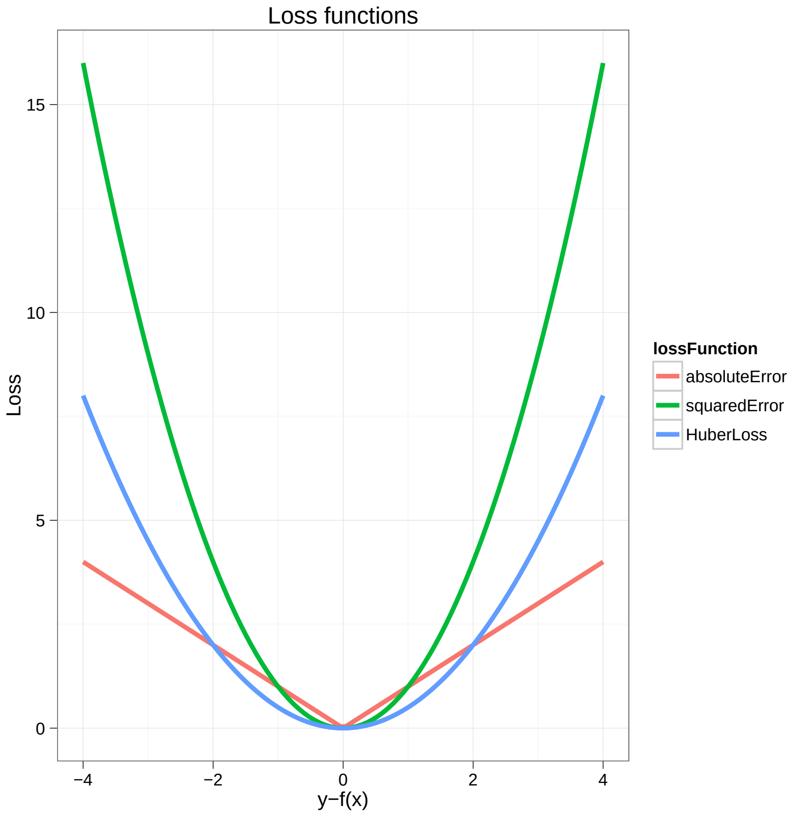

Figure 2.6.: Comparison of three different loss functions: L_1-loss function (red), L_2-loss function (green), Huber-loss function (blue).

Figure 2.6 compares the three different presented loss-functions. The L_2-loss function (green line) penalizes observations with large absolute residuals stronger than the two other loss functions. Contrary, the L_1-loss function (red line) penalizes extreme margins linearly. The Huber-loss function (blue line) can be seen as a compromise between the L_1- and L_2-loss function.

2.5. Partial Generalized Additive Models

Gu et al. [2010] developed an information-theoretical approach for dealing with concurvity and selecting variables. This new procedure is based on the mutual information (MI) and is called partial generalized additive model (pGAM).

The partial generalized additive model is able to make not only predictions but also to identify which covariates are important and how these covariates affect the response variable.
Despite of concurvity, the partial generalized additive model is able to produce stable and correct estimates of the covariates' functional effects. This happens by building a GAM (chapter 2.1) on a selected set of transformed variables. It is explained how the transformation works in detail.

Consider the standard GAM model

$$\mathbb{E}(\boldsymbol{y} \mid \boldsymbol{x}) = g(\eta(\boldsymbol{x})) = g(f_0 + f_1(\boldsymbol{x}_1) + \ldots + f_p(\boldsymbol{x}_p) \tag{2.4}$$

where g is a monotonic link function and $f_j(\boldsymbol{x})$; $j = 1, \ldots, p$ are unspecified smooth functions which allows a simple interpretation of the covariates' functional effects. If a strong functional relationship among the covariates exist, which is also known as concurvity, problems will arise [Gu et al., 2010]. The problem of concurvity is introduced in detail in chapter 3.

To solute the problem of concurvity, the new pGAM-procedure goes back to the modified backfitting algorithm given by Hastie and Tibshirani [1990] which *partially* deals with concurvity. The basic idea is to separate each smoothing operator into a projection part and a shrinking part. Afterwards all projection parts are combined into one large projection part and only to use backfitting for the shrinkage part. Hastie and Tibshirani [1990] proved that concurvity occurs only in the projection part. Thus, the modified backfitting algorithm allows to deal alone with concurvity in the projection step [Gu et al., 2010].

2.5.1. Methodology

Brief Review of Mutual Information

Mutual information (MI) is an important component of the pGAM algorithm and was first introduced by Shannon et al. [1948]. It is used to measure the dependence between two random variables $\boldsymbol{x}$ and $\boldsymbol{y}$. The MI is defined as

$$MI(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{E}\left(\log \frac{f(\boldsymbol{x}, \boldsymbol{y})}{f_x(\boldsymbol{x}) f_y(\boldsymbol{y})}\right) \tag{2.5}$$

where f, f_x and f_y are their joint and marginal probability function. It is easy to prove that there is a close relationship between MI and the notion of entropy, $H(\boldsymbol{x}) = -\mathbb{E}\log(\rho(\boldsymbol{x}_1, \dots, \boldsymbol{x}_p))$. The MI can be seen as the relative entropy between the joint distribution and the product distribution. If $H(\boldsymbol{y} \mid \boldsymbol{x}) = -E(\log \rho(\boldsymbol{y} \mid \boldsymbol{x}_1, \dots, \boldsymbol{x}_p))$ is the conditional entropy, then

$$MI(\boldsymbol{y}, \boldsymbol{x}) = H(\boldsymbol{y}) = H(\boldsymbol{y} \mid \boldsymbol{x}) = H(\boldsymbol{x}) - H(\boldsymbol{x} \mid \boldsymbol{y})$$

is valid. Mutual information can be seen as the amount of information in $\boldsymbol{x}$ that can be used to reduce the uncertainty of $\boldsymbol{y}$.

GAM and Maximization of Mutual Information

This section provides the link between GAM and the mutual information. The prerequisite that GAM chooses $\hat{\eta}(\cdot)$ to maximize the expected log-likelihood was proofed by Hastie and Tibshirani [1990].

$$\mathbb{E}\left[l(\hat{\eta}(\boldsymbol{x}), \boldsymbol{y})\right] = \max_{\boldsymbol{\eta}} \mathbb{E}\left[l(\eta(\boldsymbol{x}), \boldsymbol{y})\right]$$

where $l(\eta(\boldsymbol{x}), \boldsymbol{y})$ is the log-likelihood of $\boldsymbol{y}$ given $\eta(\boldsymbol{x})$. Another requirement is that the mutual information between $\boldsymbol{y}$ and $\eta(\boldsymbol{x})$ is equal to

$$\begin{aligned} MI(\boldsymbol{y}; \eta(\boldsymbol{x})) &= \mathbb{E}\left[\log \frac{f(\eta(\boldsymbol{x}), \boldsymbol{y})}{f_{\eta(\boldsymbol{x})}(\eta(\boldsymbol{x})) f_y(\boldsymbol{y})}\right] \\ &= \mathbb{E}\left[l(\eta(\boldsymbol{x}), \boldsymbol{y})\right] - \mathbb{E} \log f_y(\boldsymbol{y}). \end{aligned} \tag{2.6}$$

Therefore, GAM chooses $\boldsymbol{\eta}$ in a way that $MI(\boldsymbol{y}; \eta(\boldsymbol{x}))$ is maximal. At the same time, Cover and Thomas [1991] showed for any function $\eta(\boldsymbol{x})$

$$MI(\boldsymbol{y}; \eta(\boldsymbol{x})) \leq MI(\boldsymbol{y}; \boldsymbol{x}) \tag{2.7}$$

is valid. Thus, it is not possible to increase information about $\boldsymbol{y}$ by transforming the original predictors $\boldsymbol{x}$ [Cover and Thomas, 1991]. This results in the purpose to find a suitable $\eta(\boldsymbol{x})$ to maximize $MI(\boldsymbol{y}; \eta(\boldsymbol{x}))$ and come as close as possible to the upper bound $MI(\boldsymbol{y}; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_p)$.

With the help of the chain rule, a possible solution for this maximization problem is given by

$$\begin{aligned} & MI(\boldsymbol{y}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_p) \\ &\quad = MI(\boldsymbol{y}; \boldsymbol{x}_1) + M(\boldsymbol{y}; \boldsymbol{x}_2 \mid \boldsymbol{x}_1) + \ldots + M(\boldsymbol{y}; \boldsymbol{x}_p \mid \boldsymbol{x}_{p-1}, \ldots, \boldsymbol{x}_1) && (2.8) \\ &\quad = MI(\boldsymbol{y}; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{p-1}) + MI(\boldsymbol{y}; \boldsymbol{x}_p \mid \boldsymbol{x}_{p-1}, \ldots, \boldsymbol{x}_1). && (2.9) \end{aligned}$$

Cover and Thomas [1991] point out that one possible method to maximize $MI(\boldsymbol{y}; \eta(\boldsymbol{x}))$ is to construct $\eta(\boldsymbol{x})$ term by term. Each term shall come as close as possible to the equation in (2.8). The smooth term $f_1(\boldsymbol{x}_1)$ is received by fitting a GAM of $\boldsymbol{y}$ onto $\boldsymbol{x}_1$. Thus, the approximation of $MI(\boldsymbol{y}, f_1(\boldsymbol{x}_1))$ by $MI(\boldsymbol{y}, \boldsymbol{x}))$ is achieved.

Consider the model $\boldsymbol{y} = f_1(\boldsymbol{x_1}) + \boldsymbol{z}$ where $\boldsymbol{z}$ is independent from $\boldsymbol{x}_1$ and also conditionally independent from $\boldsymbol{x}_1$. Hence,

$$\begin{aligned}
MI(\boldsymbol{y}; \boldsymbol{x}_2 \mid \boldsymbol{x}_1) &= H(\boldsymbol{y} \mid \boldsymbol{x}_1) - H(\boldsymbol{y} \mid \boldsymbol{x}_2, \boldsymbol{x}_1) && (2.10)\\
&= H((f_1(\boldsymbol{x}_1) + \boldsymbol{z}) \mid \boldsymbol{x}_1) - H((f_1(\boldsymbol{x}_1) + \boldsymbol{z}) \mid \boldsymbol{x}_2, \boldsymbol{x}_1)\\
&= H(\boldsymbol{z} \mid \boldsymbol{x}_1) - H(\boldsymbol{z} \mid \boldsymbol{x}_2, \boldsymbol{x}_1)\\
&= H(\boldsymbol{z}) - H(\boldsymbol{z} \mid \boldsymbol{x}_2) && (2.11)\\
&= MI(\boldsymbol{z}, \boldsymbol{x}_2). && (2.12)
\end{aligned}$$

The equations (2.10) and (2.12) result straightforwardly from the definition of mutual information. Equation (2.11) follows directly from the premise. The next term $f_2(\boldsymbol{x}_2)$ is constructed by taking the partial residual $\boldsymbol{z} = \boldsymbol{y} - f_1(\boldsymbol{x}_1)$ and then fit a GAM of $\boldsymbol{z}$ onto $\boldsymbol{x}_2$.

How can further terms $f_3(\boldsymbol{x}_3), \ldots, f_k(\boldsymbol{x}_k)$ be constructed? Generalize the idea above and consider the model $\boldsymbol{y} = \eta_k(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) + \boldsymbol{z}$. The terms $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$ and $\boldsymbol{z}$ are assumed to be independent. Additionally, the terms are given conditionally independent of $\boldsymbol{x}_{k+1}$. This leads to

$$MI(\boldsymbol{y}; \boldsymbol{x}_{k+1} \mid \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) = MI(\boldsymbol{z}; \boldsymbol{x}_{k+1}). \tag{2.13}$$

The aim is to approximate the terms in equation (2.8). This is done by using $f_1(\boldsymbol{x}_1), \ldots, f_k(\boldsymbol{x}_k)$ as an approximation for $\eta_k(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$. Thus, the approximation only works if $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ are independent.
Assume that there is concurvity between $\boldsymbol{x}_k$ and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k-1}$. How will this affect the procedure? The approximation results to $\eta_{k-1}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k_1})$. The consequence after adding $f_k(\boldsymbol{x}_k)$ is that the partial residuals are still not independent of $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k_1})$. Thereby, it applies $\boldsymbol{z} = \boldsymbol{y} - \eta_{k-1}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k_1}) - f_k(\boldsymbol{x}_k)$ and this leads to the fact that equation (2.13) is false. For this reason, Gu et al. [2010] point out that the backfitting algorithm requires multiple passes and each function must be re-fitted iteratively. The re-fitting is possible and correct because the chain rule in equation (2.8) and (2.9) does not depend on the order of the $\boldsymbol{x}_j$'s.

2.5.2. Partial Generalized Additive Models

Covariates are not independent in the presence of concurvity. Accordingly, approaching the terms in equation (2.8) in the backfitting algorithm does not lead to the optimal result. Therefore, an alternative to approximate the equation (2.8) is necessary. Partial generalized additive models provide this alternative way by using the recursive application of the following:

$$MI(\boldsymbol{y}, \boldsymbol{x}_1, \dots, \boldsymbol{x}_p) = MI(\boldsymbol{y}; \boldsymbol{x}_1) + MI(\boldsymbol{y}; \boldsymbol{x}_2, \dots, \boldsymbol{x}_p \mid \boldsymbol{x}_1) \tag{2.14}$$

instead of using the recursive application of equation (2.9). Consider the model $\boldsymbol{y} = f_1(\boldsymbol{x}_1) + \boldsymbol{z}$ where $\boldsymbol{z}$ is independent of $\boldsymbol{x}_1$ again. In this case assume $\boldsymbol{x}_j = g_{j1}(\boldsymbol{x}_1) + \boldsymbol{x}^{(j)}$; $j = 2, \dots, p$ where $\boldsymbol{x}_1$ and $(\boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(p)})$ are independent from $\boldsymbol{z}$. Instead of equation (2.10) – (2.12) this leads to

$$\begin{aligned}
&MI(\boldsymbol{y}; \boldsymbol{x}_2, \dots, \boldsymbol{x}_p \mid \boldsymbol{x}_1) \\
&\quad = H(\boldsymbol{x}_2, \dots, \boldsymbol{x}_p \mid \boldsymbol{x}_1) - H(\boldsymbol{x}_2, \dots, \boldsymbol{x}_p \mid \boldsymbol{y}, \boldsymbol{x}_1) \\
&\quad = H((g_{j1}(\boldsymbol{x}_1) + \boldsymbol{x}^{(j)})_{j=2,\dots,p} \mid \boldsymbol{x}_1) - H((g_{j1}(\boldsymbol{x}_1) + \boldsymbol{x}^{(j)})_{j=2,\dots,p} \mid f_1(\boldsymbol{x}_1) + \boldsymbol{z}, \boldsymbol{x}_1) \\
&\quad = H(\boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(p)} \mid \boldsymbol{x}^{(1)}) - H(\boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(p)} \mid \boldsymbol{z}, \boldsymbol{x}_1) \\
&\quad = H(\boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(p)}) - H(\boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(p)} \mid \boldsymbol{z}) \\
&\quad = MI(\boldsymbol{z}; \boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(p)}).
\end{aligned} \tag{2.15}$$

Thus, in this case another procedure is necessary. In a first step, estimate $f_1(\boldsymbol{x}_1)$ by fitting a GAM of $\boldsymbol{y}$ onto $\boldsymbol{x}_1$. The estimation of the "partial effects" $g_{21}, \dots, g_{p1}$ is done by smoothing $\boldsymbol{x}_2, \dots, \boldsymbol{x}_p$ onto $\boldsymbol{x}_1$. Finally, fit $\boldsymbol{z}$ onto the adjusted variables $\boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(p)}$, which are independent from $\boldsymbol{x}_1$.

Variable Selection

When regression models are fitted, the variable selection plays a very important role. Particularly the order in which the variables enter a model make a difference to the space of possible models. Also, the final model depends on this chosen order. pGAM chooses the variables in order of decreasing mutual information of $\boldsymbol{y}$. Thus, the variable with the highest MI is chosen first. pGAM is only able to do a forward variable selection. Variables are only included in the final model if they improve the model significantly.

Indirect Estimation of MI

As mentioned in section 2.5.1, the mutual information is an important component in the pGAM procedure. Unfortunately, the direct estimation of the MI is not a trivial problem. Thus, the estimation of MI must be effected in an alternative way.
This can be realized by using a "proxy" of $MI(\boldsymbol{y};\boldsymbol{x})$ which is based on equation (2.6) and (2.7)

$$\widehat{MI(\boldsymbol{y};\boldsymbol{x})} = \max_{\boldsymbol{\eta}} MI(\boldsymbol{y};\eta(\boldsymbol{x})) = \max_{\boldsymbol{\eta}} \mathbb{E}\left[l(\eta(\boldsymbol{x}),\boldsymbol{y})\right] - \mathbb{E}\log f_y(\boldsymbol{y}). \tag{2.16}$$

This procedure is justified because $\eta(\boldsymbol{x})$ is a sufficient statistic for $\boldsymbol{y}$ and this leads to $MI(\boldsymbol{y};\eta(\boldsymbol{x})) = MI(\boldsymbol{y};\boldsymbol{x})$ [Cover and Thomas, 1991]. Gu et al. [2010] propose to consider only the maximum value of the conditional log-likelihood. Practically, this is realized by fitting a GAM of $\boldsymbol{y}$ onto each remaining covariate and choose the covariate with the largest log-likelihood (or the smallest deviance).

The pGAM Algorithm

The basic structure of the pGAM procedure is as followed. pGAM sequentially maximizes the mutual information [Shannon et al., 1948] between the response variable and the covariates. pGAM starts with a null model. Firstly, pGAM chooses to add the covariate whose mutual information with $\boldsymbol{y}$ is the largest. Secondly, it removes any functional effects of this covariate from all remaining covariates before searching for the next covariate to add. Finally, this leads to a model based on a sequence of adjusted predictor variables. The removal of the functional dependencies at each step, eliminates problems induced by concurvity and gives much more precise and reliable interpretations of the covariate effects. Consider that after the first covariate all covariates are transformed during the fitting process. The entire pGAM algorithm from Gu et al. [2010] is presented in the following:

pGAM Algorithm

(1) **Initialization:**

(a) Starting with a null model m_0 by fitting a GAM of $\boldsymbol{y}$ onto a constant; let D_0 be the deviance of m_0.

(b) Center all $\boldsymbol{x}_j$'s to have mean zero; let

$$\boldsymbol{\chi}_\omega = \{\boldsymbol{x}^{(j)} = \boldsymbol{x}_j, j = 1, \dots, p\}$$

be the initial set of "working variables".

(c) Set $D = D_0$ and $m = m_0$.

(2) **Iterate Main Procedure:**

(a) For each working variable $\boldsymbol{x}^{(j)}$ in $\boldsymbol{\chi}_\omega$, fit a GAM of $\boldsymbol{y}$ onto $\boldsymbol{x}^{(j)}$. Record the deviance d_j, and the degree of freedom df_j for $\boldsymbol{x}^{(j)}$. Collect d_i into a vector $\boldsymbol{d}$.

(b) Choose i such that d_i is the smallest element of $\boldsymbol{d}$. Remove d_i from $\boldsymbol{d}$ and $\boldsymbol{x}^{(j)}$ from $\boldsymbol{\chi}_\omega$.
Form a new model m_{new} by adding $\boldsymbol{x}^{(i)}$ into m. Let D_{new} be the deviance of m_{new}.

(c) Test whether D_{new} is a significant improvement over D:

- Improvement is <u>not</u> significant:
  If $\boldsymbol{\chi}_\omega$ is not empty, then go back to step 2(b).

- Improvement <u>is</u> significant:
  - For every $\boldsymbol{x}^{(j)} \in \boldsymbol{\chi}_\omega (j \neq i)$, fit

    $$\boldsymbol{x}^{(j)} = g_{ji}(\boldsymbol{x}^{(i)}) + \boldsymbol{\varepsilon}_j$$

    by smoothing $\boldsymbol{x}^{(j)}$ onto $\boldsymbol{x}^{(i)}$. Record the fitted functions g_{ij} and replace each

    $$\boldsymbol{x}^{(j)} = \boldsymbol{x}^{(j)} - g_{ij}(\boldsymbol{x}^{(i)})$$

    in $\boldsymbol{\chi}_\omega$.
  - Let $D = D_{new}$ and $m = m_{new}$.

- ○ If χ_ω is not empty, then go back to step 2(a).

(3) **Output:**
Run until all variables are tried out and the model m and the g_{ji}'s are put out.

3. Concurvity

3.1. Introduction

What are predictive maps of species distributions [Franklin, 2009]? What are the short-term or long-term effects of air pollution on the population of a country [He, 2004]?

These and similar questions increase on importance nowadays. With the help of reliable statistical data and models these questions should be answered. Species distribution modeling (SDM) is just one example of this. However, it is an increasingly important one [Franklin, 2009] which helps to assess changes of landscapes [Miller et al., 2004; Peters and Herrick, 2004]. One further example can be regression approaches to model the rent level or epidemiological models for cancer atlases.

These applications with their particular data structure provide other requirements to the statistical models. The first standard assumption in linear models is that the observations are independent [Yee and Mitchell, 1991]. This assumption violates the "first law of geography" [Tobler, 1979] as well as ecology [Legendre and Fortin, 1989; Fortin and Dale, 2005]. It states that near things are similar which concludes that nearby locations have similar values because they are likely to influence each other. Thus, the data in these applications are spatially correlated.

Thereby, a special modeling is necessary for this special data structure. Generally, the predictor variables in such models show a strong spatial correlation. The modeling and its related problems are briefly explained with an example. At this point, the application, later presented in chapter 5, is already anticipated.

The aim is to estimate a model for the occurrence of certain tree species in Bavaria. This leads to a binary response variable y_i. As predictor variables precipitation and temperature are available and as well the spatial coordinates at which these variables were measured. Thereby, the predictor variables show a spatial correlation.

The generalized additive model (chapter 2), a flexible modeling technique, shall be used to model the relationship between dependent and independent variables. With the help of a stepwise modeling strategy, the problem "concurvity" is introduced. Firstly, assume the following model

$$\begin{aligned} \mathbb{E}(y_i) &= \mathbb{P}(y_i = 1) \\ &= \beta_0 + \beta_1(\textbf{precipitation}) + \beta_2(\textbf{temperature}). \end{aligned} \tag{3.1}$$

This model (3.1) ignores completely the special data structure. That means concretely the spatial autocorrelation of data. This situation is illustrated schematically in figure 3.1. The red bar represents the covariate effect. Due to the spatial correlation of the covariates, it is possible to imagine that the covariates effect can be decomposed into two parts: the "covariate-part" (green) and the "spatial-part" (yellow). As mentioned before, the model (3.1) ignores

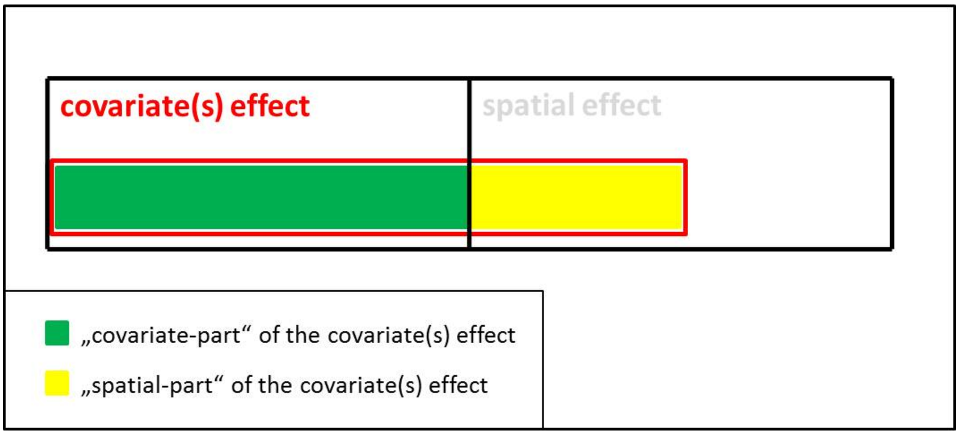
covariate(s) effect
spatial effect
🟩 „covariate-part" of the covariate(s) effect
🟨 „spatial-part" of the covariate(s) effect

Figure 3.1.: Decomposition of the spatial correlated covariate(s): "covariate part" of the covariate(s) effect (green); "spatial-part" of the covariate(s) effect (yellow).

completely the yellow “spatial-part”.

What are the consequences of this?
Not accounting for spatial autocorrelation, will lead to biased estimation and too optimistic confidence intervals and increasing first type error [Legendre, 1993; Wagner and Fortin, 2005; Segurado et al., 2006]. Additionally, variable selection is conceivably predisposed towards more strongly autocorrelated predictors [Lennon, 2000].

This problem can be solved by adding a spatial term to the model (3.1). This approach is sketched in figure 3.2 again. The red bar indicates for the covariate effect. The blue bar represents the new introduced spatial effect.

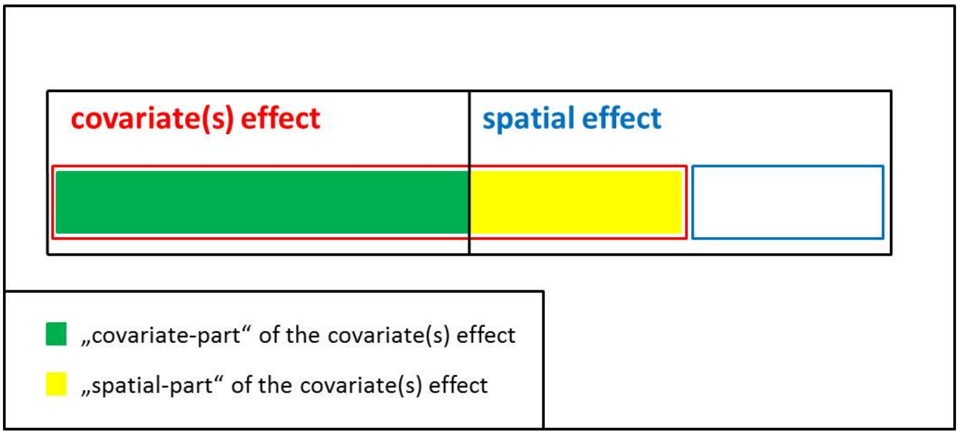
covariate(s) effect
spatial effect
„covariate-part“ of the covariate(s) effect
„spatial-part“ of the covariate(s) effect

Figure 3.2.: Improve the model with spatial correlated covariate(s) with the help of an extra spatial effect (blue): “covariate part” of the covariate(s) effect (green); “spatial-part” of the covariate(s) effect (yellow).

$$\begin{aligned}\mathbb{E}(y_i) &= \mathbb{P}(y_i = 1) \\ &= \beta_0 + \beta_1(\textbf{precipitation}) + \beta_2(\textbf{temperature}) + \underbrace{g(\textbf{Long}, \textbf{Lat})}_{\text{spatial effect}} .\end{aligned} \tag{3.2}$$

The spatial autocorrelation is considered by the extra spatial effect $g(\textbf{Long}, \textbf{Lat})$. This fact is graphically represented by figure 3.3. The blue bar representing the spatial effect, covers the “spatial effect”-part of the covariate effect. Thereby, the spatial effect absorbs the spatial autocorrelation of the covariates/data. Additionally, the spatial effect serves as a surrogate for all other unobserved [Fahrmeir et al., 2009]. Thereby the estimations are stabilized.

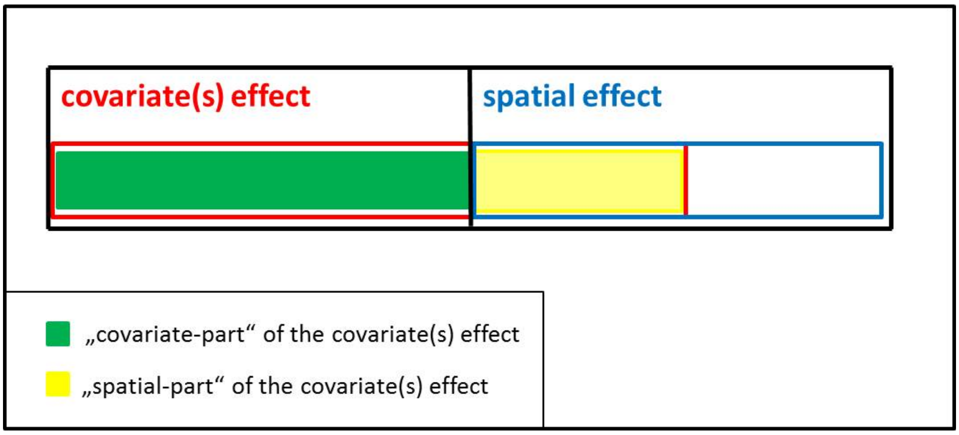
covariate(s) effect
spatial effect
„covariate-part“ of the covariate(s) effect
„spatial-part“ of the covariate(s) effect

Figure 3.3.: Improve the model with spatial correlated covariate(s) with the help of an extra spatial effect (blue): "covariate part" of the covariate(s) effect (green); "spatial-part" of the covariate(s) effect (yellow).

However, introducing a spatial effect creates new problems. The problem arises because of the fact that the covariates precipitation and temperature are spatially correlated. Adding a spatial effect to the model at the same time, leads to "multicollinearity in non-linear models". This problem is better known as the term "concurvity" [Buja et al., 1989; Hastie and Tibshirani, 1990; Guisan et al., 2002; He, 2004]. Therefore, concurvity can be seen as the existence of multiple solutions when a generalized additive model is fitted [Hastie and Tibshirani, 1990]. Concurvity leads to instability and difficult interpretability of the estimated covariate effects. To date, the impact of concurvity on the parameter estimates has not been fully investigated [He, 2004].

What are the possible consequences of concurvity in general?
The presence of concurvity in the data and the use of generalized additive models is risky, especially, when the association is weak, the model can seriously overestimate parameters and underestimate their variances [He, 2004]. Note, that concurvity is considered in the calculation of standard errors. The

greater the concurvity is, the greater the standard error will be [He, 2004]. However, the underestimation of standard errors and biased regression coefficients due to concurvity lead to significance tests with inflated type 1 error [Ramsay et al., 2003*b*,*a*]. This can result in declaring erroneously a statistically significant effect, even when none exists.

Inferential problems when using generalized additive models in the presence of concurvity are discussed in several recent papers, for example by Ramsay et al. [2003*b*]; Figueiras et al. [2005] and Lumley and Sheppard [2003].

The reason why concurvity occurs and a possible way to solve this problem is discussed in the following section.

3.2. The %ll%-Operator

If covariate and spatial effects are modeled at the same time in order to cover spatial autocorrelation and unobserved heterogeneity, it will lead to wrong or attenuated effects in the presence of "concurvity" [He, 2004]. That is caused because spatial autocorrelation cannot clear separate between spatial and covariate effect. This situation is schematically illustrated in figure 3.2. Flexible modeling of the spatial effect includes it consists of enough degrees of freedom for absorbing the covariate effect partially (figure 3.3). Consider the simple geo-additive model as presented in section 2.2

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{Z}\boldsymbol{\gamma} + \boldsymbol{\varepsilon} \\ &= \boldsymbol{Z}_1\boldsymbol{\gamma}_1 + \ldots + \boldsymbol{Z}_q\boldsymbol{\gamma}_q + \boldsymbol{Z}_{\text{spatial}}\boldsymbol{\gamma}_{\text{spatial}} + \boldsymbol{\varepsilon} \end{aligned} \tag{3.3}$$

with the design matrix $\boldsymbol{Z}$. The model (3.3) consists of $1, \ldots, q$ covariates and a spatial effect.

For that reason, the question arises if there is another way to model the "spatial-part" of the covariates effects. One possible option is to modify the spatial effect. It is hoped that the falsification of the estimated covariate effects can be prevented or weakened. The %ll%-operator will carry out these

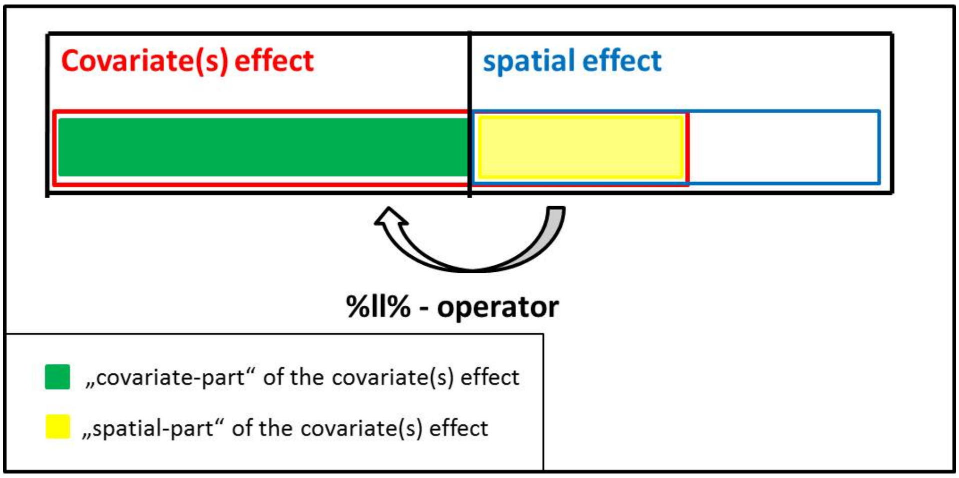
Covariate(s) effect
spatial effect
%ll% - operator
„covariate-part“ of the covariate(s) effect
„spatial-part“ of the covariate(s) effect

Figure 3.4.: Idea of the %ll%-Operator: separate the spatial autocorrelation of the spatial and covariate effect.

modifications of the spatial effect.

The basic idea is to modify the spatial effect in a way that can only reproduce the variability which cannot be explained by covariate information in principle. Technically, it can be reached by making the basis functions used for spatial effect orthogonal to the basis functions to the covariate effect.

Consider $\boldsymbol{X}_{\text{spatial}}$ to be the design matrix of a spatial effect and $\boldsymbol{X}_{\text{covar}}$ the design matrix of the covariates effects. $\boldsymbol{X}_{\text{spatial}}$ and $\boldsymbol{X}_{\text{covar}}$ are spatially correlated with each other. To get rid of the "concurvity"-problem, make the spatial effect $\boldsymbol{X}_{\text{spatial}}$ orthogonal to the basis functions to the covariate effect $\boldsymbol{X}_{\text{covar}}$. This is done by

$$\boldsymbol{X}^{\text{mod}}_{\text{spatial}} = (\boldsymbol{I} - \boldsymbol{X}_{\text{covar}}(\boldsymbol{X}^T_{\text{covar}}\boldsymbol{X}_{\text{covar}})^{-1}\boldsymbol{X}^T_{\text{covar}})\boldsymbol{X}_{\text{spatial}}. \tag{3.4}$$

Thus, $\boldsymbol{X}^{\text{mod}}_{\text{spatial}}$ is the residuals of $\boldsymbol{X}_{\text{spatial}}$ regressed on $\boldsymbol{X}_{\text{covar}}$. Now $\boldsymbol{X}^{\text{mod}}_{\text{spatial}}$ and $\boldsymbol{X}_{\text{spatial}}$ are orthogonal by construction and their coefficients' standard errors are therefore not inflated by concurvity [Hodges and Reich, 2010]. Reich et al. [2006] extend this idea to non-normal observables. Apply this idea to the model (3.3). Therefore, modify the design matrix of the spatial effect $\boldsymbol{Z}_{\text{spatial}}$

with the help of equation (3.4). Thereafter, Hodges and Reich [2010] suggest to replace $\boldsymbol{Z}_{\text{spatial}}$ with the modified $\boldsymbol{Z}^{\text{mod}}_{\text{spatial}}$ in the model (3.3):

$$\boldsymbol{y} = \boldsymbol{Z}_1\boldsymbol{\gamma}_1 + \ldots + \boldsymbol{Z}_q\boldsymbol{\gamma}_q + \boldsymbol{Z}^{\text{mod}}_{\text{spatial}}\boldsymbol{\gamma}_{\text{spatial}} + \boldsymbol{\varepsilon}.$$

Now, the modified spatial effect is orthogonal to the covariates effect and no longer be inflated by concurvity [Hodges and Reich, 2010].

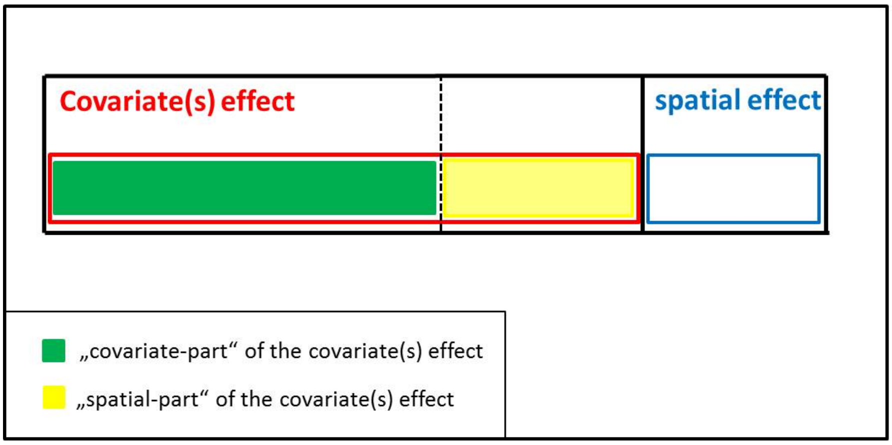
Covariate(s) effect
spatial effect
🟩 „covariate-part“ of the covariate(s) effect
🟨 „spatial-part“ of the covariate(s) effect

Figure 3.5.: Impact of the %ll%-Operator: separate the spatial autocorrelation of the spatial and covariate effect.

Figure 3.5 shows the result of the modification of the spatial effect by the %ll%-operator. The yellow "spatial-part" of the covariate effect only belongs to the covariate effect. Thus, the spatial effect, represented by the blue bar, covers only the spatial autocorrelation and unobserved heterogeneity.

3.3. Implementation Details

This section introduces briefly the R [R Development Core Team, 2012] add-on package `mboost` [Hothorn et al., 2009]. In addition, the practical implementation of the %ll%-idea as an extension of the `mboost`-package in R is presented.

3.3.1. The `mboost` package

The R add-on package `mboost` was developed by Hothorn et al. [2009]. It allows modern regression modeling and beyond this provides an interface be-

tween classical regression models and machine-learning approaches for complex interaction models [Hothorn et al., 2009]. The models are fitted with the help of model-based boosting methods as introduced in chapter 2.4 and result in interpretable models.
As the present thesis only deals with generalized additive models, only the relevant parts for that are presented.

3.3.2. Fitting Generalized Additive Models: `gamboost`

For generalized additive models the R package `mboost` offers a flexible and powerful interface because of its combination of a distributional and a structural assumption (see section 2.1 for details).
The distributional assumption is specified by the distribution of the outcome. In comparison to this, the structural assumption specifies the types of effects that are used in the model, i.e. it represents the deterministic structure of the model. The structural assumption defines how the predictors are related to the conditional mean of the outcome and it is given by using base-learners.

To fit structured additive models, the function `gamboost()` can be used:

```
gamboost(formula, data = list(),
         baselearner = c("bbs", "bols", "btree", "bss", "bns"),
         dfbase = 4, ...)
```

With the help of this function, it is possible to fit linear or (non-linear) additives models via component-wise boosting. The user only has to specify in the `formula`-argument which variable should enter the model in which fashion, for example as a linear or a smooth effect. This is done by the different `baselearner`. The specification of these different fashions will be briefly discussed in the following section.

3.3.3. Choosing Base-Learners

As mentioned before, the structural assumption of the model, especially the types of effects that are used, can be specified in terms of base-learners. Each base-learner results in a related type of effect. For example, a base-learner can be either linear (`bols`) or a smooth effect (`bbs`).

However, it should be considered to prevent the single base-learners from overshooting. The degrees of freedom of single base-learners should be kept small enough. Hothorn et al. [2009] propose 4 degrees of freedom or even less. Furthermore, the authors point out that the small initial degrees of freedom, the final estimate that results from these base-learners, can adopt higher order degrees of freedom due to the iterative nature of the algorithm.

Linear and Categorial Effects

The function `bols()` can be used to fit linear or categorial effects of variables. This function allows the definition of (penalized) ordinary least squares base-learners.

Smooth Effects

`bbs()` base-learners allow the definition of smooth effects based on B-Splines with difference penalty. B-Splines are described in section 2.3 in detail. Usually, this base-learner is used in the analysis later.

Smooth Surface Estimation

The base-learner `bspatial()` can be seen as an extension of P-Splines to two dimensions which is given by bivariate P-Splines. Bivariate P-Splines are introduced in section 2.3.2. With this help, it is possible to fit spatial effects and smooth interactions surfaces.

The %ll%-Operator

This section presents the implementation details in `R` of the new %ll%-operator. The theoretical background of the idea, making the spatial effect orthogonal to the basis functions to the covariate effect, is presented in section 3.3.3. The complete code of the operator is given in the Appendix.

Consider the two base-learner, base-learner 1, named in the following “bl1” and base-learner 2, named in the following “bl2”. Here, bl1 represents a spatial effect. The spatial effect is specified with the help of the `bspatial(·, ·)` base-learner:

bl1: `bspatial(x.coord, y.coord)`

Compared with that, bl2 consists of $1, \dots, p$ base-learner for the $1, \dots, p$ spatially measured covariates. This covariates for example can be modeled with the help of the `bbs(·)` base-learner. As an example and for a better understanding, consider one covariate $x1$, modeled by the `bbs(·)` base-learner:

bl2: `bbs(x1)`

In order to get rid of the “concurvity” problematic, the spatial effect needs to be made orthogonal to the basis functions to the covariate effect. This is done with the help of the %ll%-operator:

`bspatial(x.coord, y.coord)` %ll% `bbs(x1)`

The centerpiece of the %ll%-operator is the `Xfun`-function. In it, the orthogonalization is performed.

```
### Xfun
Xfun <- function(mf, vary, args){

    ## create X and K matrices
    newX1 <- environment(bl1$dpp)$newX
    newX2 <- environment(bl2$dpp)$newX

    ## extract X and K matrices
    X1 <- newX1(mf[, bl1$get_names(), drop = FALSE])
    K1 <- X1$K
    if (!is.null(l1)) K1 <- l1 * K1
    X1 <- X1$X

    X2 <- newX2(mf[, bl2$get_names(), drop = FALSE])
    K2 <- X2$K
    if (!is.null(l2)) K2 <- l2 * K2
    X2 <- X2$X
```

Firstly, the desgin matrices $\boldsymbol{X}$ and the penalty matrices $\boldsymbol{K}$ of the two base-learner “bl1” and “bl2” are extracted. Thereafter, the orthogonalization is performed. The design matrix $\boldsymbol{X1}$ of base-learner “bl1” should be orthogonal to design matrix $\boldsymbol{X2}$ of base-learner “bl2”.

```
    ## make X1 orthogonal to X2
    ## X1orth <- qr.resid(qr(X2), X1) = I - (X2 (X2'X2)^-1 X2') X1
    X1orth <- qr.resid(qr(X2), X1)

    ## new design matrix X
    X <- X1orth

    ## new penalty matrix K
    K <- K1
```

The orthogonalization is done with the help of `qr.resid`-function.
The `qr.resid(qr(X2), X1)`-function is equivalent to
$\boldsymbol{X}_1\text{orth} = \boldsymbol{I} - (\boldsymbol{X}_2(\boldsymbol{X}_2^T\boldsymbol{X}_2)^{-1}\boldsymbol{X}_2^T)\boldsymbol{X}_1$. `X1orth` corresponds to the residuals of the design matrix `X1` of “bl1” regressed on the design matrix `X2` of “bl2”. The new design matrix $\boldsymbol{X}$ only consists of the orthogonalized design matrix `Xorth` of base-leaner “bl1”. Afterwards, `X1orth` is orthogonal to `X2`. A new penalty matrix $\boldsymbol{K}$ is also required. The penalty matrix is not modified. As a new penalty matrix $\boldsymbol{K}$, the penalty matrix $\boldsymbol{K1}$ of base-learner “bl1” is used.

```
    ## return
    list(X = X, K = K)
}
```

Last but not least, the new design matrix $\boldsymbol{X}$ and penalty matrix $\boldsymbol{K}$ are returned.

If the model consists of $\boldsymbol{x}_1, \ldots \boldsymbol{x}_p$ covariates, the spatial effect is to be made orthogonal to all $\boldsymbol{x}_1, \ldots \boldsymbol{x}_p$ covariates. One small thing needs to be considered when the model is specified. It must be ensured that only one base-learner stands on the right side of the %ll%-operator. This is achieved by the %+%-operator [Hothorn et al., 2009]. The %+%-operator merges several base-learners to one huge base-learner:

$$\texttt{bbs(}x_1\texttt{)} \text{ \%+\%} \ldots \text{ \%+\% } \texttt{bbs(}x_p\texttt{).}$$

The %ll%-operator can be used as usual:

$$\texttt{bspatial(x.coord, y.coord)} \text{ \%ll\% } [\, \texttt{bbs(}x_1\texttt{)} \text{ \%+\%} \ldots \text{ \%+\% } \texttt{bbs(}x_p\texttt{)}]$$

Thus, the “new” spatial effect is orthogonal to the $\boldsymbol{x}_1, \ldots \boldsymbol{x}_p$ covariates effects.

4. Simulation Studies

This section investigates the performance of the %ll%-operator. All data analysis which are presented in this thesis have been carried out by using the R system for statistical computing [R Development Core Team, 2012], version 2.14.2.

4.1. Data Generating Process

4.1.1. Model

Before simulation studies can be conducted, there must be a sample of data with concurvity. Therefore, consider the model

$$y = f_{\text{geo}}(\mathbf{coords}) + \sum_{j=1}^{p} f_j(\boldsymbol{x}_j) + \boldsymbol{\varepsilon}$$

with optional $j = 1, \ldots, p$ covariates.

The model consists of the following single components:

- $f_{\text{geo}}(\mathbf{coords})$ is a spatial effect. Thus, the spatial effect is the realization of a spatial random field.

  $$f_{\text{geo}}(\mathbf{coords}) \sim MVN(\mathbf{0}, \boldsymbol{\Sigma}_s)$$

  Accordingly, *coords* represents the spatial coordinates (e.g. Longitude, Latitude).

- $f_j(\boldsymbol{x}_j)$ stand for the j-th covariate effect. A single $\boldsymbol{x}_j$ is generated by

  $$\boldsymbol{x}_j \sim MVN(\mathbf{0}, \boldsymbol{\Sigma}_{s,j} + \boldsymbol{\sigma}_j \boldsymbol{I})$$

Hence, $\boldsymbol{x}_j$ can be interpreted as the sum of the realization $g_j(\mathbf{coords})$ of a spatial random field and of an identical and independent normally distributed random variable $\boldsymbol{\varepsilon}_j \sim N(\mathbf{0}, \boldsymbol{\sigma}^2 \boldsymbol{I})$. The concurvity of the $\boldsymbol{x}_j$ is controlled by

$$\boldsymbol{x}_j = g_j(\mathbf{coords}) + \boldsymbol{\varepsilon}_j$$

Primarily, the strength of "concurvity" is regulated by the covariance function of the spatial random field $g_j(\mathbf{coords})$ and a specific Signal-to-Noise-Ratio. Details are examined in the following passages "Correlation function", "Signal-to-Noise-Ratio" and "SNRconcurvity" in section 4.1.2.

- The model variance $\boldsymbol{\varepsilon}$ is independent and identical distributed. The variance is subjected to the following distribution

$$\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \boldsymbol{\sigma}^2).$$

4.1.2. Settings of the Data Generating Process

There are some settings in the data generating process (DGP). These settings will be presented in the following section and briefly discussed. The `dgp()`-function allows the generation of data with concurvity. The DGP is based on the model presented in section 4.1. The function is called as shown:

```
> dgp <- function(N, gridType = c("irreg", "reg"),
        coord = list(xmin = 1, xmax = 10, ymin = 13, ymax = 19),
        covType = c("exp", "matern"), myTheta = 1, mySmoothness = 1,
        covariates = 5, dFxj = c("1", "2", "3", "4", "5"),
        SNRconcurvity = 1, SNRspatial = 1, SNRepsilon = 1,
        setSeed = 12012)
```

- **Sample size:**
  `N` allows regulating the sample size.

- **Grid type:**
  This setting allows choosing between two grid types. '`irreg`' produces an irregular grid. In contrast '`reg`' simulates a regular grid. In addi-

tion, with the help of the argument `coord` it is possible to specify the coordinates of the grid exactly.

- **Correlation function:**
  Firstly, `covType` allows to define which covariance type should be used. The exponential (`‘exp’`) and the Matèrn (`‘matern’`) covariance functions are implemented. `myTheta` and `mySmoothness` are additional parameters of the covariance function. Thus, `myTheta` is a range parameter. `mySmoothness` is the Matèrn smoothness parameter which controls the number of derivatives in the process. Figure 4.1 illustrates several visualizations of either Matèrn or exponential covariance functions.

- **Number and effect of the covariates:**
  The argument `covariates` allows to specify how many covariates should be generated. Thus, `dFxj` determines the smooth effect of the single covariate. For the smooth effect it is possible to choose between five different parametric functions:

  (1) $f(x) = 0.25 \cdot \sin(\frac{\pi}{2} \cdot x)$

  (2) $f(x) = 0.25 \cdot (x^4)$

  (3) $f(x) = 2 \cdot \frac{\sin(x^2)}{(x^2)}$

  (4) $f(x) = -0.75 \cdot \cos(0.75 \cdot \pi \cdot x)$

  (5) $f(x) = x$

  To make the estimation of the functions much more easier and more comparable, each covariate is scaled to an interval from -4 to 4. A visualization of the five different smooth functions can be seen in figure 4.2.

- **Signal-to-noise-ratios:**
  There are three different signal-to-noise-ratios implemented to inspect the strengths and weakness of the methods:

  1. `SNRconcurvity`:

$$\text{SNRconcurvity} = \frac{\text{sd}(g_j(\mathbf{coords}))}{\text{sd}(\varepsilon_j)}$$

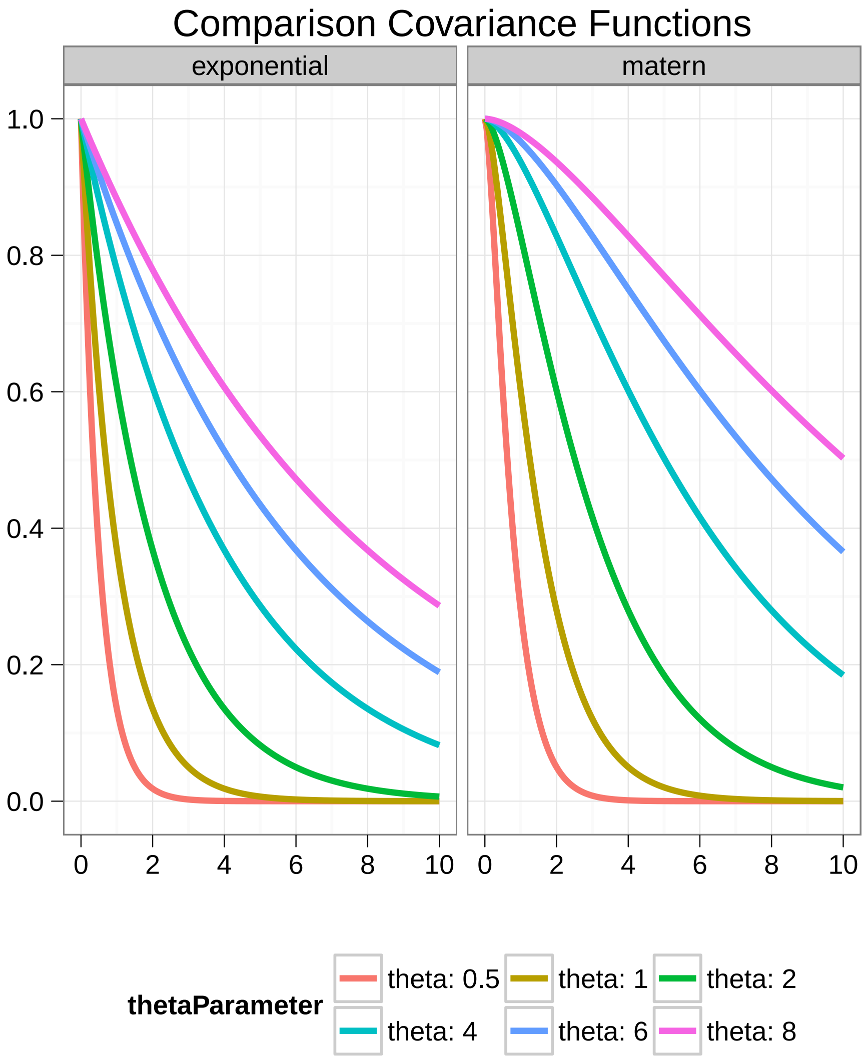

Figure 4.1.: Comparison of different exponential (left) and Matèrn (right) covariance functions for several `myTheta` parameters.

SNRconcurvity regulates the variance ratio between $g_j(\mathbf{coords})$ and $\boldsymbol{\varepsilon}_j$. Therefore, it corresponds to the strength of the spatial correlation of the covariates.

2. `SNRspatial`:

$$\text{SNRspatial} = \frac{\text{sd}(f_{\text{geo}}(\mathbf{coords}))}{\text{sd}(\sum_j f_j(\boldsymbol{x}_j))}$$

SNRspatial regulates the variance ratio between $f_{\text{geo}}(\textbf{coords})$ and $\sum_j f_j(\boldsymbol{x}_j)$. This variance ratio controls which ratio of the explainable variance corresponds to the spatial effect.

3. `SNRepsilon`:

$$\text{SNRepsilon} = \frac{\text{sd}(f_{\text{geo}}(\textbf{coords}) + \sum_j f_j(\boldsymbol{x}_j))}{\text{sd}(\boldsymbol{\varepsilon})}$$

SNRepsilon regulates the variance ratio between $f_{\text{geo}}(\textbf{coords}) + \sum_j f_j(\boldsymbol{x}_j)$ and $\boldsymbol{\varepsilon}$. This "classical signal-to-noise-ratio" controls which ratio of data variance can be explained by the model.

- **Reproducibility:**
  This setting allows to set a seed for reproducible results.

4.2. Simulation Framework

The simulation design to investigate the performance of the %ll%-operator and its comparison to other methods are presented in detail in the following section. Three different simulations with different settings shall be considered to investigate the performance of %ll%-operator. Furthermore, the %ll%-operator is compared to two other models.

Simulation 1:
In the first simulation the following model shall be considered:

$$\boldsymbol{y} = f_{\text{geo}}(\textbf{coords}) + f_1(\boldsymbol{x}) + \boldsymbol{\varepsilon}.$$

This first simple simulation only consists of one covariate and an additional spatial effect. The smooth effect of the covariate is generated by the parametric function (1).

Simulation 2:
The second simulation is based on the following model:

$$\boldsymbol{y} = f_{\text{geo}}(\mathbf{coords}) + \sum_{j=1}^{3} f_j(\boldsymbol{x}_j) + \boldsymbol{\varepsilon}.$$

This simulation consists of a spatial effect and additional of three covariates. The covariates are generated by using three different parametric functions (1), (2) and (3).

Simulation 3:
Simulation 3 considers the model:

$$\boldsymbol{y} = f_{\text{geo}}(\mathbf{coords}) + \sum_{j=1}^{5} f_j(\boldsymbol{x}_j) + \boldsymbol{\varepsilon}.$$

Simulation 3 differs from simulation 2 because it consists of two additional covariates. The five covariates are generated by five different available parametric functions. Each function is only used once.

4.2.1. Methods

The `gamboost`-model with the %ll%-operator modification shall be compared to the “basis”-`gamboost`-model and the pGAM-model. Each of the three methods will be calculated in every simulation.

Firstly, the “basis”-`gamboost`-model is fitted:

```
m <- gamboost(y ~  bbs("x1") + ... + bbs(x5) +
 bspatial(x.coord, y.coord),
 data = dat,
 control = boost_control(mstop = 200, nu = 0.2))
```

Depending on the simulation, each of the one to five covariates is modeled non-parametrically. This is done with the help of the P-Splines base-learner `bbs`(·). The “default”-settings for this base-learner are applied. This means that P-Splines of degree 3 with 20 knots and a penalty matrix based on the

second differences are used.
The spatial effect is fitted with the help of the `bspatial`(·) base-learner. The "default"-settings are used again. Concretely, this means that `bspatial`(·) base-learner relies on bivariate Tensorproduct-P-Splines for the estimation of the spatial effect. Note that the `bspatial`(·) base-learner is equivalent to the `bbs`(·) base-learner with degree 6. The penalty term is constructed by using the bivariate extensions of the univariate penalties in x and y directions.
Pre-simulations have shown that an adjustment of the hyperparameter is necessary. For this reason, the number of initial boosting iterations `mstop` and step size or shrinkage parameter `nu` are changed. The optimal m_{stop} iteration is calculated with the help of a 25-k-fold bootstrap. Concretely, the `mstop`-parameter is increased to 200. The step-length-parameter `nu` is also increased to 0.2.

Afterwards, a `gamboost`-model with the %ll%-operator modification is fitted. This model is called %ll%-model in the following.

```
mll <- gamboost(y ~ bspatial(x.coord, y.coord) +
 bbs("x1") + ... + bbs(x5) +
 bspatial(x.coord, y.coord) %ll% (bbs(x1) %+% ... %+% bbs(x5)),
 data = dat,
 control = boost_control(mstop = 200, nu = 0.2))
```

The covariates are fitted as in the "basis"-`gamboost`-model. In the `gamboost`-model with the %ll%-operator modification, the specification of the spatial effect differs from the "basis"-`gamboost`-model. The new %ll%-Operator is used. Concretely, the covariate effects are removed from the spatial effect. Details of the %ll%-Operator are shown in section 3.3.3. For each single base-learner of the new %ll%-operator-base-learner
`bspatial() %ll%[bbs() %+%...%+% bbs()]` the "default"-setting is used. This is used for a better comparison with the "basis"-`gamboost`-model.
Like in the "basis"-`gamboost`-model, pre-simulations have proved that an adjustment of the hyperparameter is necessary. For this reason, the number of initial boosting iterations `mstop` and step-length-parameter `nu` is changed. The initial `mstop`-parameter is increased to 200 and additionally the optimal m_{stop}

iteration is calculated with the help of a 25-k-fold bootstrap. The step-length-parameter nu is as well increased to 0.2.

Concluding, the pGAM-model is fitted.

```
obj <- pGAM(y, X, thresh = 0.95, thresh.type = "Ftest")
```

The pGAM-method works different than the previous two. It relies internally on the gam-algorithm which is provided by the mgcv-package in order to fit each of five covariates. Thus, the "default"-settings for the gam-algorithm are used.

4.2.2. Settings

As mentioned in section 4.1.2, there are many settings in the DGP to investigate the performance of the %ll%-operator and the strengths and weaknesses of the methods. After that, for a certain combination of these settings, the term "SET" will be used. The following SETs are passed:

- **N:** The sample size is set to 500 for each SET.

- **gridType:** Only an irregular grid ("irreg") is considered.

- **covType:** Both implemented covariance functions, exponential and Matèrn, are considered:
  In the case of a exponential covariance function, myTheta's value is 1. In contrast, with a Matèrn covariance function, myTheta's value is 4. Thus, the exponential covariance decreases quickly whereas the Matèrn covariance function decreases slowly.

- **SNRconcurvity:** Three different degrees of concurvity shall be considered. SNRconcurvity = 0.3 stands for "large" concurvity. SNRconcurvity = 1 represents "medium" concurvity. Compared with that, SNRconcurvity = 10 means "small" concurvity.

- **SNRspatial:** Three different degrees of SNRspatial shall be considered. SNRspatial = 0.1 means that a large part of the explainable variance corresponds to the covariate effect ("covariate >> spatial"). Contrary, SNRspatial = 10 means that the explainable variance corresponds to the spatial effect (“covariate << spatial”). SNRspatial = 1 means that the explainable variance corresponds equally to the covariate and spatial effect (“covariate = spatial”).

- **SNRepsilon:** Three different degrees of SNRepsilon shall be considered. SNRepsilon = 0.2 means that there is a lot of extra noise. This setting is named “noisy”. In contrary, SNRepsilon = 10 means that there is almost no noise and the data variance can be explained by the model. This setting is named “clear”. SNRepsilon = 1 stands for no extra noise and is named “normal”.

Thus, in total 54 SETs must be passed in each simulation. Moreover, each SET is repeated 10 times for each of the three simulations. This is done in order to calculate several performance measurements.

4.3. Results

In the following section the results from the three different simulation studies are presented. Firstly, a general descriptive analysis of the the data generating process is carried out. Additionally, the complete results and graphics can be found on the included CD-ROM.

4.3.1. Descriptive Analysis

Figure 4.2 visualizes one exemplary realization of a data generating process consisting of five covariates and 500 observations based on an irregular grid. The different “signal-to-noise”-ratios are equal to 1. The five available parametric functions for the smooth effect of the covariates are used. The left part of figure 4.2 shows the smooth function for the certain covariate. The selected parametric functions consists of different complexities. Although, the smooth function (1) and (4) are almost similar. The right part of the figure presents the relationship between a single covariate and the predictor variable $\boldsymbol{y}$. It is noticeable that the data center is located at the 0 in x-direction. Generally, less data are located at the edges.

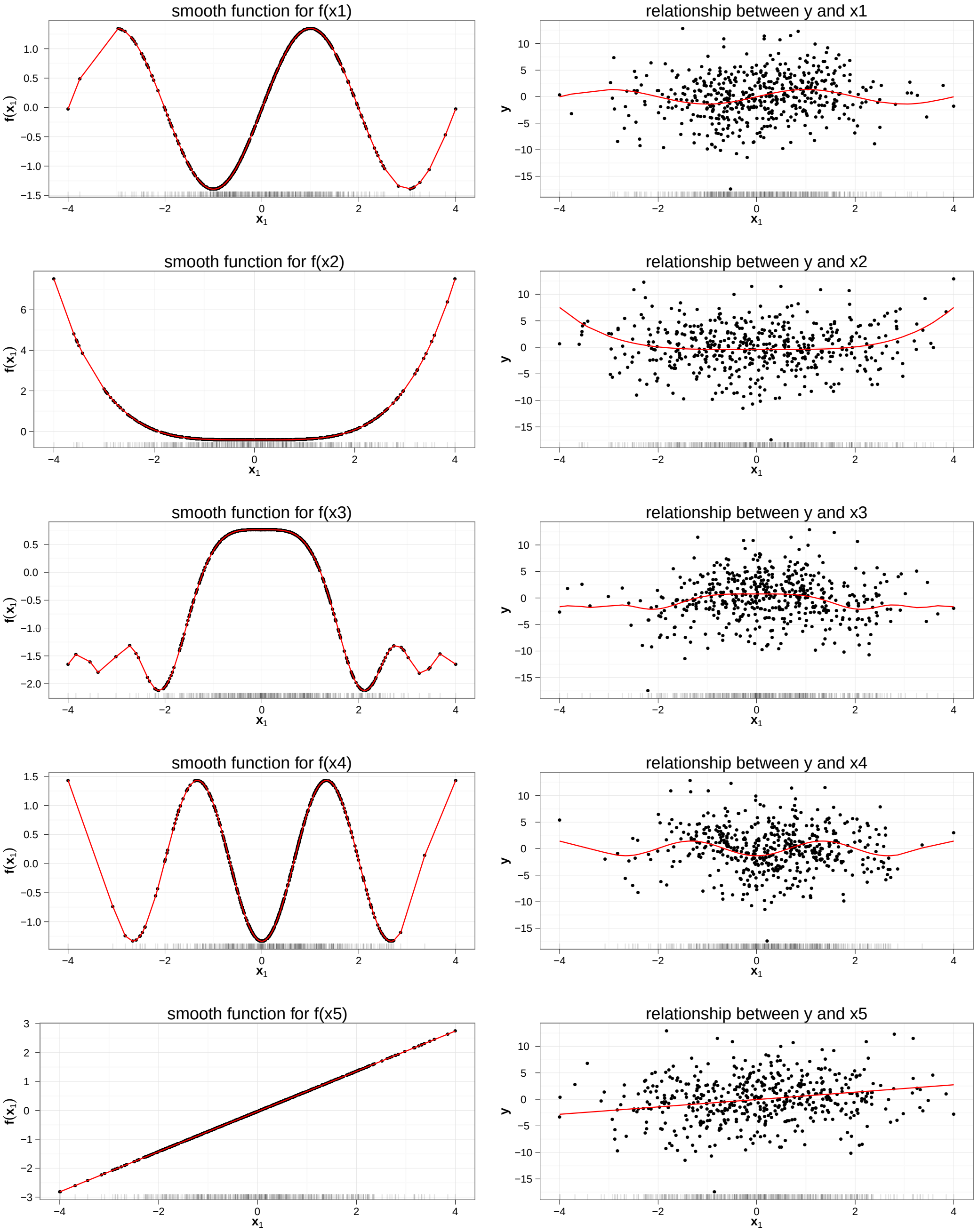

Figure 4.2.: Parametric Smooth Functions of the DGP

Figure 4.3 displays two actually realized spatial effects of the data generating process. The upper part of figure 4.3 presents a realization of a spatial random field with exponential covariance function. In this case, the `myTheta` parameter is chosen equal to 1. This covariance function corresponds to the gold curve

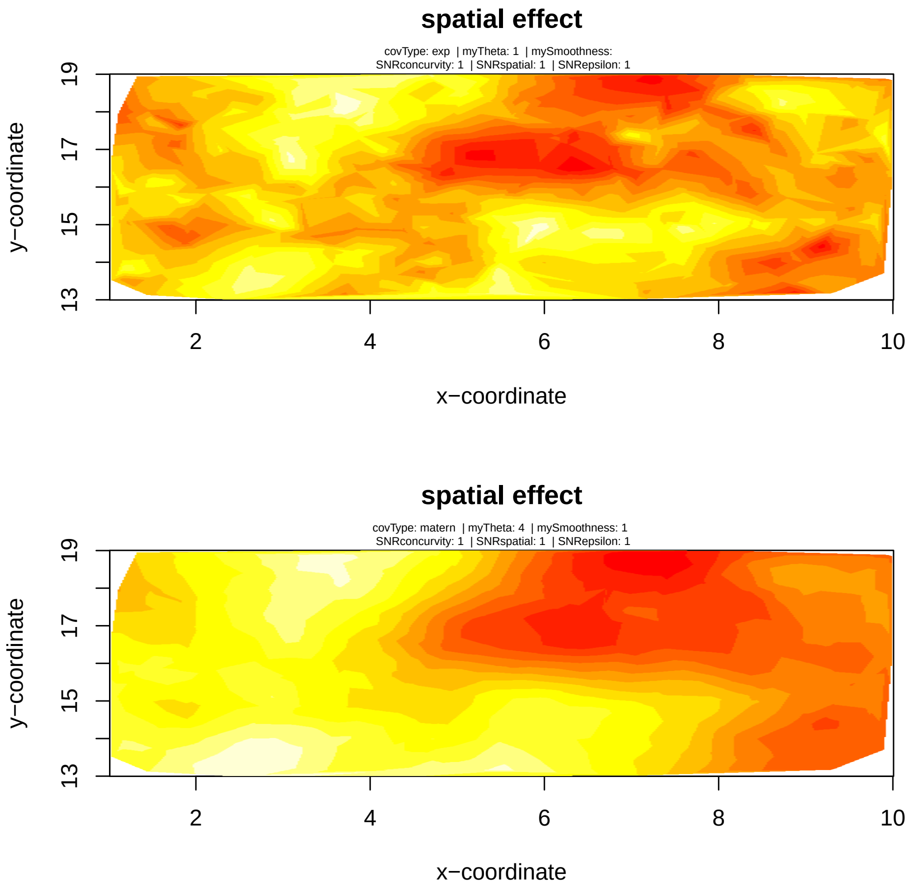

Figure 4.3.: Spatial Effect of the DGP: A realization of a spatial random field with exponential covariance function in the upper part. In contrary, a realization with Matèrn covariance function in the lower part.

in the left part of figure 4.1. This chosen covariance function decreases very rapidly. This is reflected in the visualization. The colors are not very smooth and change fastly. Thus, there is not a very high autocorrelation within the coordinates.

In contrary, the lower part of figure 4.3 presents the realization of a spatial random field with Matèrn covariance function. For this spatial effect the `myTheta` parameter is equal to the blue curve in the right part of figure 4.1. The chosen Matèrn covariance function decreases very slowly. Consequently, there is a very large autocorrelation within the coordinates. This can be recognized by the fact that the colors are smooth and change slowly in the visualization.

4.3.2. Simulation 1: One covariate

This first simple simulation study only consists of one covariate and an additional spatial effect to investigate whether the %ll%-method works at all or cases in which the %ll%-model works better than other methods. That is done before complicated models are studied once. The spatial effect is modeled with an exponential covariance function and once with a Matèrn covariance function. The two different spatial effects are visualized in figure 4.3.

The following plots are all constructed in the same way. Every plot is divided into six parts. The three different concurvity-levels (SNRc) are found in vertical direction and the two different covariance functions are located in horizontal direction. On the x-axis of the respective part, the associated and analyzed "signal-to-noise"-levels are ablated. On the logarithmic y-axis of the respective part, the corresponding Root Mean Squared Error (RMSE) is plotted. The RMSE is a measure for the deviation between the estimated and the true values.
The "basis"-model is represented by the red color. With the green color the %ll%-model is depicted and the blue color stands for the pGAM-model.
The points in the plot represent the median of the RMSE of the 10 replications. The inter-quartile range is indicated by the point range.

Figure 4.4 illustrates the result of the model fit of $\boldsymbol{y}$ by the three different methods. At first glance, two settings can be identified. In case of the exponential covariance function the overall fit with the %ll%-model is clearly better than with the other two models. This happens when `SNRs` is equal to 0.1 ("covariate >> spatial"). That means that a large part of the explainable variance corresponds to the covariate effect. The %ll%-model is able to recognize this much better than the other two models. This is also supported by figure 4.5 which shows the adaptation of the first covariate. This adaptation is the best one. The median of the %ll%-model is clearly below the median of the other models.

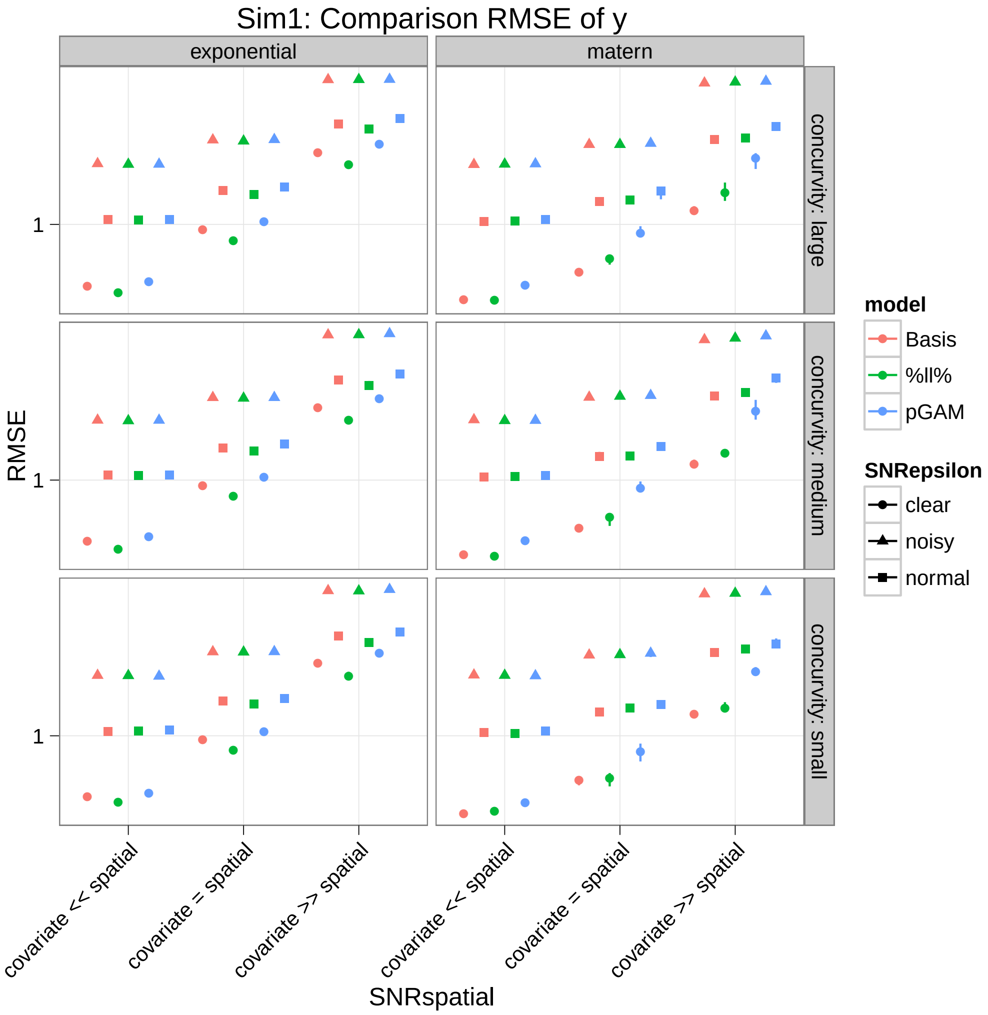

Figure 4.4.: Simulation 1: Comparison RMSE (median & inter-quartile range) of y; “Basis”-Model (red), %ll%-Model (green) and pGAM-Model (blue).

Generally, it is noticeable that the RMSE is much higher in extreme settings (“noisy” or “covariate >> spatial”) than in other settings. Tendencially, the fit is worse in “noisy” settings (`SNRe` = 0.2). For the boosting models this can also be explained by the m_{stop}. As mentioned in section 2.4.3, the hyperparameter m_{stop} is very important. Figure 4.7 displays the comparison of the m_{stop}. Note that m_{stop} is small in “noisy” settings (`SNRe` = 0.2). Thus, the boosting-models have almost no chance to capture the complex model structure correctly. Additionally, Bühlmann and Hothorn [2007] point out that overfitting is possible if the boosting algorithm is stopped too early. If there is a “clear” setting and the data variance can be explained by the model (`SNRe = 10`), the %ll%-model performs slightly better than the other two.

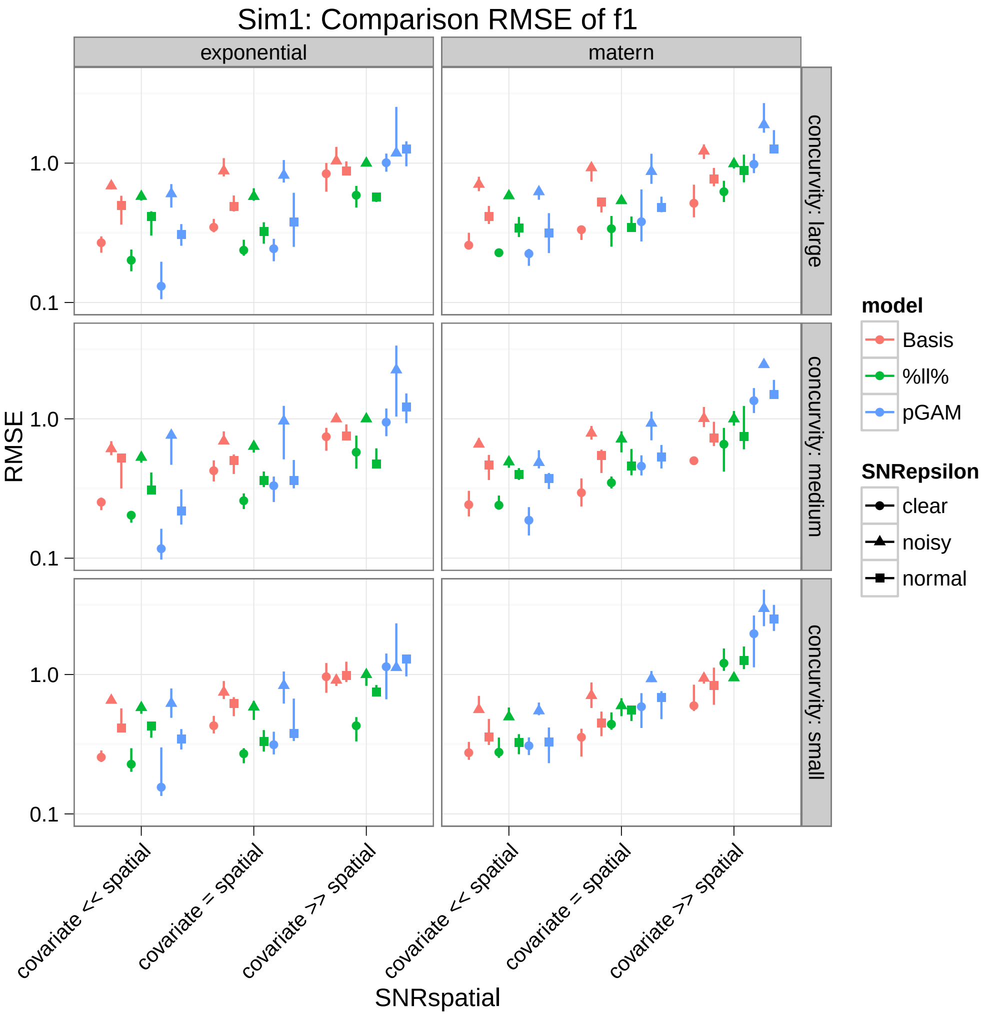

Figure 4.5.: Simulation 1: Comparison RMSE (median & inter-quartile range) of $f_1(x)$; “Basis”-Model (red), %ll%-Model (green) and pGAM-Model (blue).

The presence of a Matèrn covariance function and a `myTheta`-parameter with value 4 ensures that there is a very high spatial autocorrelation in the data. Considering the figure, one setting can be identified to be fitted considerably worse than the others. It is the case with “large” concurvity (`SNRc` $= 0.1$). Estimating this setting correctly, it is a challenge for all three models. The pGAM-model performs the worst. The two boosting-models differ little. Generally, extreme settings tend to higher RMSE. The reason for that is located in the early m_{stop} iteration shown by figure 4.7. It is of interest, if there are differences in the fit of the variables although the overall fit is similar.

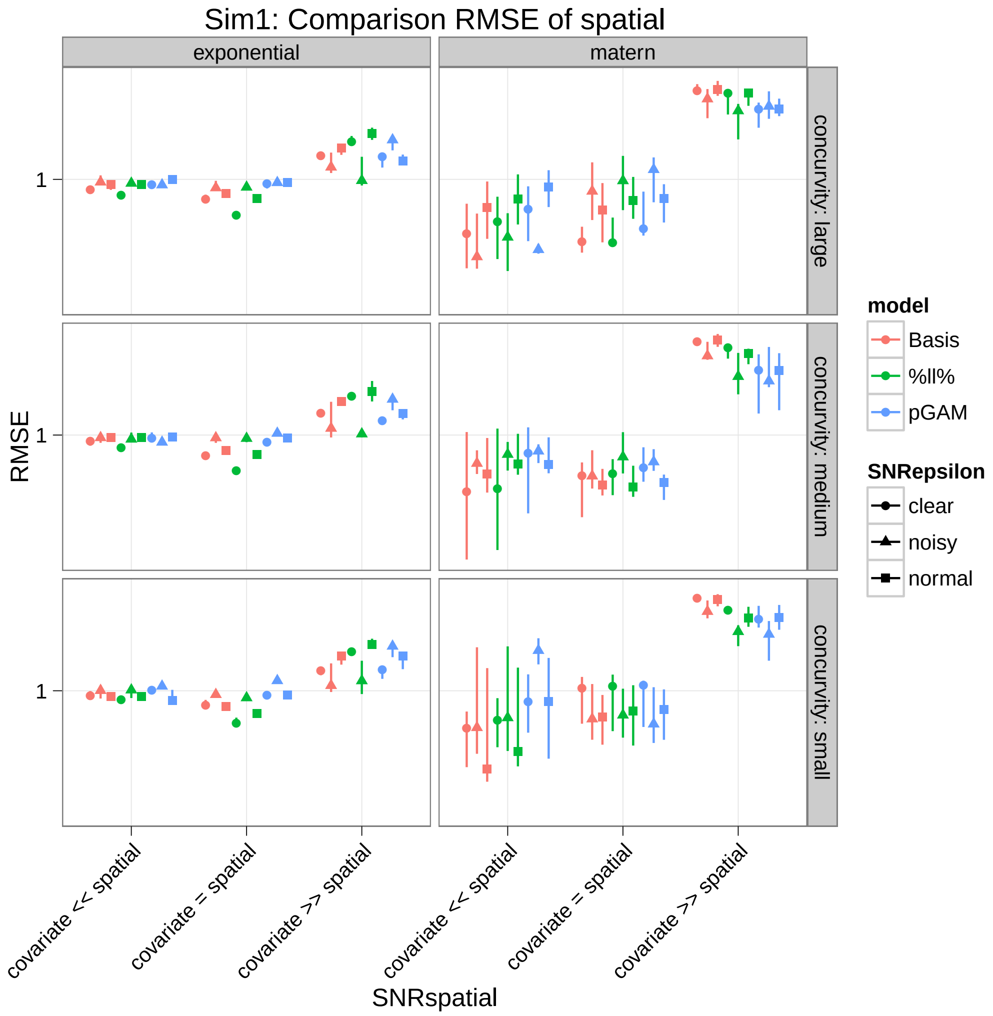

Figure 4.6.: Simulation 1: Comparison RMSE (median & inter-quartile range) of spatial effect; “Basis”-Model (red), %ll%-Model (green) and pGAM-Model (blue).

Figure 4.5 shows the fit of the first covariate $\boldsymbol{x}_1$. In the case of exponential or Matèrn covariance function the pGAM-model performs either considerably worse or at most as good as the other two models in all settings. The pGAM-models also have much larger range, especially in “noisy” settings (`SNRe = 0.2`). The pGAM-model works really bad in the setting “covariate >> spatial” (`SNRs` $= 0.1$). With extreme settings (“noisy” or “covariate >> spatial”) the RMSE for the basis-model is almost equal to 1. Instead of the covariate effect, the spatial effect is chosen by the model in these cases. Generally, in case of “large” or “medium” concurvity (`SNRc` = 1 or `SNRc` = 0.3) the %ll%-model performs better or at least as good as the basis-model. Thus, there are differences in the fit of the single covariate although the overall fit is very similar.

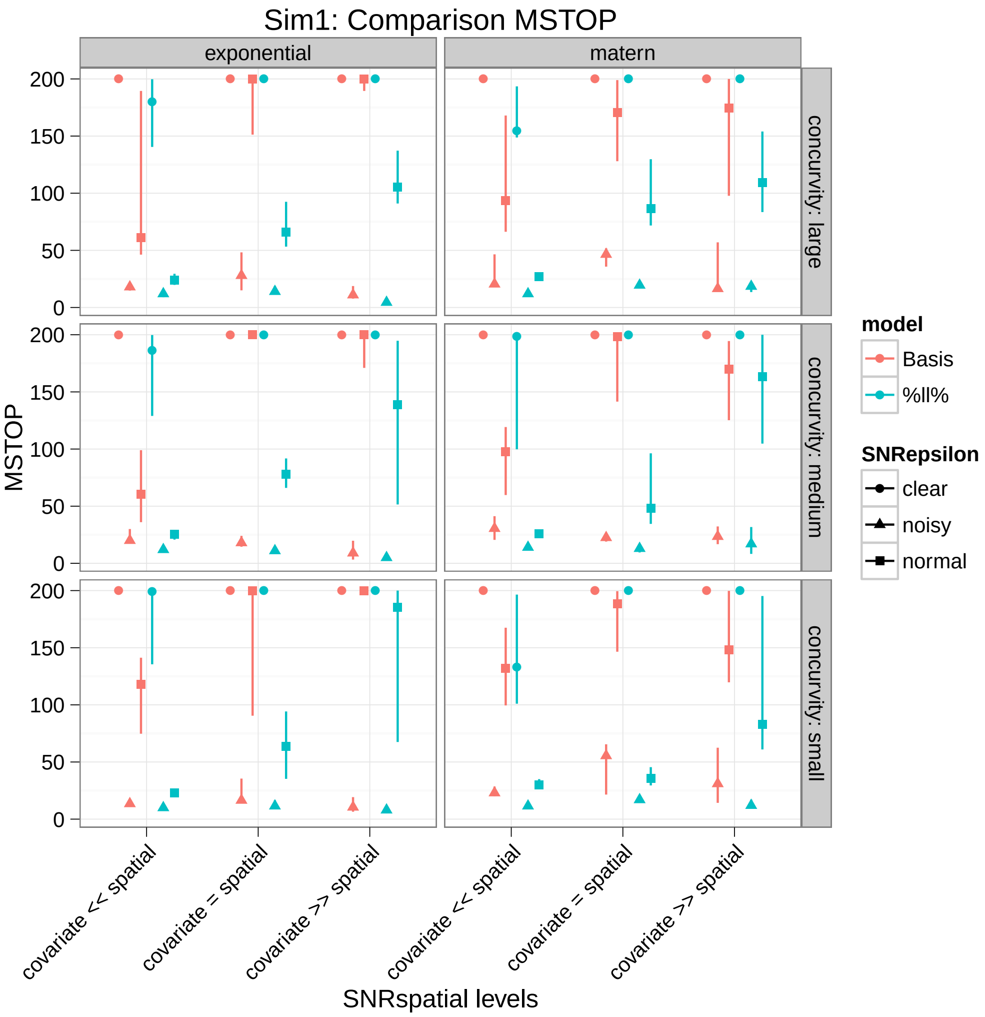

Figure 4.7.: Simulation 1: Comparison m_{stop} (median & inter-quartile range); “Basis”-Model (red) and %ll%-Model (green).

The fit of the spatial effect is presented by figure 4.6. For the exponential covariance function it is noticeable that settings in which are excellently estimated by the %ll%-model, the fit of the spatial effect is considerably worse than for the other models. This is due to the construction of the spatial effect in this model. The spatial effect has similarities to the task of a dustbin. The spatial effect only explains the variance which can not be explained by the covariates. In the other settings the fit of the spatial effects is quite similar. In the big range in settings with Matèrn covariance, it can be seen that these settings are much harder to estimate. Especially, in the case where the covariate effect dominates the spatial effect, the %ll%-model provides the clearly better fit. The adaptation in the other settings is pretty similar.

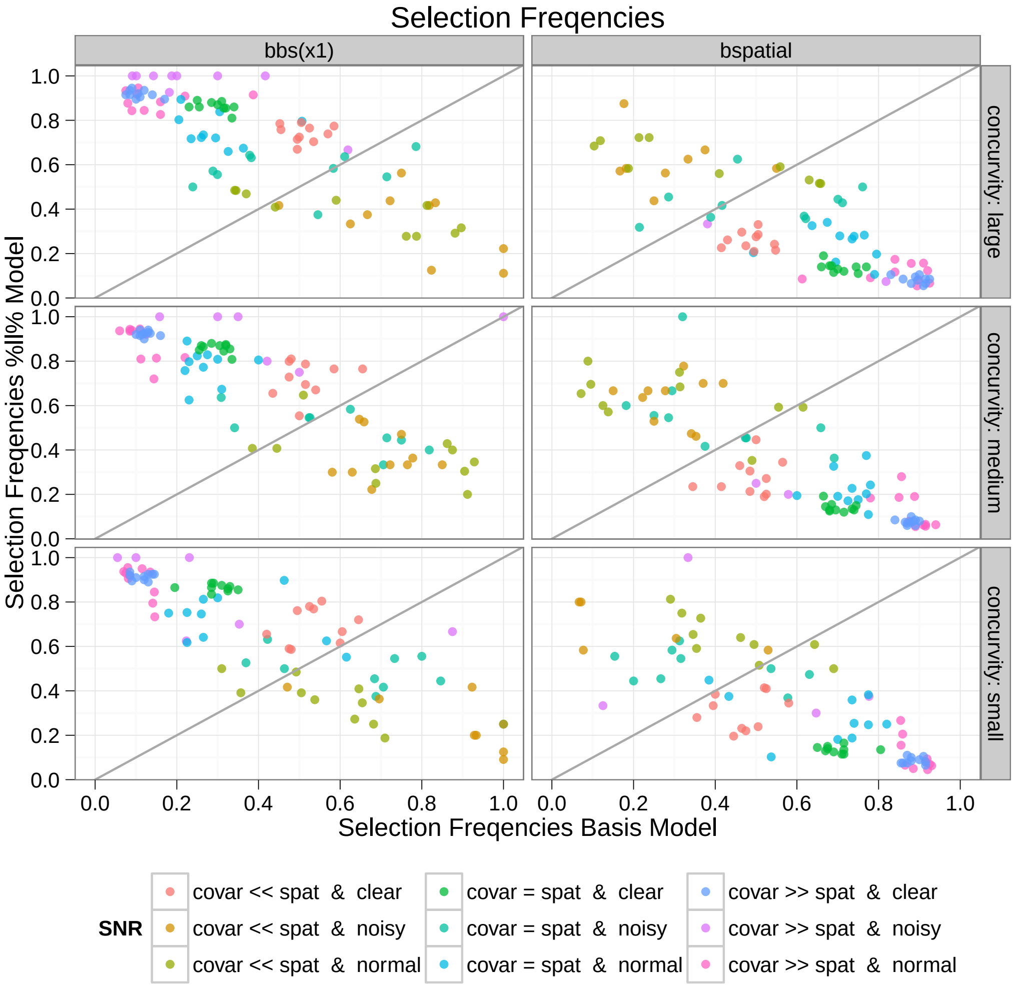

Figure 4.8.: Simulation 1: Comparison Selection Frequencies; exponential covariance.

Figure 4.7 displays the comparison of the m_{stop} hyperparameter of the two boosting models. The pattern for both covariances is very similar. As mentioned before, the early stopping in the "noisy" settings (`SNRe` $= 0.2$) makes it quite difficult for the boosting-models to capture correctly the complex model structure.

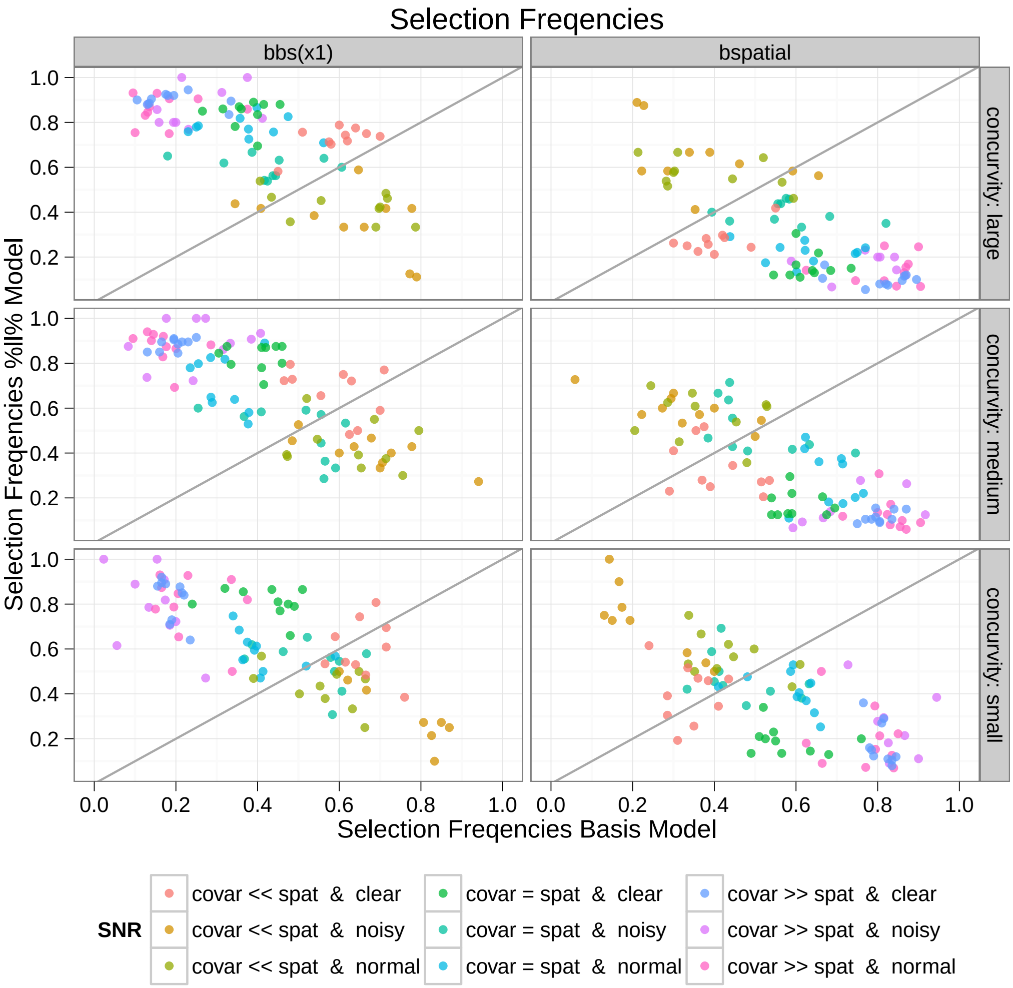

Figure 4.9.: Simulation 1: Comparison Selection Frequencies; Matèrn covariance.

The selection frequencies of both boosting models for the exponential covariance function are shown in figure 4.8. For the Matèrn covariance function the selection frequencies are presented in figure 4.9. As mentioned previously, the selection frequencies of the %ll%-model are clearly better than those of basis-model. Principally, the %ll%-model selects the `bspatial` base-learner only in settings with a strong spatial effect (`SNRs` = 10) in contrast to the basic model. Thus, the %ll%-model constitutes the “truth” in a much better way. The %ll%-

operator provides the separation of the spatial autocorrelation between spatial and covariate effect. It does not matter how strong the concurvity actually is. In contrary, the basis-model in settings where the covariate effect dominates (`SNRs` $= 0.1$), usually the `bspatial` base-learner is selected falsely.

4.3.3. Simulation 2: Two covariates

With the help of simulation 2 and 3, the performance of the %ll%-operator is investigated by an increasing number of covariates. Simulation 2 consists of 2 covariates and an additional spatial effect. The result of simulation 2, illustrated in figure 4.10, has a similar pattern as simulation 1. In the exponential covariance case the setting "covariate >> spatial" is particularly striking. This setting has a higher RMSE than the others. However, this setting is estimated much better by the %ll%-model than by the other two ones. Generally, the

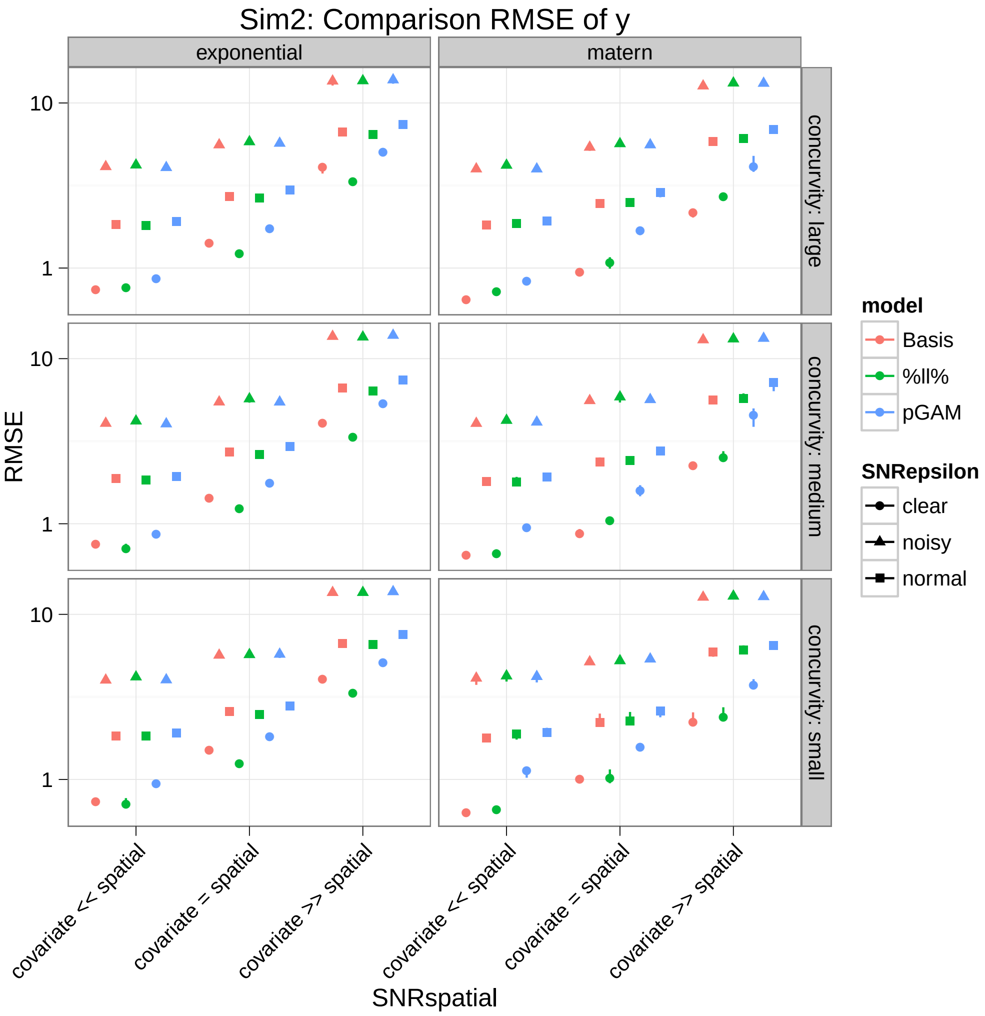

Figure 4.10.: Simulation 2: Comparison RMSE (median & inter-quartile range) of y; "Basis"-Model (red), %ll%-Model (green) and pGAM-Model (blue).

adaptations of "noisy" settings ($\texttt{SNRe} = 0.2$) are bad and result in higher RMSE. In the Matèrn covariance case, there are no clear differences between the two boosting models. As in the exponential covariance case, the setting "covariate >> spatial" is very challenging and therefore results in a higher RMSE. Note that the adaptation with the help of the pGAM-model is worse in the two covariance cases than in the boosting models. Although, the overall fit differs little, it introduces the questions if there are visible differences in the fit of the individual covariates.

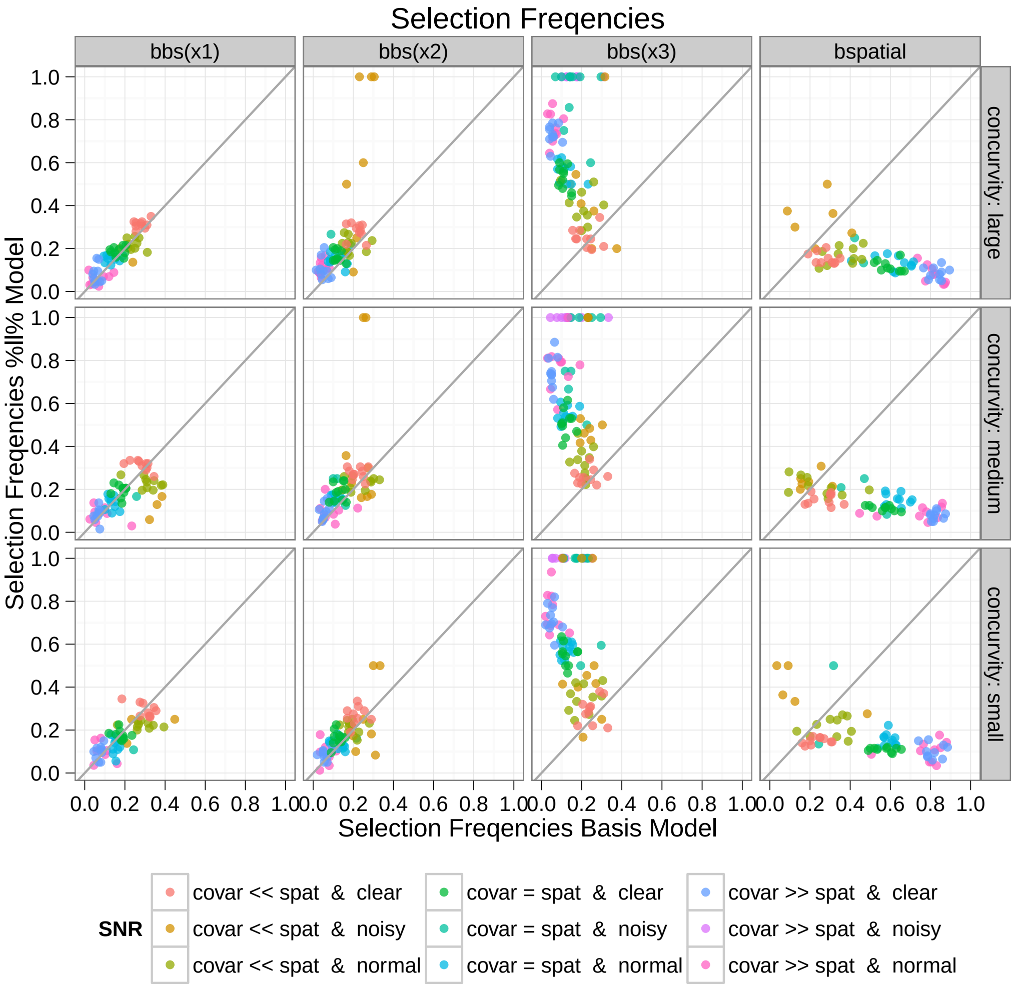

Figure 4.11.: Simulation 2: Comparison Selection Frequencies; exponential covariance

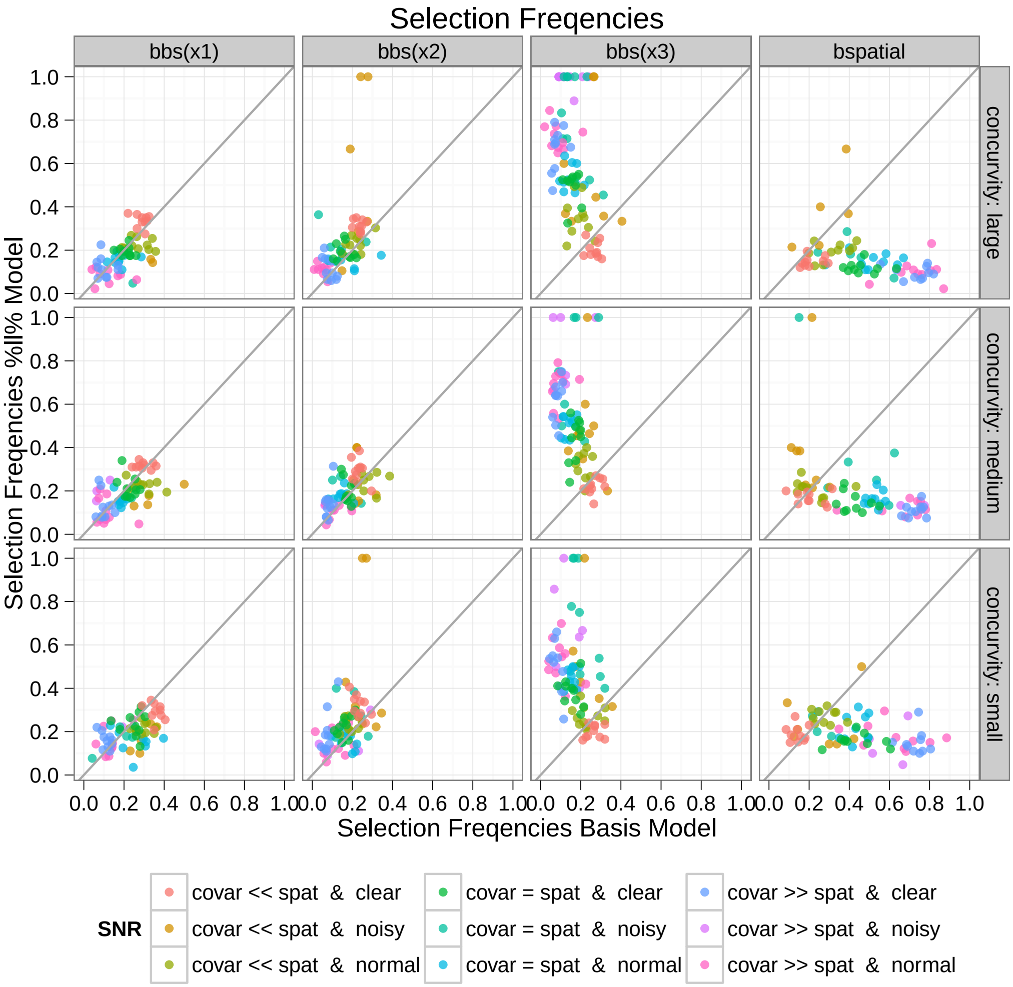

Figure 4.12.: Simulation 2: Comparison Selection Frequencies; Matèrn covariance

To find an answer to this question, use the drawing of figure 4.11 and 4.12. Figure 4.11 presents the selection frequencies for the exponential covariance case. However, figure 4.12 shows the selection frequencies for the Matèrn covariance case. No great difference can be found when the two figures are compared. Thus, the type of the covariance function does not affect the %ll%-operator.

However, large differences, concerning the selection frequencies between the two boosting models, can be found. The first base-learner `bbs(x1)`, it is tendencially preferred by the boosting model. On the other hand, base-learner

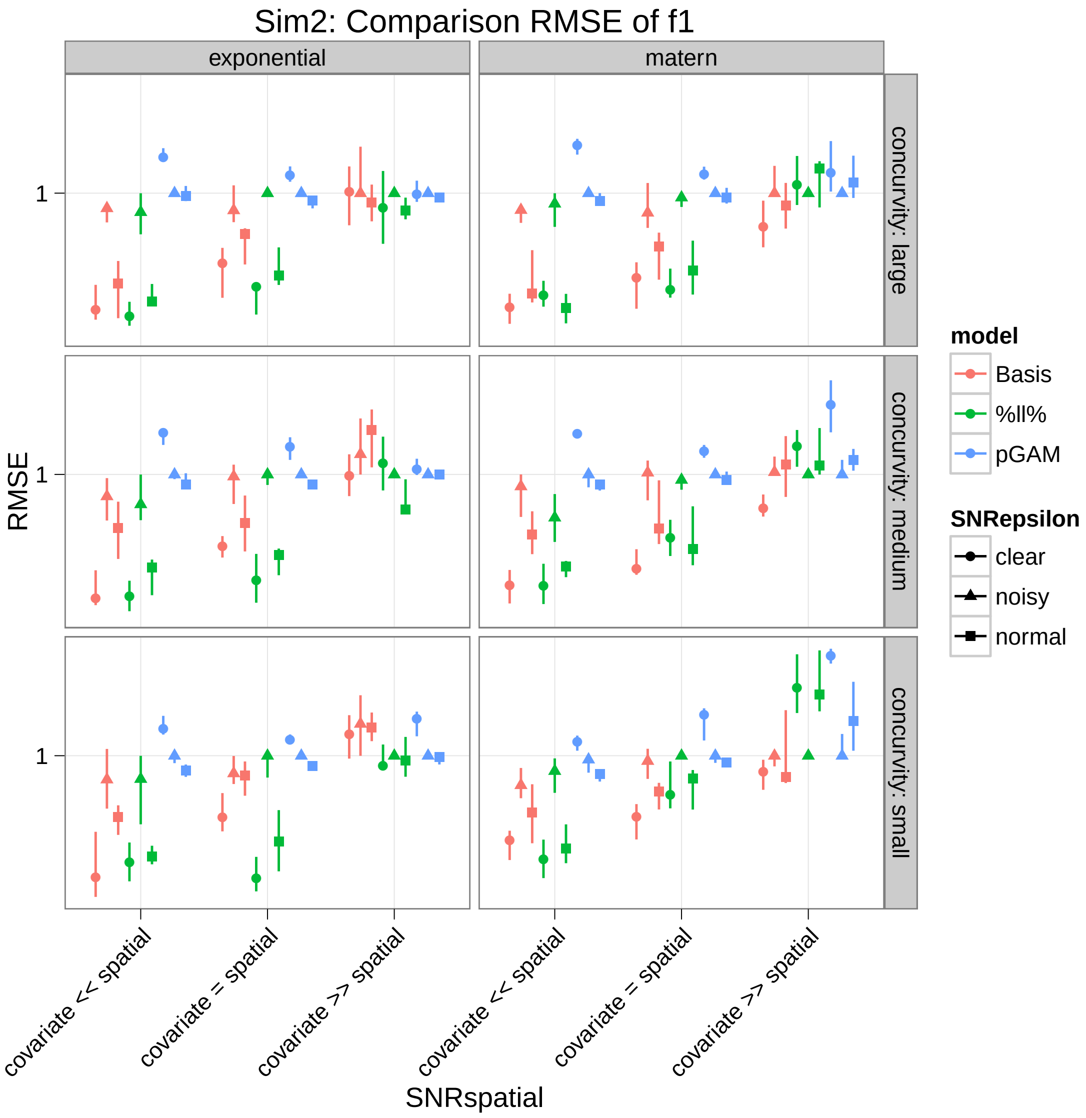

Figure 4.13.: Simulation 2: Comparison RMSE (median & inter-quartile range) of $f_1(x)$; “Basis”-Model (red), %ll%-Model (green) and pGAM-Model (blue).

`bbs(x2)` and `bbs(x3)` are preferred and base-learner `bbs(x3)` is selected most often by the %ll%-model. In contrary, the basis-model identifies the `bspatial` base-learner as the most important one. Thus, it can be seen that the %ll%-operator does a great job and provides the separation of the spatial autocorrelation between spatial and covariate effect again. The larger the concurvity is, the less likely the `bspatial` base-learner is selected in the %ll%-model. On the other hand, the basis-model tries to explain almost all of the variance with the `bspatial` base-learner. Thus, the question if the selection frequencies affect the fit of individual covariate is introduced.

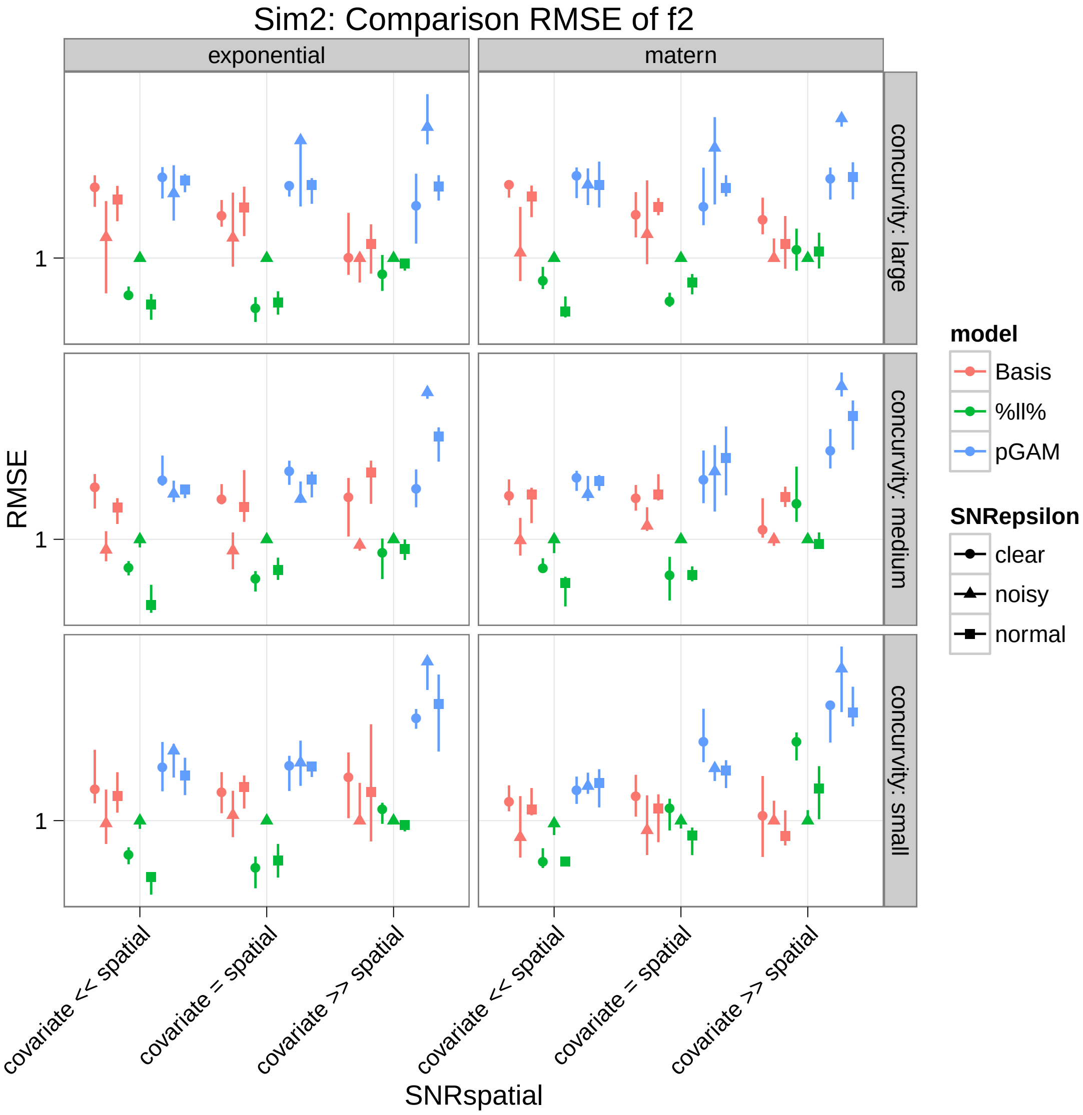

Figure 4.14.: Simulation 2: Comparison RMSE (median & inter-quartile range) of $f_2(x)$; “Basis”-Model (red), %ll%-Model (green) and pGAM-Model (blue).

Figure 4.13 presents the fit of the first covariate. Tendencially, concerning the selection frequencies this covariate is preferred by the basis-model, even though, the %ll%-model performs a bit better than the basis-model and much more better than the pGAM-model. Especially, the pGAM-model has great difficulties with the setting “covariate >> spatial” (SNRs $= 0.1$) again. In all

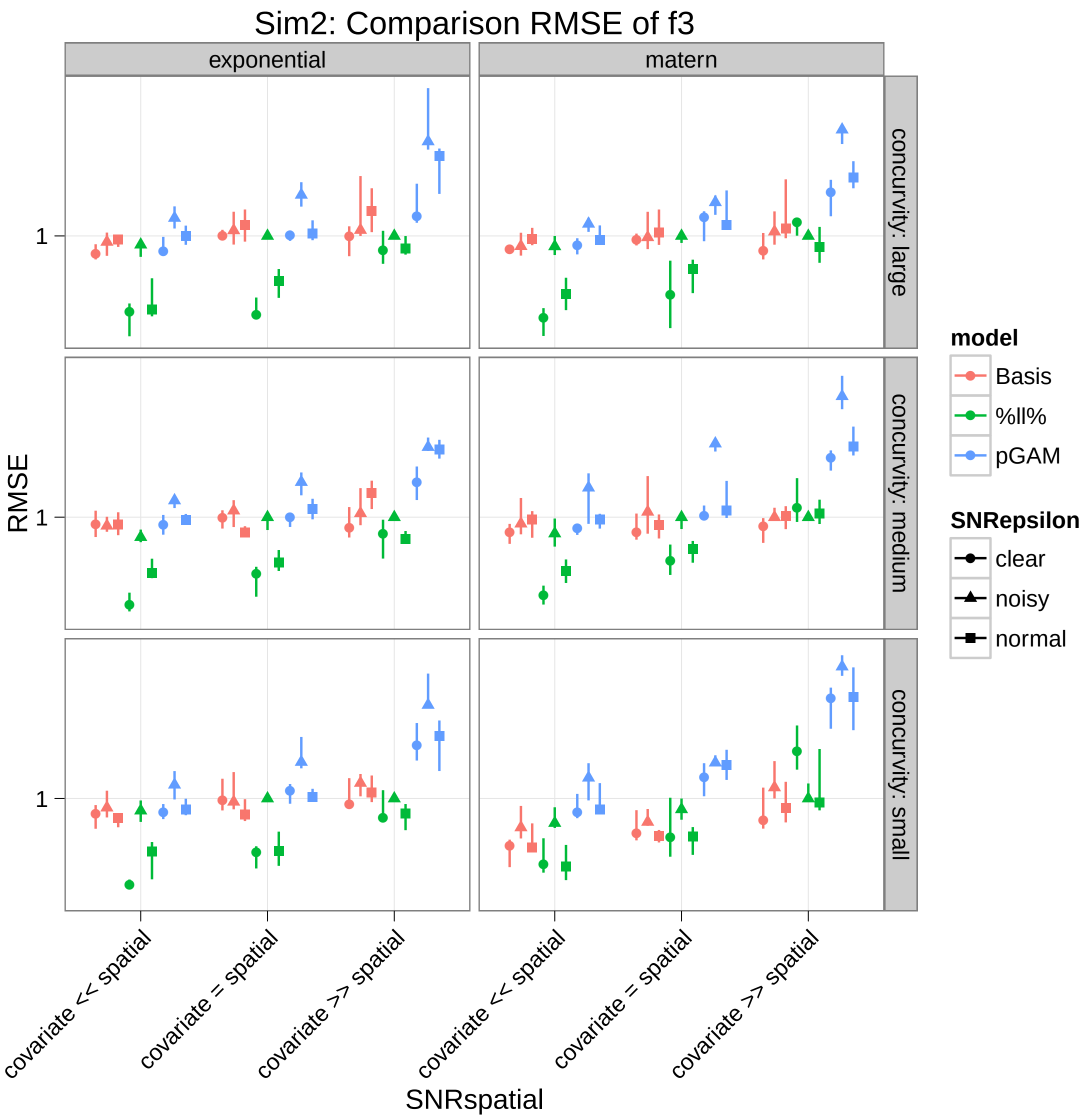

Figure 4.15.: Simulation 2: Comparison RMSE (median & inter-quartile range) of $f_3(x)$; “Basis”-Model (red), %ll%-Model (green) and pGAM-Model (blue).

other cases, the pGAM-model is even worse than the boosting models. For the fit of the first covariate, the kind of the covariance function seems to be relatively unimportant. The results do not differ between the two covariance functions.

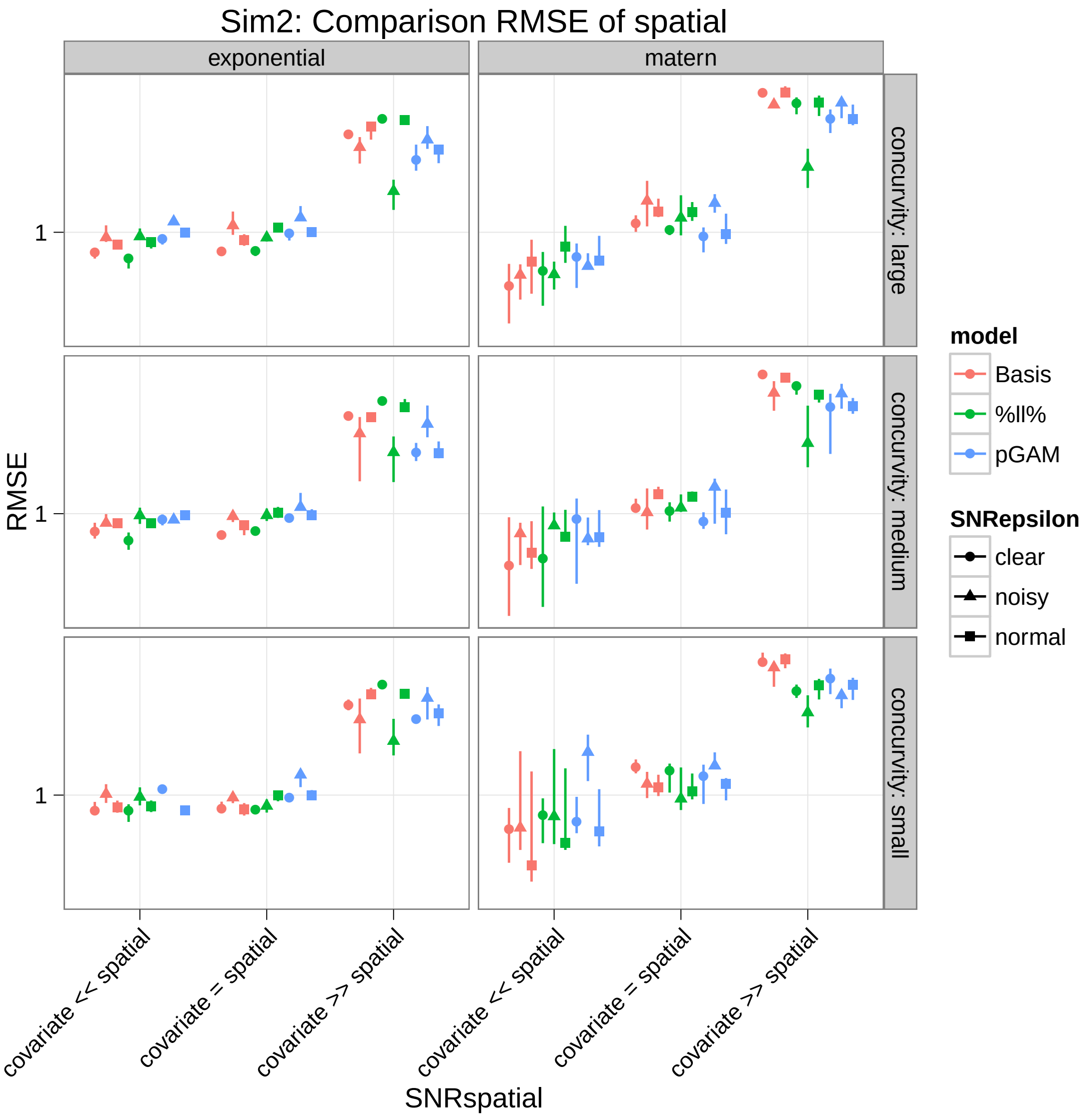

Figure 4.16.: Simulation 2: Comparison RMSE (median & inter-quartile range) of spatial effect; “Basis”-Model (red), %ll%-Model (green) and pGAM-Model (blue).

Figure 4.14 shows the fit of the second covariate. According to the selection frequencies, this covariate is clearly preferred by the %ll%-model. This can be seen in the adjustment. The fit of the second covariate is definitely best with the %ll%-model, no matter of the type of the covariance function. Only in the “noisy” setting (`SNRe` = 0.2), the two boosting models are about the same excellence. In all settings, the pGAM-model provides the worst fit in turn.

The fit of the third covariate is presented in figure 4.15. Mainly, this covariate was chosen by the %ll%-model as shown in figure 4.11 and 4.12. This covariate

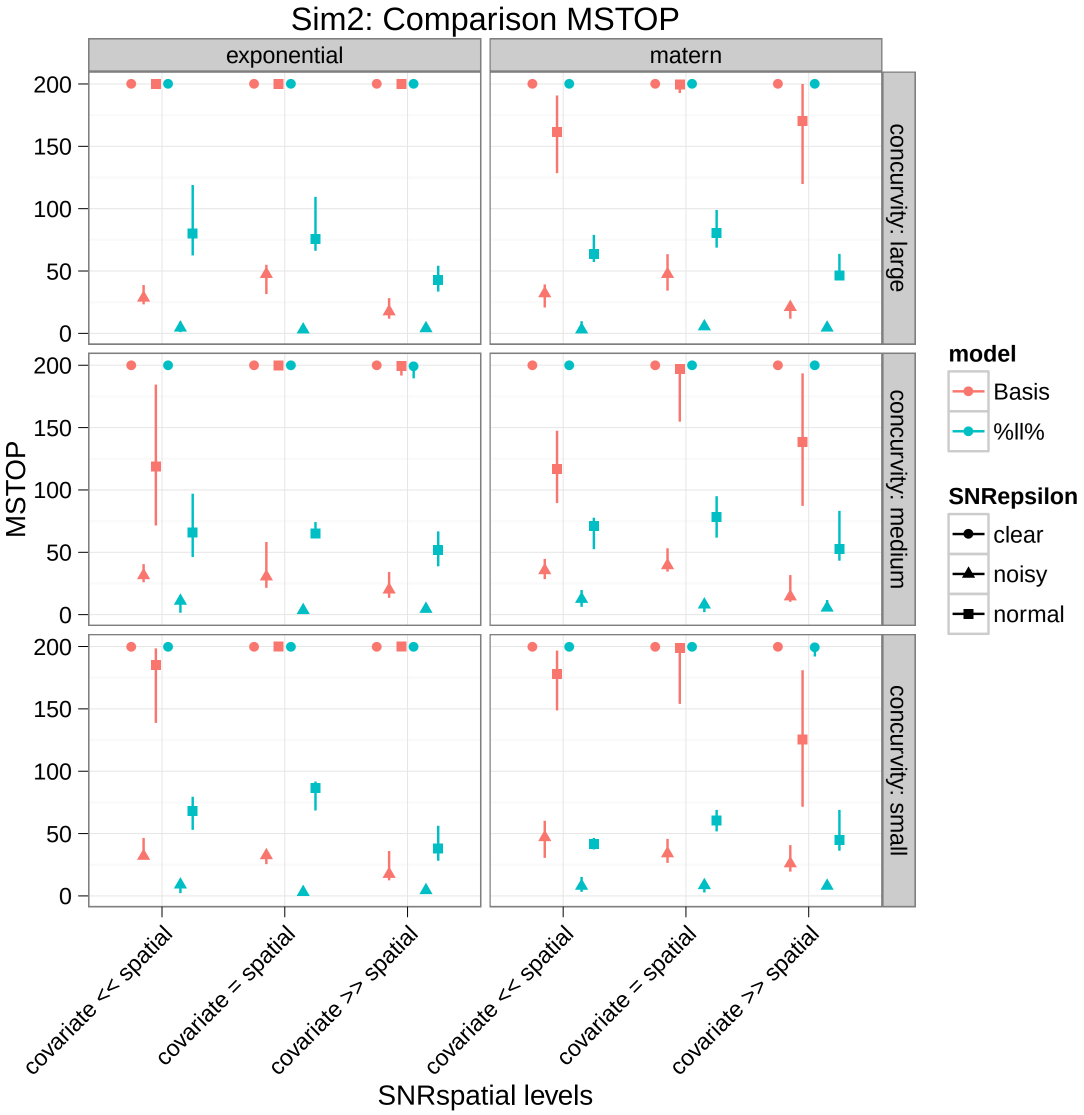

Figure 4.17.: Simulation 2: Comparison m_{stop} (median & inter-quartile range); “Basis”-Model (red) and %ll%-Model (green).

is fitted best by the %ll%-model, no matter of the degree of the concurvity and the type of the covariance function. Again, only in the “noisy” setting (`SNRe`

= 0.2), the two boosting models are about the same. Especially, the pGAM-model has great difficulties with the setting “covariate >> spatial” (`SNRs` = 0.1).

Considering the fit of the spatial effect, the setting “covariate >> spatial” (`SNRs` = 0.1) is particularly striking in figure 4.16. This setting is fitted worse than the other ones. In the other settings and the exponential covariance function, there are no clear differences between the models. In contrary, in case of Matèrn covariance function, the range of the RMSE of all three models is much greater. In the challenging setting “covariate >> spatial” (`SNRs` = 0.1) the %ll%-model performs clearly better than the other two.

Figure 4.17 displays the comparison of the m_{stop} hyperparameter of the two boosting models. The pattern for both covariances is similar. Again, the early stopping in the “noisy” settings (`SNRe` = 0.2) makes it relatively difficult for the boosting-models to capture the complex model structure correctly [Bühlmann and Hothorn, 2007]. This is the reason for no clear differences of both models in this setting.

4.3.4. Simulation 3: Five covariates

In simulation 3 the number of covariates increases more than in simulation 2. Thus, that is used an own parametric smooth function with different complexity for every single covariate. As previous simulations have already shown, the selection frequencies of the two boosting models are not affected by the type of the covariance function. Figure 4.18 and 4.19 show a very similar pattern for the exponential and Matèrn covariance. However, the high selection frequency for base-learner `bbs(x5)` in the %ll%-model is very noticeable,

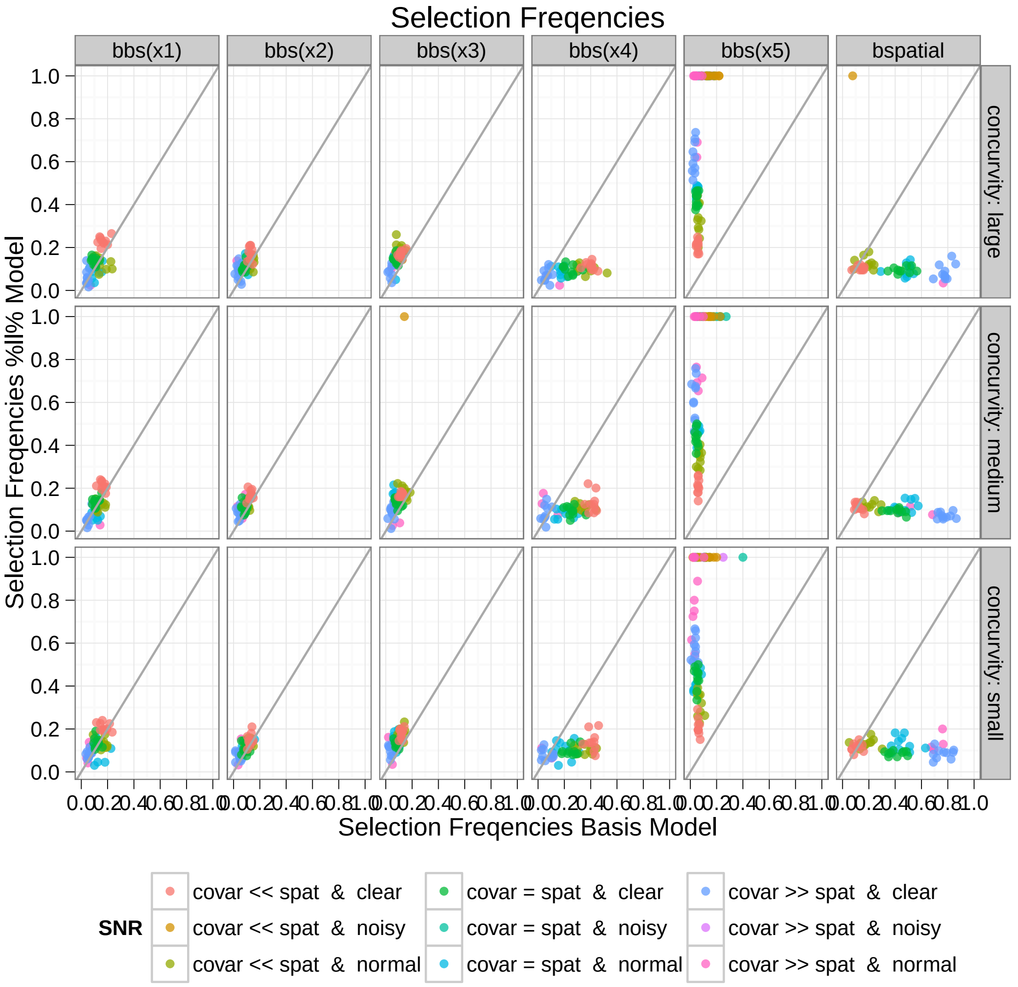

Figure 4.18.: Simulation 3: Comparison Selection Frequencies: exponential covariance; "Basis"-Model (red) and %ll%-Model (green).

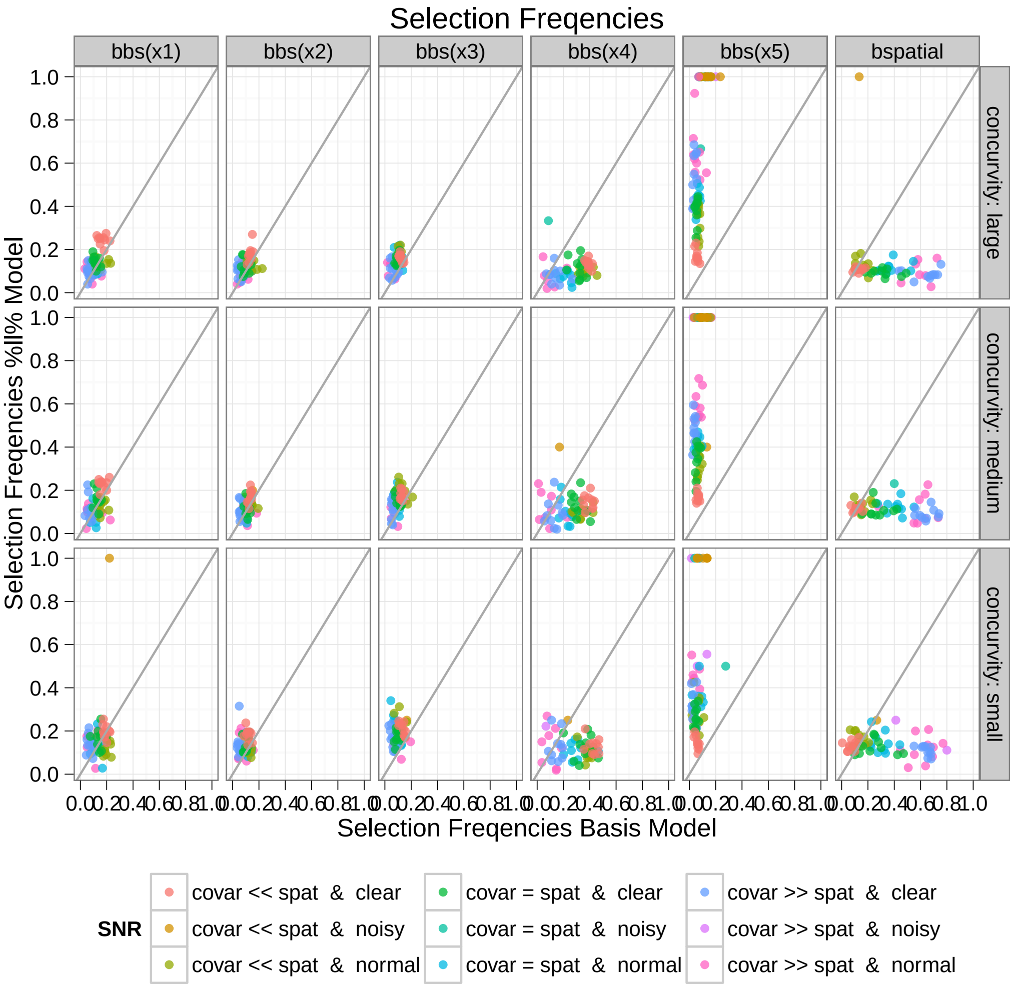

Figure 4.19.: Simulation 3: Comparison Selection Frequencies: Matèrn covariance; “Basis”-Model (red) and %ll%-Model (green).

no matter of concurvity level. In contrary, this base-learner is not chosen by the basis-model at all. Focusing on this, it can be observed that this mainly affects the totally “noisy” settings (`SNRe` $= 0.2$). The m_{stop} provides more help. Figure 4.20 shows a very small m_{stop} for the “noisy” setting. Thus, again the boosting-models have no chance to capture the complex model structure correctly [Bühlmann and Hothorn, 2007]. Therefore, the strange selection frequencies for this base-learner are explainable. Compared to that, the basis-model identifies the `bspatial` base-learner as most important, no matter of the concurvity level. The %ll%-model only chooses the `base-learner` in settings

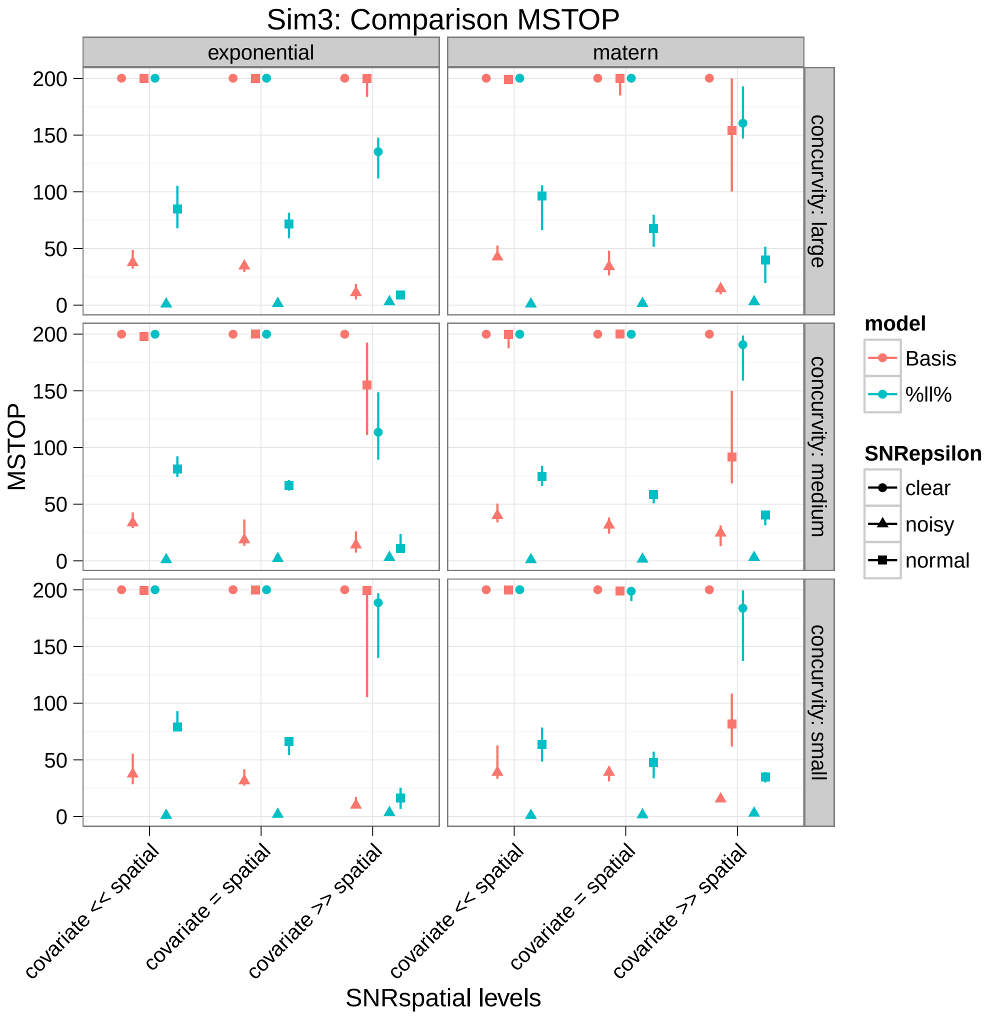

Figure 4.20.: Simulation 3: Comparison m_{stop} (median & inter-quartile range); "Basis"-Model (red) and %ll%-Model (green).

which are dominated by the spatial effect (`SNRs` = 0.1). Again, it can be seen that the %ll%-operator does a good job. After that, the `bbs(x4)` base-learner is the second most important one.

Generally, the m_{stop} pattern does not clearly differ from the previous simulations. Tendencially, the %ll%-model has a smaller m_{stop} than the basis-model. In the "noisy" settings (`SNRe` = 0.2) the small m_{stop} affects the results as shown by the selection frequencies or as later shown by the fit of single covariates.

With increasing model complexity it is now to be investigated, how well the models perform overall and how well the single covariates are fitted. As mentioned above, the small m_{stop} is responsible for the high RMSE in the “noisy” settings. Generally, the type of the covariance function plays a secondary part for the overall fit. Again figure 4.21 shows that the overall fit has a similar pattern to the previous simulations. The type of the correlation function is not an important issue. There are only few cases where are clear differences between the models. The %ll%-model has a slight advantage in settings dom-

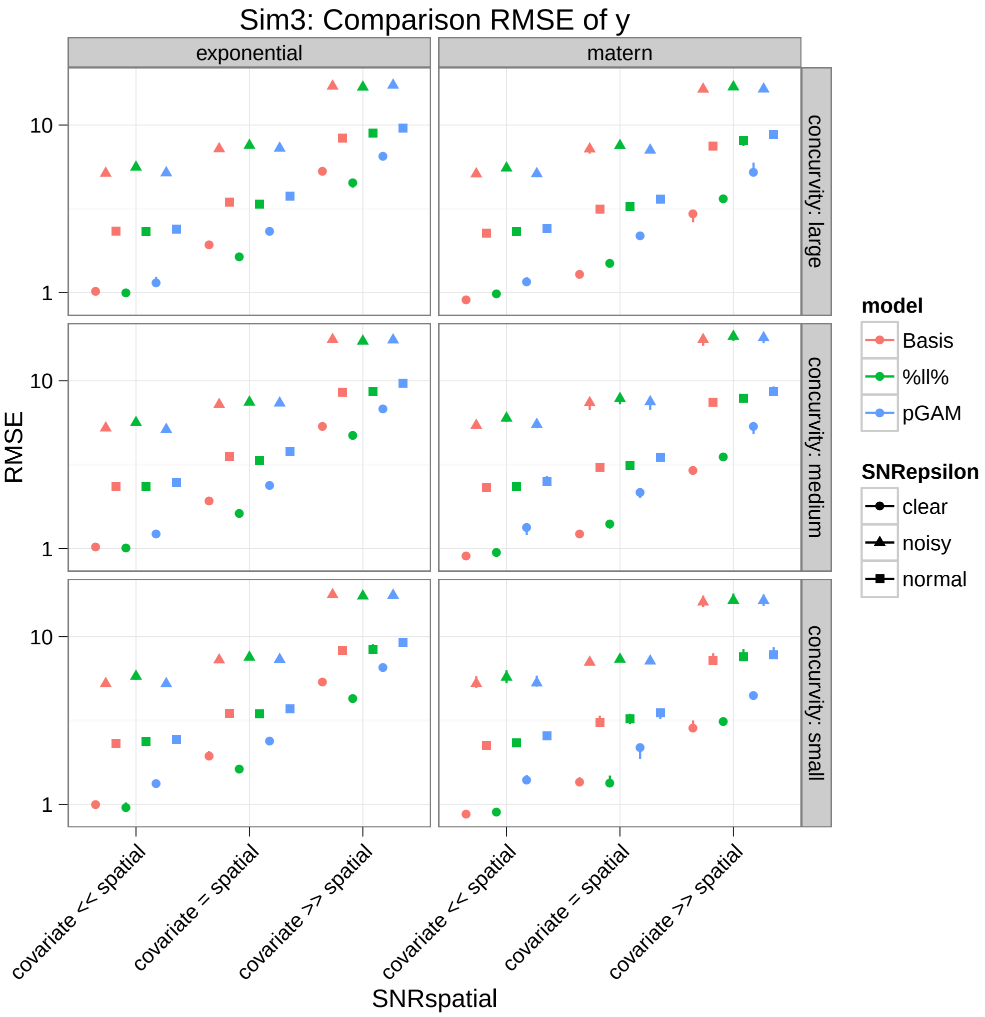

Figure 4.21.: Simulation 3: Comparison RMSE (median & inter-quartile range) of y; “Basis”-Model (red), %ll%-Model (green) and pGAM-Model (blue).

inated by the covariate effect (SNRs $= 0.1$).

The individual fits of the covariates are analyzed. Figure 4.22 presents the adaptation of the first covariate. Thus, it can be seen clearly that the pGAM performs worse than the boosting-models, no matter of the strength of the concurvity. Especially, the pGAM model performs worse in settings with a high weight on the covariates (SNRs $= 0.1$). The adaptation will be even worse if the covariance type switches from exponential to Matèrn. There are marginal differences between the %ll%-model and the basis-model. Tendencially, the

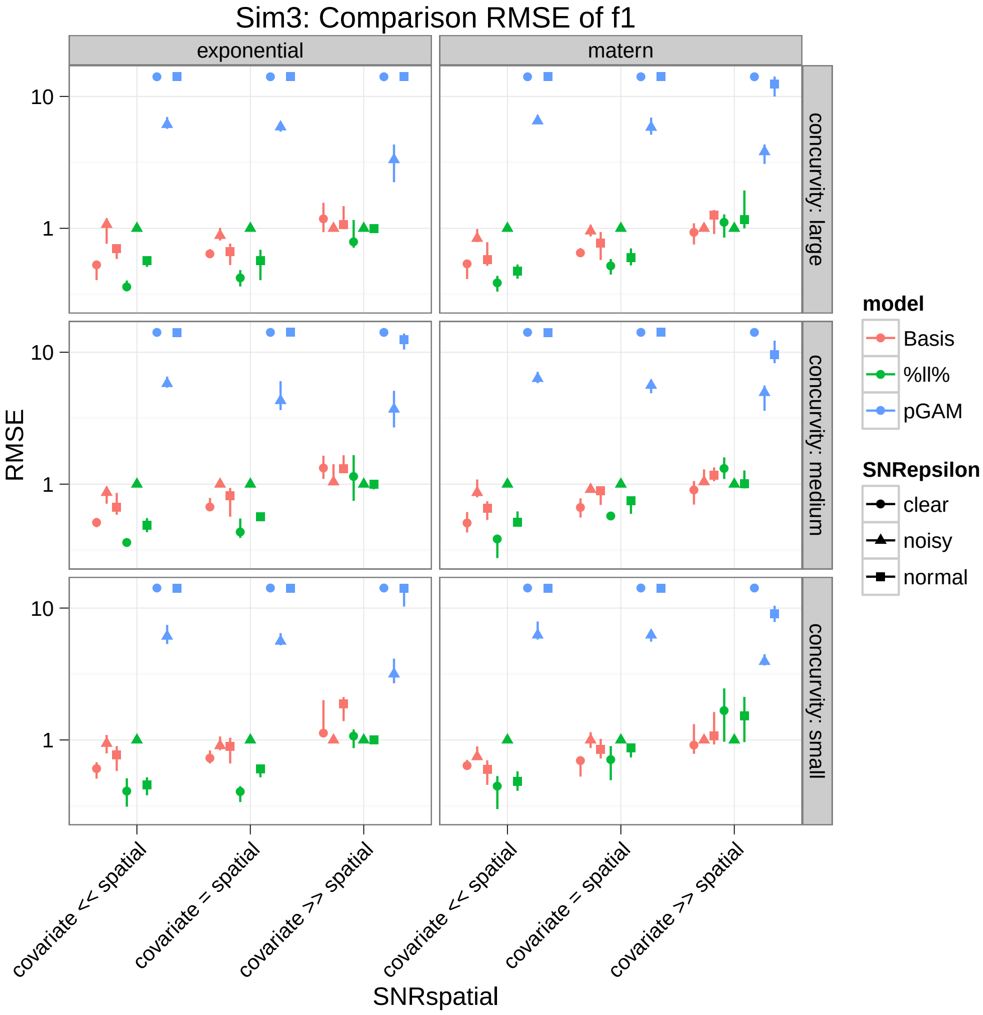

Figure 4.22.: Simulation 3: Comparison RMSE (median & inter-quartile range) of $f_1(x)$; "Basis"-Model (red), %ll%-Model (green) and pGAM-Model (blue).

%ll%-model provides the slightly better adaptation independent from the covariance type and the strength of concurvity. Only in the setting "covariate >> spatial" (SNRs = 0.1) in the Matèrn case, the basis-model has advantages compared to the %ll%-model.

The fit of the second covariate is presented in figure 4.23. The strength of

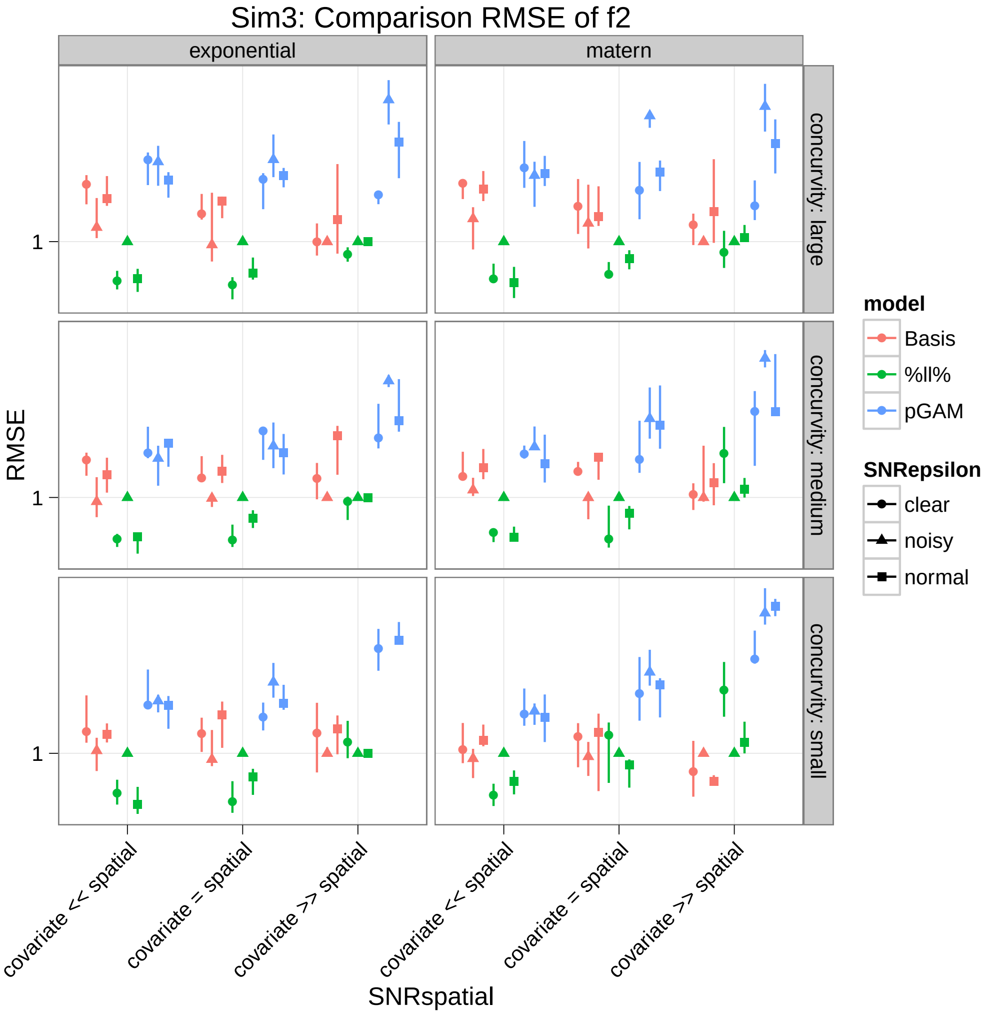

Figure 4.23.: Simulation 3: Comparison RMSE (median & inter-quartile range) of $f_2(x)$; "Basis"-Model (red), %ll%-Model (green) and pGAM-Model (blue).

the concurvity and the covariance function does not affect the fit at all. The %ll%-model performs always better than the comparable models. Again, the pGAM model provides the worst adaptation. Thus, the Matèrn case presents even greater challenge for the model.

The fit of covariate three is dominated by the %ll%-model. Figure 4.24 shows that in all settings the %ll%-model is the best, no matter of the type of covariance or the strength of concurvity. Thereby, the advantage is clearer in the

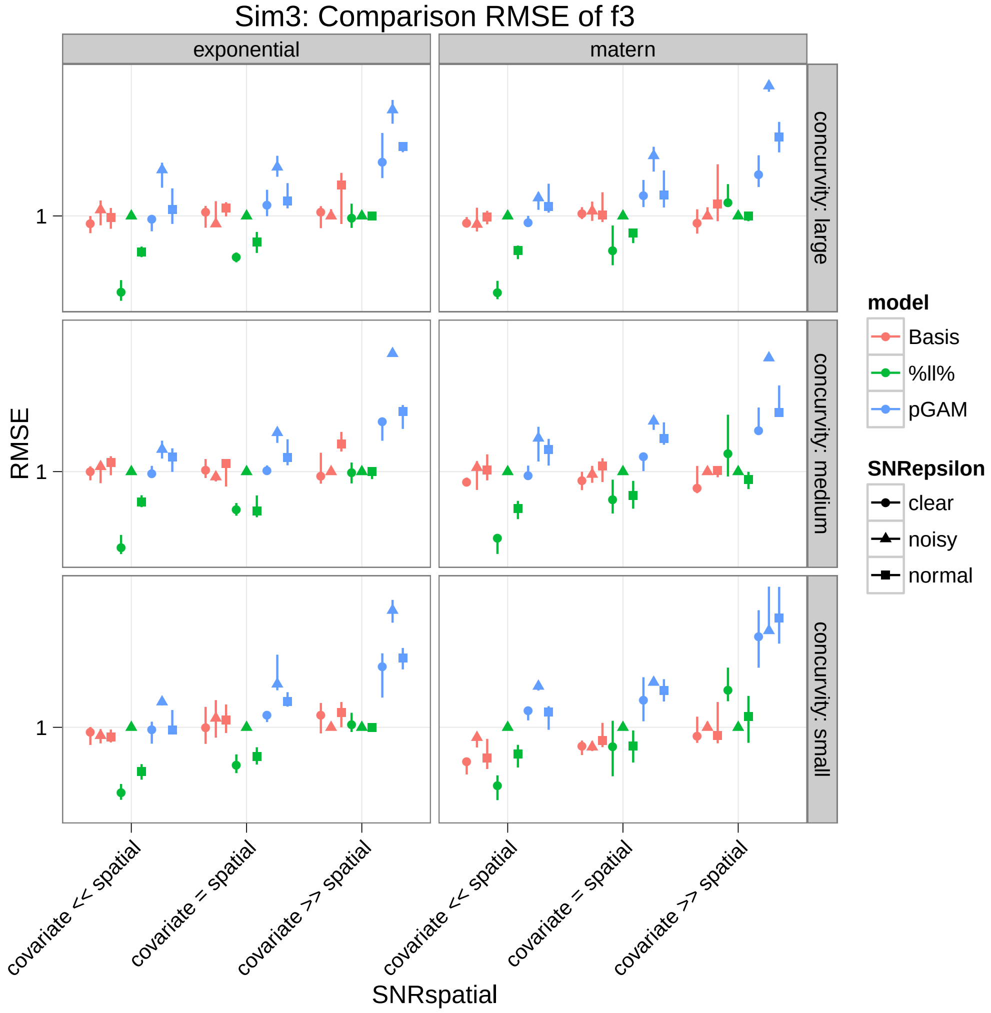

Figure 4.24.: Simulation 3: Comparison RMSE (median & inter-quartile range) of $f_3(x)$; "Basis"-Model (red), %ll%-Model (green) and pGAM-Model (blue).

exponential covariance case than in the Matèrn covariance case. Again, the pGAM-model particularly discloses its weakness in the setting "covariate >> spatial" (SNRs $= 0.1$).

Figure 4.25 focuses on the fit of the fourth covariate. In the exponential covariance case tendencially the %ll%-model is always better than the basis-model. In contrary, in the Matèrn covariance case the %ll%-model and the basis-model have pretty similar results. Again, the pGAM-model performs very badly. In most cases the pGAM-model does not recognize the covariate as to be important and therefore does not choose that one.

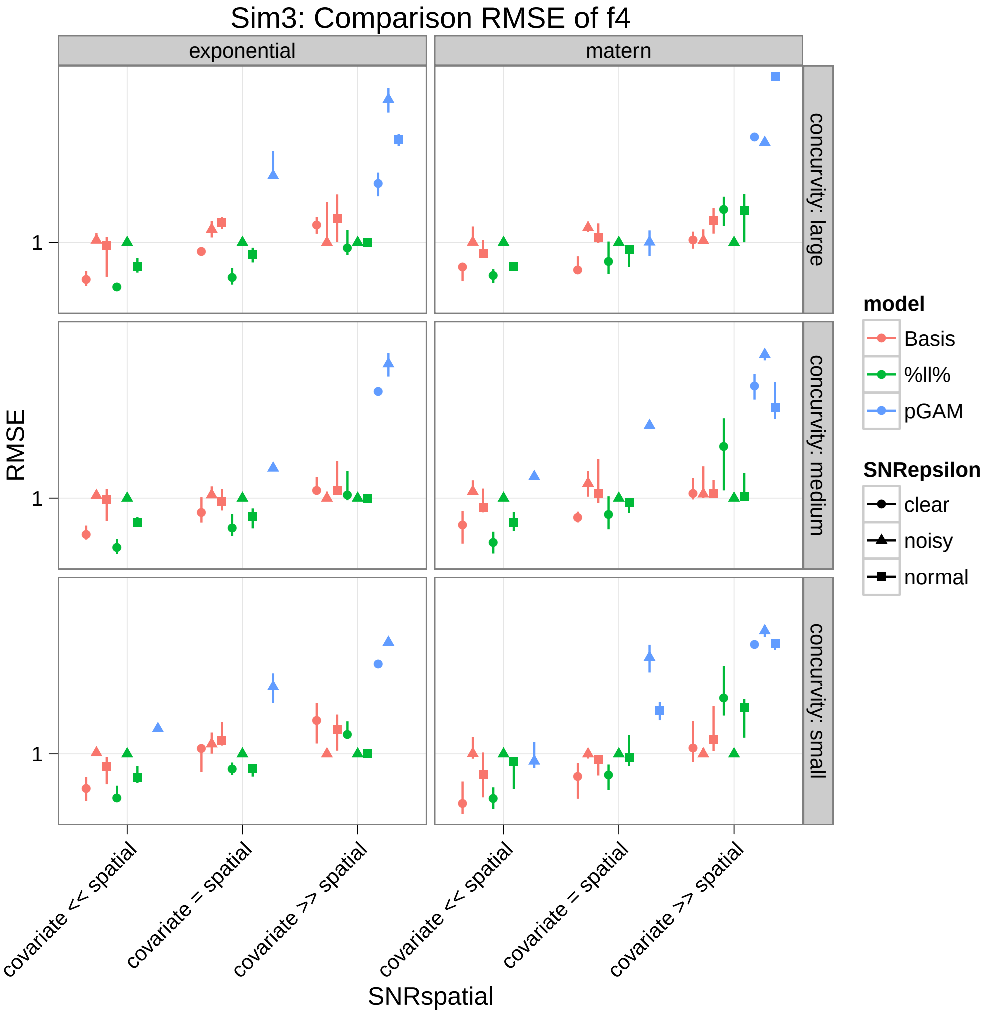

Figure 4.25.: Simulation 3: Comparison RMSE (median & inter-quartile range) of $f_4(x)$; "Basis"-Model (red), %ll%-Model (green) and pGAM-Model (blue).

The construction of the fifth covariate is based on the simple linear parametric function $f(x) = x$. Figure 4.26 presents great differences in the fit of the covariate between the models. The pGAM model clearly works the worst. Thus, the

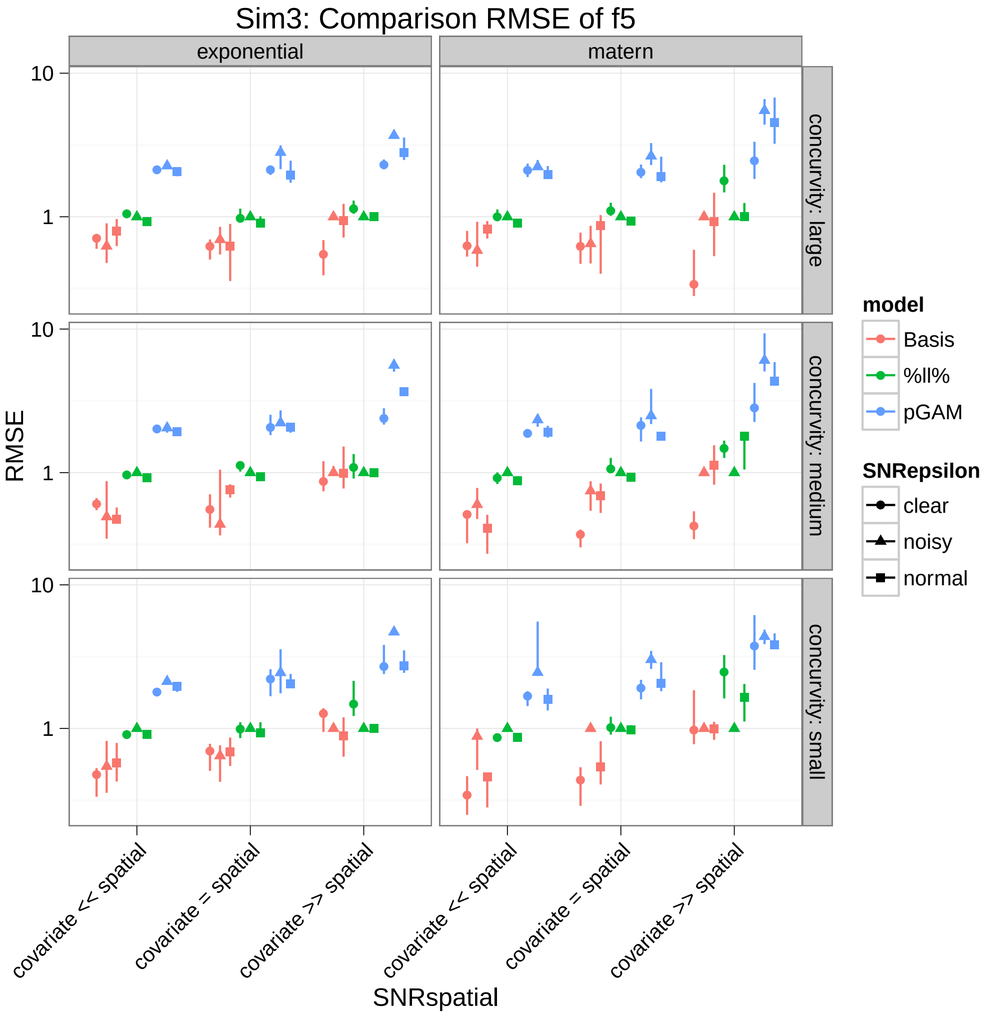

Figure 4.26.: Simulation 3: Comparison RMSE (median & inter-quartile range) of $f_5(x)$; “Basis”-Model (red), %ll%-Model (green) and pGAM-Model (blue).

Matèrn covariance produces a greater range than the exponential covariance. Although, the selection frequencies of the %ll%-model consider this covariate of great importance, the fit is clearly worse than in the basis-model in every setting, no matter which covariance function. It is possible that the chosen base-learner `bbs(·)` is too complex for the simple function. The basis-model is not affected that strong by that. There are further simulations necessary to answer this question correctly.

By considering at figure 4.27 and its fit of the spatial effect, one setting strikes out. Setting "covariate >> spatial" (SNRs = 0.1) puts a strong weight on the covariate effect. Thus, the spatial effect is designed to capture only the otherwise unexplainable variance. In this setting, the %ll%-model performs clearly

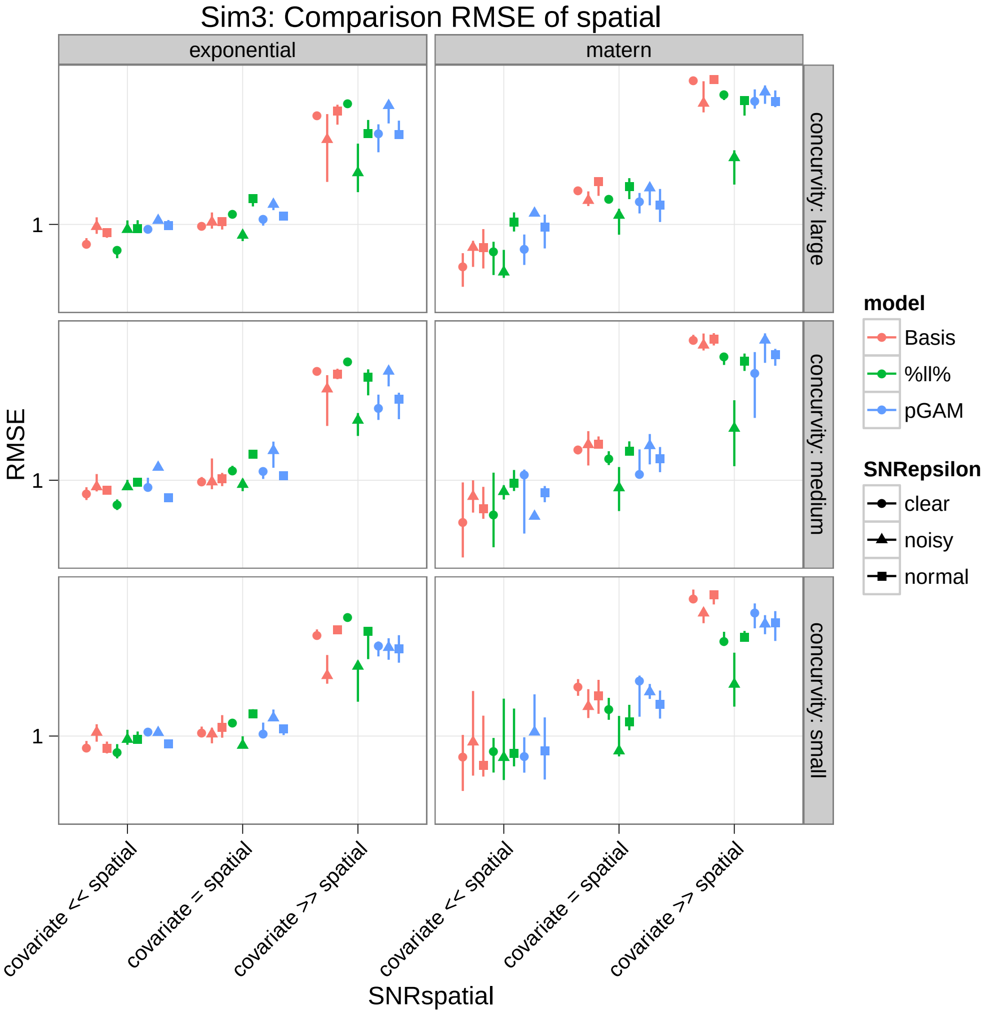

Figure 4.27.: Simulation 3: Comparison RMSE (median & inter-quartile range) of spatial effect; "Basis"-Model (red), %ll%-Model (green) and pGAM-Model (blue).

better than the two other models, no matter of the strength of concurvity or the covariance type. For the other settings in exponential covariance case, there are no clear differences between the models. Compared to that, the Matèrn covariance function causes greater difficulties. This can be seen at the larger range of RMSE.

4.3.5. Summary

This section summarize briefly the important simulation results. The presented simulations indicate that if more than one covariate and the spatial effect modified by the %ll%-operator is included to a possible model, then the fit of a single covariate is not affected by a certain covariance type.

Additionally, the selections frequencies in models with the %ll%-operator are totally independent from the covariance type and the strength of concurvity at all.

The operator has also strengths when a large part of the explainable variances corresponds to the covariate effect. The simulations show that a model with the %ll%-operator is clearly superior compared to models without this modification of the spatial effect.

A possible weakness of the methodology could also be revealed. Simulation 3 chooses a too complicated base-learner. The basis-model is still able to produce a good fit. In contrary, the %ll%-model is not able to recognize this and perform worse than the basis-model.

5. Real data example

There is hardly any other industry except for the forestry which is so dependent on the natural environment. Successful management of the forest is linked inextricably to the adaptation to natural climatic conditions. The income of forestry depends particularly on water safeness and the temperature changes whom they are confronted. Climate and soil are the main factors of production and determine the possibilities and limits of the forestry land use [Kölling, 2008].

However, decisive parameters for the management of forest areas change with an onset of climate change. In moderate climate change and therefore climate stabilization at a new level, adaptation measures promise success. In contrary, this is less probable with rapid and sustained changes [Umweltbundesamt, 2006]. It is necessary to minimize economic damages by adapting to the climate change already today. This requires to detect regional impacts of the climate change early. Thus, damage can be prevented or at least limited through active adaptation [Umweltbundesamt, 2006].

The detachment of the increasingly poorly matched and unstable Spruce Forests is possible via the ecological forest conversion within a reasonable period. The success of this forest conversion activities depends crucially on knowing the location of the affected species in future climate [Umweltbundesamt, 2006]. What are the future prospects of the main trees species Beech, Spruce and Pine? Particularly, the Beech (*Fagus sylvatica*) was grown increasingly in the last 20 years. If this trend will continue, it depends on the potential for adaptation of the Beech. This is controversially discussed for some time now [Sutmöller et al., 2008].

One of the most important tree species is the Spruce (*Picea abies*) in Germany. Regarding the climate change, the Spruce is considered to be a difficult tree [Kölling et al., 2007]. Originally, the distribution area of the Spruce was mainly placed in Central- and Eastern-Europe and Scandinavia. Meanwhile, the Spruce is also found in the lower areas of Central Europe [Schmidt-Vogt, 1989]. The Spruce reacts particularly sensitive to temperature and also has a low potential for adaptation [Roloff and Grundmann, 2008].

Therefore, the main forestry task is the choice of appropriate tree species [Sutmöller et al., 2008]. An adaptation to the consequences of climate change is inevitable. Since the forests of tomorrow must be planted today, this future question is a today's issue. The following analysis examines whether the Spruce is able to win this challenge.

5.1. Data collection

The core of the models builds the National Forest Inventory [Schmitz et al., 2004] for Bavaria. The data is supplemented with actual values to precipitation and temperature from the global climate database "World Clim" [Hijmans et al., 2005].
The National Forest Inventory 2001/2002 [Schmitz et al., 2004] was a nationwide terrestrial, carried out on a random basis with permanent sample points. It was acquired in all states and properties consistently. The following analysis are limited to Bavaria. The sample (cluster) distribution is based on a nationwide 4 km × 4 km quadrangle grid, determined by the Gauß-Krüger coordinates system. Partially the quadrangle grid is regionally intensified to get more accurate or regionally differentiated information. Each cluster covers a quadrangle with sides of 150 m. The cluster coordinates give the location of the south-west corner of the cluster. It was noted which tree species occurs at each cluster corner. Accordingly, for each tree species there is a binary response variable.

Additionally, the *total precipitation consists of the months May till September* and the *mean temperatures of the months June, July + August* as a cli-

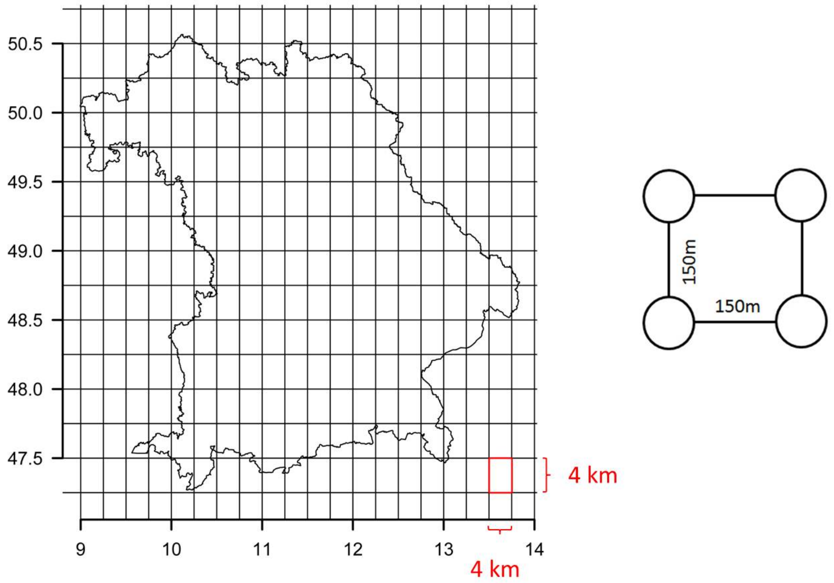

Figure 5.1.: Sketch of the sample design of the National Forest Inventory 2001: Partition of Bavaria in a uniform grid (left) and an inventory tract on a special grid point (right).

matic specific variable at each measuring point. “WorldClim”, a global climate database, provides long-term average of the years 1960 till 1990 with a spatial resolution of about one square kilometer [Hijmans et al., 2005].

The predictions are based on the results of the regional climatic model “WETTREG” of the firm Climate & Environment Consulting Potsdam GmbH, instructed by the Umweltbundesamt [Spekat et al., 2007]. The analyses are based on the moderate climate scenario B1.

Table 5.1 summarizes the used variables. The binary response variables of the eight different tree species have the value 1 in case of presence and in case of absence the value 0. Only the variables *mean temperatures of the months June, July + August* (T_678WC) and *total precipitation of the month May till September* (P_5to9WC) are used. These variables are described in detail in section 5.2.1. In addition, prediction variables of the moderate climate scenario B1 are provided in table 5.2.

type	variable name	explanation
response variable	Apseu	Sycamore (1 = presence, 0 = absence)
	Fsylv	Beech (1 = presence, 0 = absence)
	Fexce	Ash-Tree (1 = presence, 0 = absence)
	Qrobu	English Oak (1 = presence, 0 = absence)
	Saria	Haw (1 = presence, 0 = absence)
	Aalba	Fir (1 = presence, 0 = absence)
	Pabie	Spruce (1 = presence, 0 = absence)
	Psylv	Scotch Pine (1 = presence, 0 = absence)
predictor variables	LONG	Longitude
	LAT	Latitude
	T_678WC	Mean Temperature of months June, July + August [°C]
	P_5to9WC	Total Precipitation of months May till September [mm]

Table 5.1.: variables description

type	variable name	explanation
prediction variables	T_678_21	WETTREG B1 scenario 2071-2100 Mean Temperature of months June, July + August [°C]
	T_678_21	WETTREG B1 scenario 2071-2100 Total Mean Precipitation of months May till September [mm]

Table 5.2.: additional variables Bavaria

5.2. Analysis

Exemplified by the Spruce (*Picea abies*) the competing models are presented and their results are discussed in this section. All data analysis in this section have been carried out using the R system for statistical computing [R Development Core Team, 2012], version 2.14.2.

5.2.1. Descriptive Analysis

The available data cover the entire surface of Bavaria. Figure 5.2 illustrates that there are (comparatively small) holes in the theoretical sampling grid. A Spruce at about 82% of the 5992 measuring points could be found, especially at the Alps and the Czech border. The Spruce population density decreases rather at Lower Franconia.

The left part of figure 5.3 shows that the total mean precipitation of the months May – September range between 288 mm and 400 mm. The total mean pre-

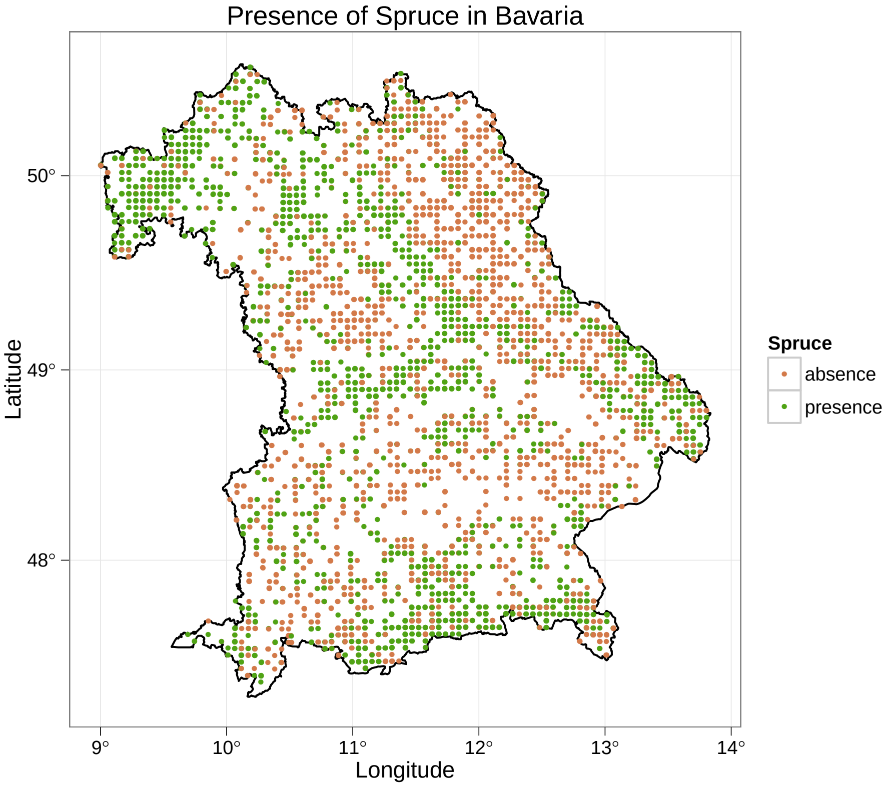

Figure 5.2.: Measuring points of Spruce at Bavaria, colored by the value of the response variable Spruce.

cipitation is the highest with values about over 600 mm at the Alps and the Prealps. Going northwest means less precipitation. It is noticeable that when precipitation is over 400 mm, there is mostly a Spruce observed. The right part of figure 5.3 shows that the median of precipitation is 350 mm in areas without Spruce. In contrast, the median is 420 mm in areas with Spruce.

The total mean precipitation does not change on average barely (20 mm) in the summer months of the years 2071 till 2100. Figure 5.4 displays that there are regionally differences, especially the margin of deviation is higher. As a peak 1000 mm precipitation is expected at the Alps. Contrary, the expected precipitation goes back to values of 230 mm at the lowlands. In the Alps and Prealps precipitation increases, the precipitation decreases in the low mountains and

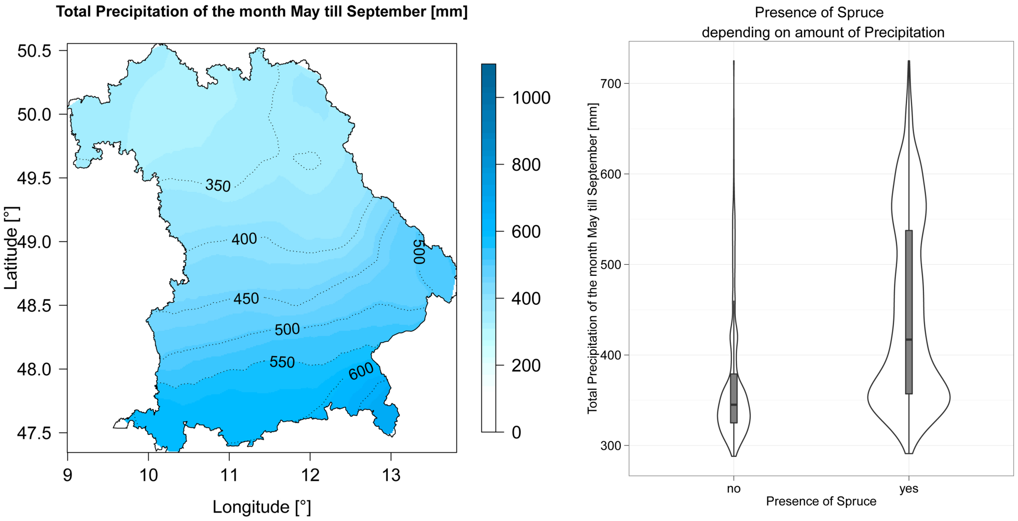

Figure 5.3.: Precipitation: interpolated values depend on the location (left) and distribution of the repsonse variable (right).

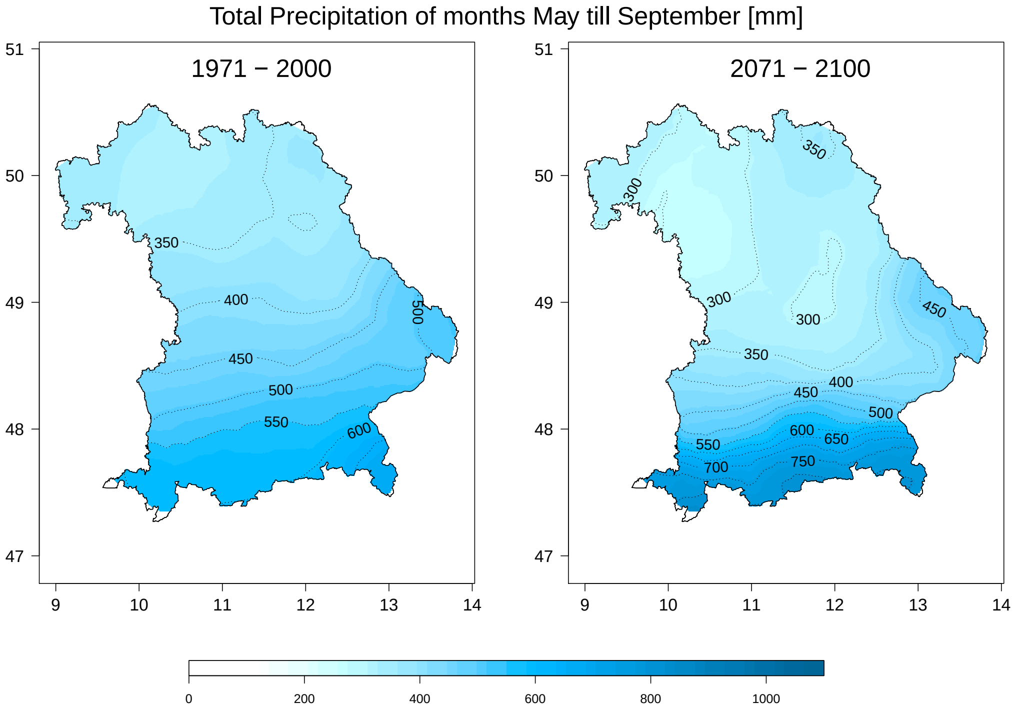

Figure 5.4.: Precipitation: actual values (left) and climate scenario WETTREG B1 (right).

along the Czech border. The same can be observed in large parts of Frankonia.

The Spruce prefers cool temperatures besides humid climate. Figure 5.5 shows that descriptive because a Spruce is found about 15 degrees at nearly every measuring point. The temperature values have a range between 13 degrees at the Alpine valleys and 18 degrees at the Main. The temperature increases tendencially from south to north. At higher altitudes (Bavarian Forest, Fichtelgebirge) it is much cooler. Most of the observations are between 15 degrees and 16.5 degrees. It is common knowledge that the temperature will increase in the coming years. The moderate scenario assumes an average temperature rise of about 1.4 degrees. Both in the so far coolest regions and warmest it is getting warmer. The range remains at about 2.5 degrees.

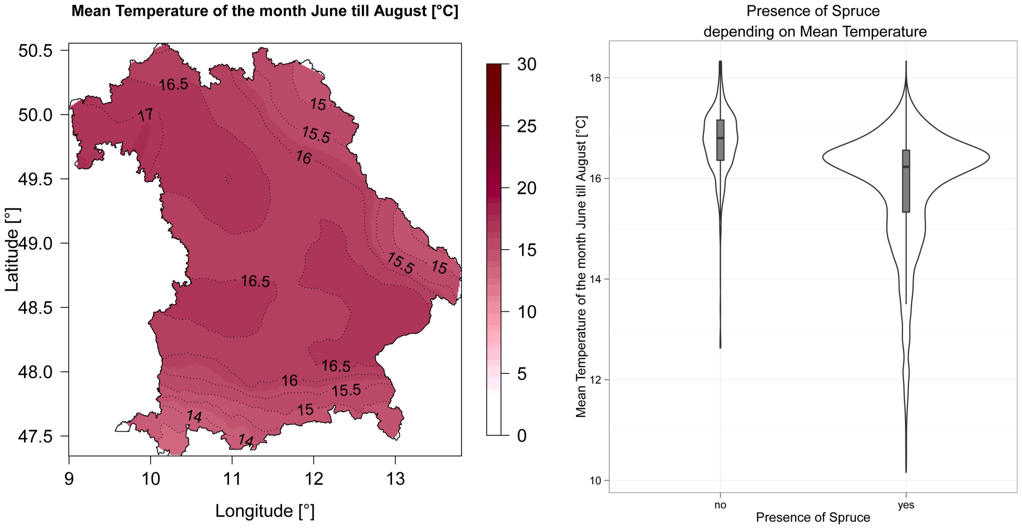

Figure 5.5.: Average temperature: interpolated values depend on the location (left) and distribution of the response variable (right).

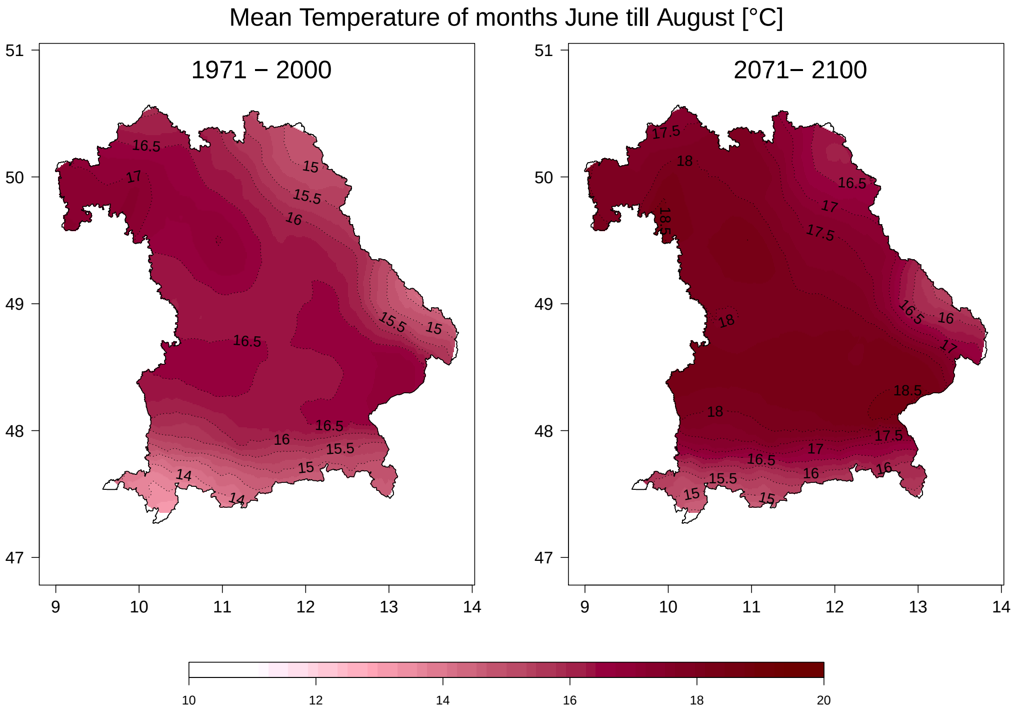

Figure 5.6.: Average temperature: actual values (left) and climate scenario WETTREG B1 (right).

5.2.2. Model

Based on the results of the descriptive analysis (section 5.2.1), the aim is to estimate a model for presence of a Spruce in Bavaria.

As predictor variables, precipitation and temperature are included in the model. The effect of both metrical variables, *mean temperature of the months June, July + August* and *total precipitation of the months May – September*, is modeled in a non-parametrical fashion with the help of P(enalized)-Splines (section 2.3.2).
A spatial component is modeled in order to take account of the spatial correlation of the data. The surface of *longitude* and *latitude* is estimated with the help of Bivariate P(enalized)-Splines (section 2.3.2).

The estimation of the model

- Response variable: $y_i \in \{1, 0\}$ $i = 1, ..., 5992$

$$y_i | \eta_i \sim B\left(1, \pi_i\right)$$

- Expected Value:

$$\mathbb{E}\left(y_i\right) = \mathbb{P}\left(y_i = 1\right) = \pi_i = logit^{-1}\left(\eta_i\right)$$

- Predictor:

$$\eta_i = \beta_0 + f_1\left(\text{temperature}_i\right) + f_2\left(\text{precipitation}_i\right) + g\left(\text{Long}_i, \text{Lat}_i\right)$$

As in the simulation study (chapter 4), the same three models shall be used again. The three competing models are presented in detail in the following. The model comparison of the two mboost-models is of primary interest, especially, how the %ll%-operator affects the response curves and the spatial effect. To compare the models better, only the “default”-settings are used.

- **Model 1: "Basis" mboost-model**

```
gamboost(Pabie ~ bbs(T_678WC) + bbs(P_5to9WC) +
    bspatial(LONG, LAT), family = Binomial(),
    data = bay,
    control = boost_control(mstop = 200, trace = TRUE, nu = 0.2))
```

Both covariates, temperature and precipitation are modeled non-parametrically with the help of the bbs-base-learner. The "default"-settings for this base-learner are used. A spatial effect is modeled to absorb the spatial autocorrelation of the data. Additionally, the spatial effect also serves as a surrogate for all other unobserved. The spatial effect is modeled with the help of the bspatial-base-learner. The "default" settings for this base-learner are also used. The response variable "Pabie" is binary. This is captured by the family "Binomial". The initial m_{stop}-parameter is increased to 200. The optimal m_{stop} iteration is calculated with the help of a 25-k-fold bootstrap. The step-length-parameter ν is also increased to 0.2.

- **Model 2: mboost-model with the %ll%-operator**

```
gamboost(Pabie ~ bbs(T_678WC) + bbs(P_5to9WC) +
    bspatial(LONG, LAT) %ll% [bbs(T_678WC) %+% bbs(P_5to9WC)],
    family = Binomial(), data = bay,
    control = boost_control(mstop = 200, trace = TRUE, nu = 0.2))
```

With the help of the bbs-base-learner, the two covariates are also modeled as in the basis-model non-parametrically. The "default" settings are used again and a spatial effect is modeled to cover the spatial autocorrelation of the data as well. The spatial effect is modified with the %ll%-operator. In this model, the spatial effect only covers the otherwise unexplained variance. As in the basis-model, the binary response is modeled with the family "Binomial" again. Furthermore, the initial m_{stop}-parameter is also increased to 200. The optimal m_{stop} iteration is determined with the help of a 25-k-fold bootstrap. The step-length-parameter ν is also increased to 0.2.

- **Model 3: pGAM-model**

  ```
  pGAM(y, X, thresh = 0.95, family = binomial(link = "logit"))
  ```

  The pGAM method is used as in the simulation study. Thus, a third comparable model is calculated. The covariates temperature and precipitation and a spatial effect are included again in the model and modeled non-parametrically. The “default” settings are used. The binary response is considered with the family “`Binomial`”.

Linkage to simulation studies

The aim is to link the simulation studies with the application in order to assess the results in a better way. It is interesting to note, if the application is equal to certain setting where the newly developed %ll%-operator possibly works greatly. Thus, there is the purpose to estimate the signal-to-noise with the help of the application data:

- SNRconcurvity:
  $$\widehat{SNRc} = \frac{\text{sd}[g_j(\mathbf{coords})]}{\text{sd}(\boldsymbol{\varepsilon}_j)}$$
  This signal-to-noise cannot be estimated from the data. It is impossible to determine $\text{sd}[g_j(\mathbf{coords})]$ and the corresponding $\text{sd}(\boldsymbol{\varepsilon}_j)$.

- SNRspatial:
  $$\widehat{SNRs} = \frac{\text{sd}[g(\mathbf{Long}, \mathbf{Lat})]}{\text{sd}[f_1(\mathbf{temperature}) + f_2(\mathbf{precipitation})]}$$
  The estimation of the three models of this signal-to-noise ratio differ slightly:
  - $\widehat{SNRs}_{\text{Basis}}$: 0.68

  - $\widehat{SNRs}_{\%ll\%}$: 0.47
  - $\widehat{SNRs}_{\text{pGAM}}$: 0.19

  Thus, all three models estimate a relatively small `SNRs`. Compared to the simulation studies, all the estimated `SNRs` can be located between the setting "covariate >> spatial" (`SNRs` $= 0.1$) and "covariate = spatial" (`SNRs` $= 1$).

- SNRepsilon:

  $$\widehat{SNRe} = \frac{\text{sd}[g(\mathbf{Long}, \mathbf{Lat}) + f_1(\mathbf{temperature}) + f_2(\mathbf{precipitation})]}{\text{sd}(\boldsymbol{\varepsilon})}$$

  This signal-to-noise cannot be estimated from the data. The problem is that the `mboost`-package has no implementation to extract the residuals ($\boldsymbol{\varepsilon}$) of a logit-model and therefore to calculate $sd(\boldsymbol{\varepsilon})$.

Thus, only `SNRs` remains to classify the results. The simulation studies show that in the setting when the explainable variance corresponds to covariate effects rather than to the spatial effect, the %ll%-model performs clearly better than the other two models.
Therefore, it is expected that the %ll%-model will provide the best result. The response curves will stabilized and the spatial effect will only reproduce the variability which cannot be explained by two covariates. Therefore, the spatial effect allows a small-scale resolution.

5.3. Results

The following section presents the results of the model introduced in section 5.2.2. Figure 5.7 and figure 5.9 represent the estimated effects of the climate variables of the three different models. The presences and absences are plotted above and below the response curves. The following interpretations of the marginal effects are only valid for constant other covariates. At first glance, a

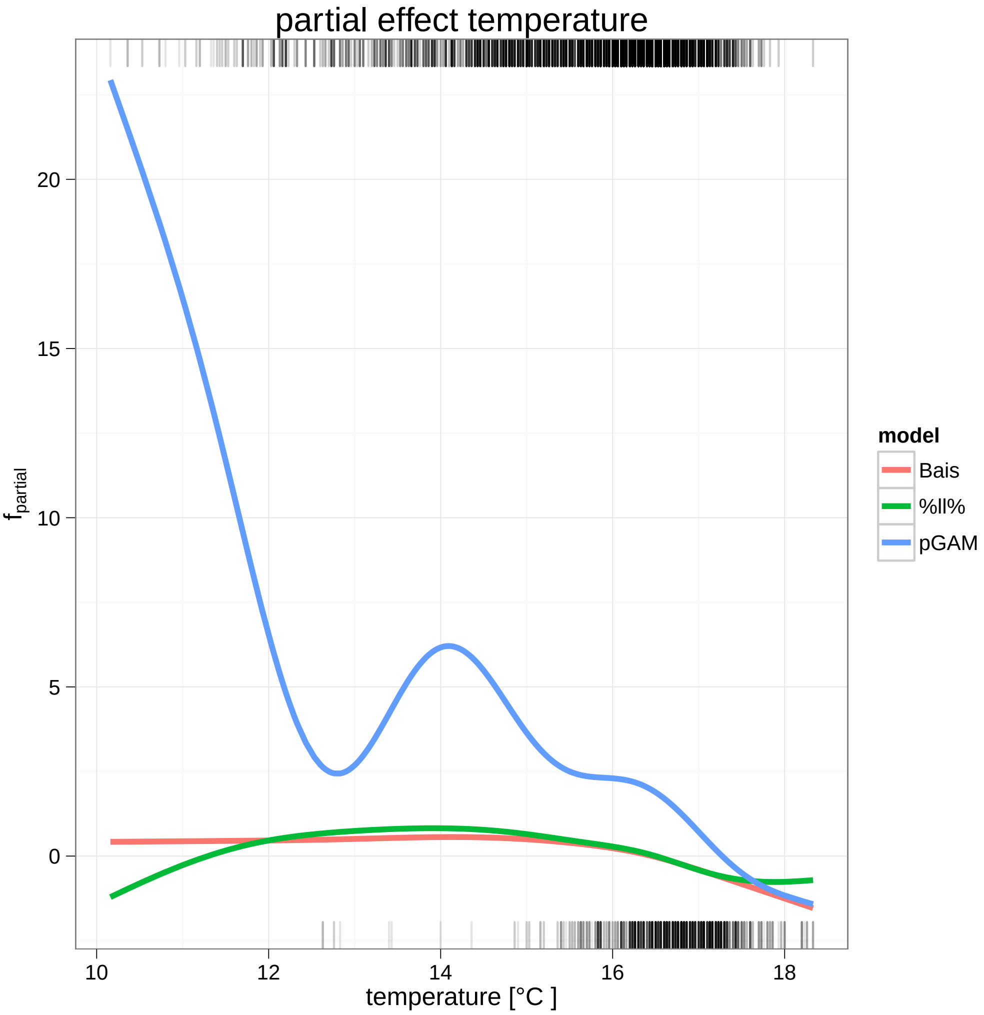

Figure 5.7.: Comparison Temperature Response Curve: Basis-Model (red), %ll%-Model (green) and pGAM-Model (blue).

very huge effect of the pGAM model can be noticed. The chance for the presence of a Spruce falls very strong linear down to a temperature of 13 degrees in the pGAM model. This chance has another small peak at a temperature of 14 degrees. After that, it falls sharply at higher temperatures.

In this representation, it is difficult to say something about the boosting effects. For this reason, figure 5.8 focuses on the effect of the boosting-models. In this case, the basis model provides the chance for the presence of a Spruce pretty constant up to about 15 degrees. From just 15 degrees, then the chance falls quite clearly. The estimated effect by the %ll%-model looks a bit different.

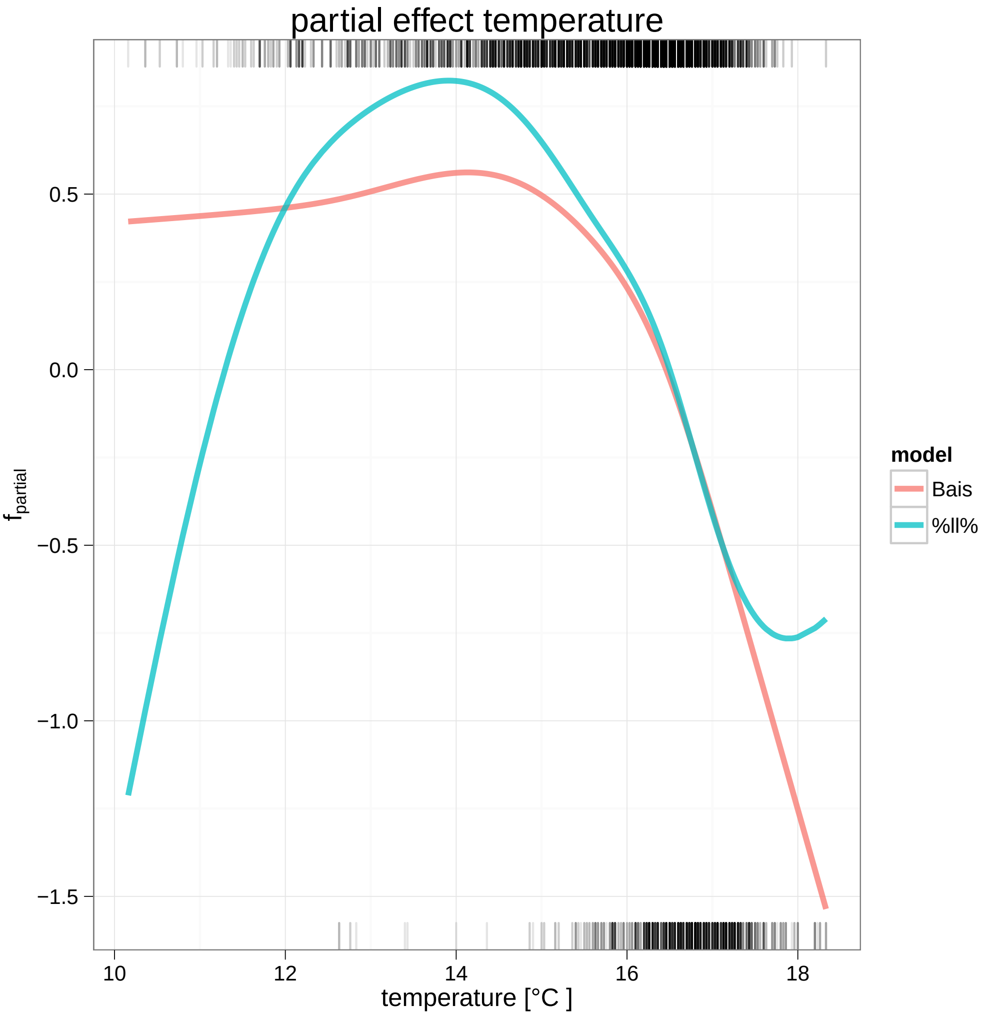

Figure 5.8.: Comparison Temperature Response Curve of the Boosting Models: Basis-Model (red) and %ll%-Model (blue).

The chance for the presence of a Spruce raises linearly with increasing temperature. The chance reaches its peak at a temperature of about 15 degrees. Thereafter, the chance falls nearly linear with increasing temperature.

Figure 5.9 shows the estimated precipitation effect by the three models. A strange impact of the pGAM model effect can be registered. The pGAM model is not able to estimate a continuous smooth effect. From the estimated effect a trend to a rising chance for the presence of a Spruce at higher precipitation can be only observed. Now, figure 5.10 focuses on the boosting effects

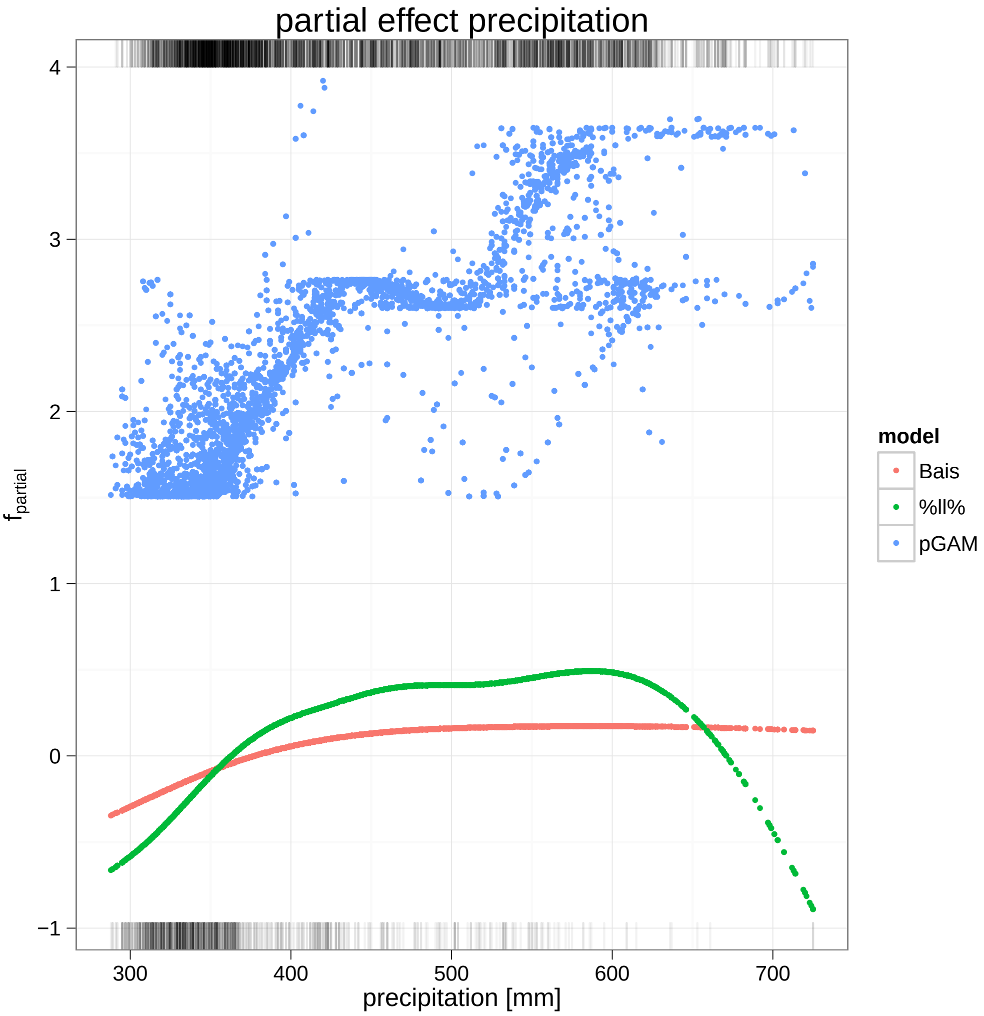

Figure 5.9.: Comparison Precipitation Response Curve: Basis-Model (red), %ll%-Model (green) and pGAM-Model (blue).

to compare these much better. The basis model provides the chance for the presence of a Spruce, raising linearly with increasing precipitation. At a rainfall of 500 mm the chance remains rather constant. The estimated effect by

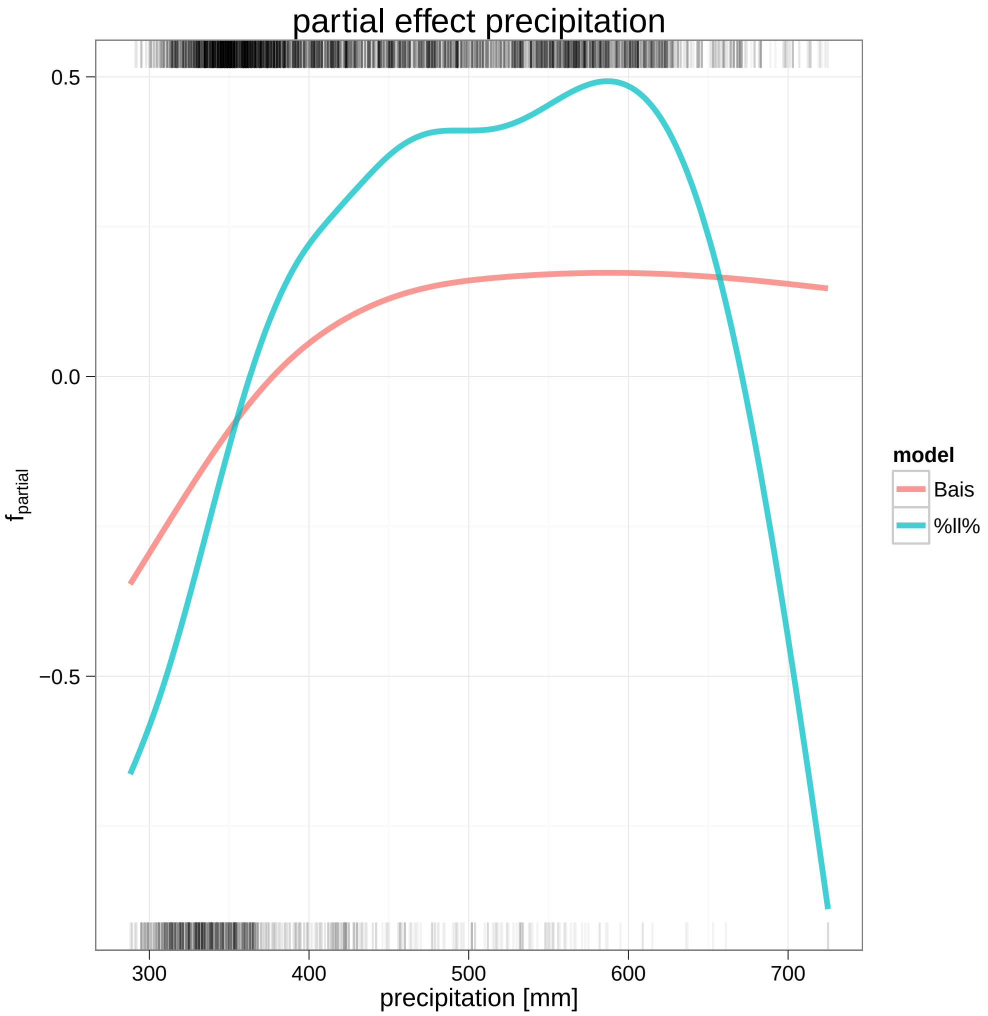

Figure 5.10.: Comparison Precipitation Response Curve of the Boosting Models: Basis-Model (red) and %ll%-Model (blue).

the %ll%-model looks a bit different again. The chance for the presence of a Spruce raises noticeably with increasing precipitation. The chance reaches its peak at a precipitation of about 600 mm degrees. Thereafter, the chance falls nearly linear with increasing precipitation.

In the following, the spatial effect of the three different models are considered in more detail. Figure 5.11 shows the spatial effect of the basis boosting model.

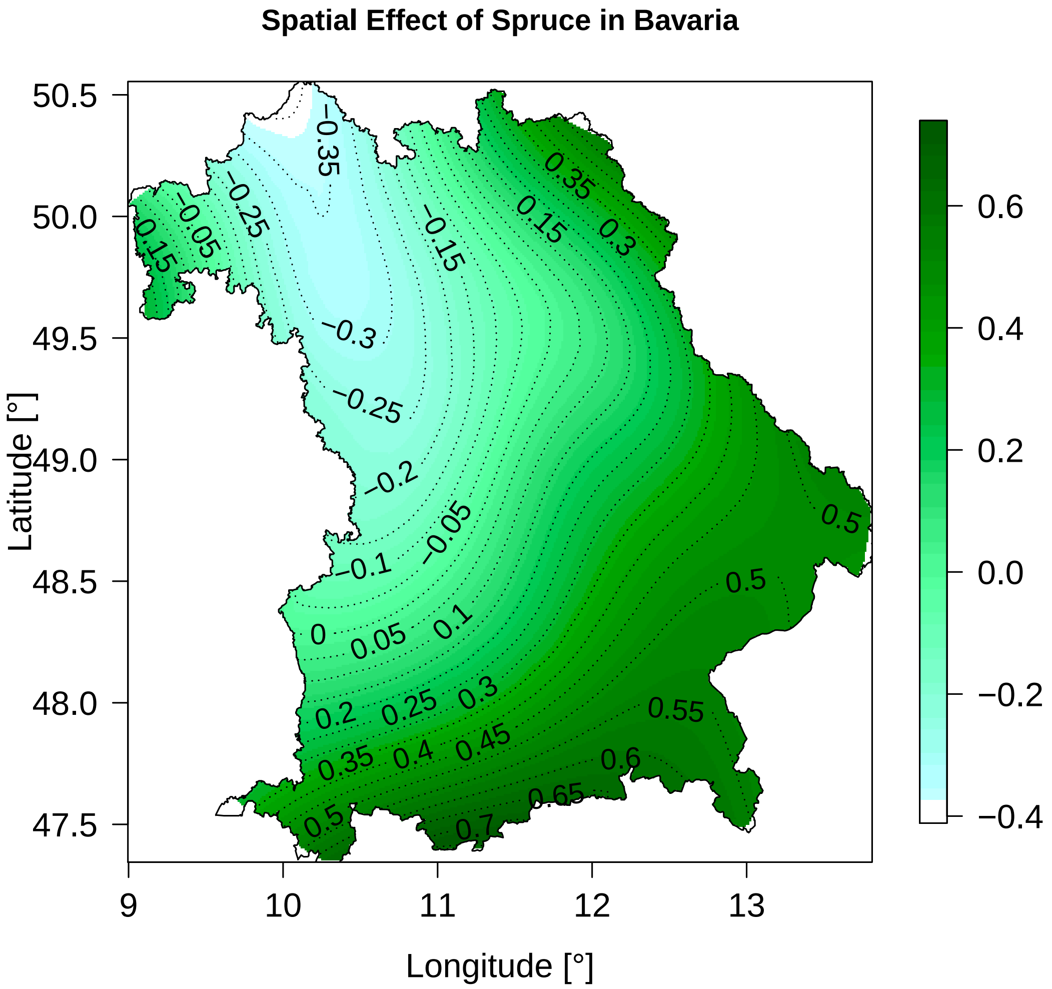

Figure 5.11.: Spatial Effect of the Basis-Boosting Model.

The spatial effect lies in the range from -0.4 to 0.6. The spatial effect determines that the greatest chance for the presence of a Spruce is in the Alps, the Bavarian Forest and the Upper Palatinate. Figure 5.12 displays the estimated spatial effect of the %ll%model. The spatial effect of the %ll%-model looks slightly different again and lies in the range between -1 and 1. The greatest chance for the presence of a Spruce is in the Bavarian Forest, Upper Franconia and the Upper Palatinate. Figure 5.13 presents the estimated spatial effect of the pGAM-model. The spatial effect of the pGAM-model has the greatest range and differs clearly from the other two models. The range lies between

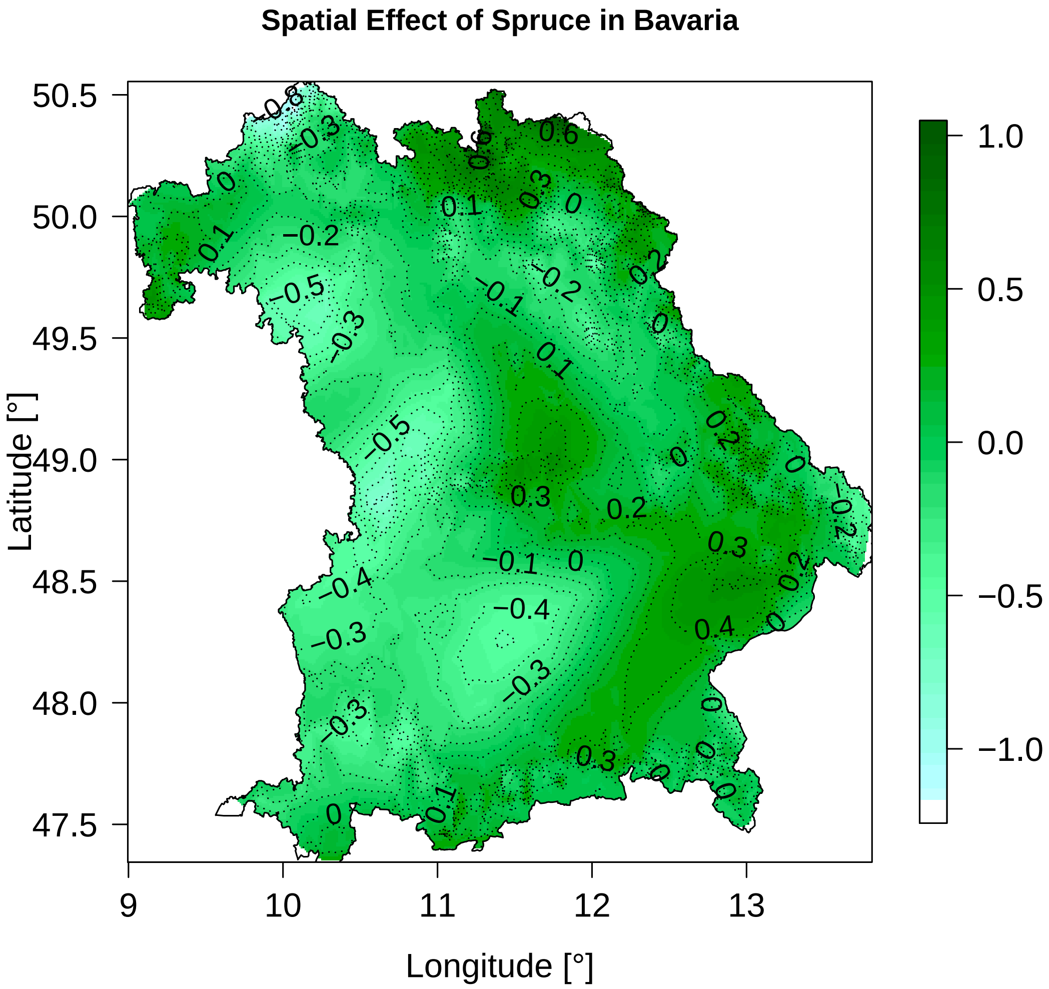

Figure 5.12.: Spatial Effect of the%ll%-Model.

1.5 and 4. The greatest chance for the presence of a Spruce is at the edge of the Bavarian Forest.

The estimated effects of pGAM models differ greatly from the others. Actually, these effects do not correspond to the expectations of ecologists [Ewald, 2009] at all. Furthermore, the measure of the goodness of fit with the help of the AIC [Akaike, 1974] advises clearly against the further use of the pGAM-model. Therefore, with a value of 4295 the AIC of the pGAM-model is the worst. In contrary, the %ll%-model has the lowest AIC with value of 4055. The AIC of

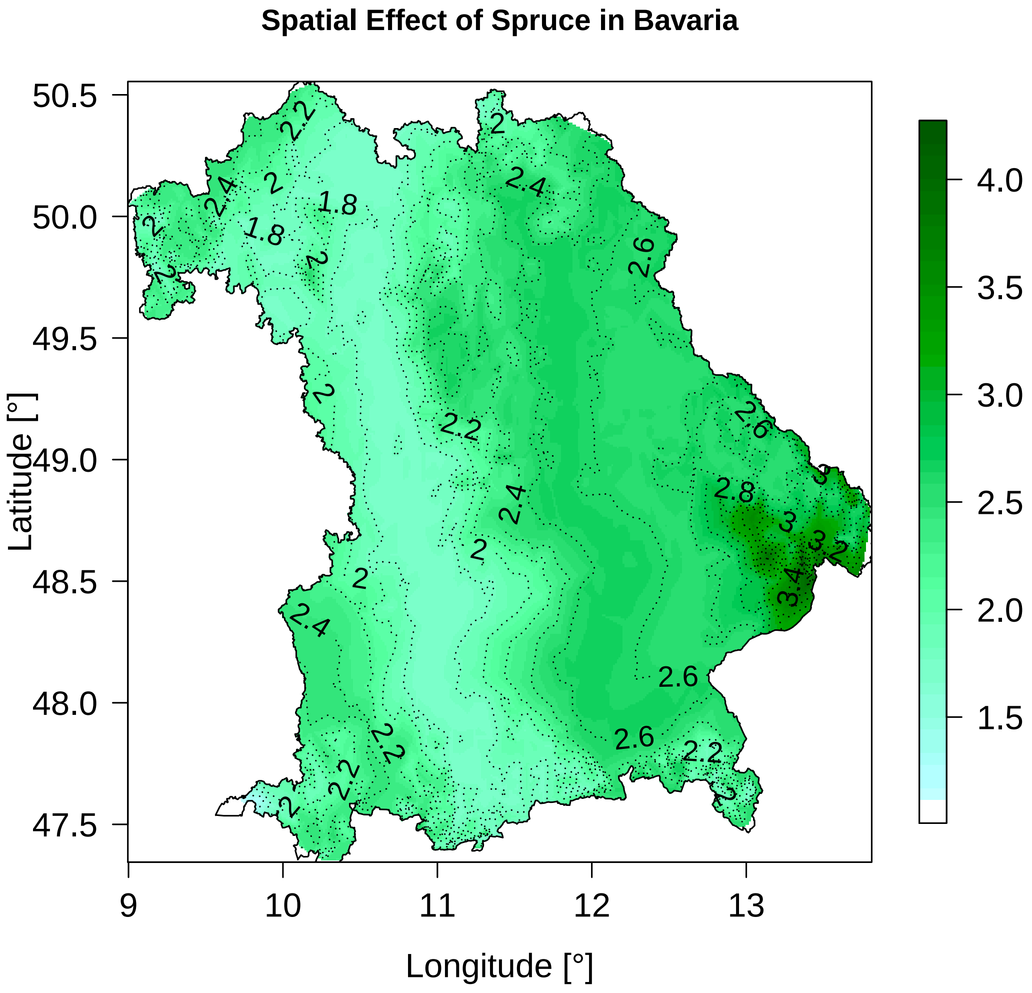

Figure 5.13.: Spatial Effect of the pGAM-Model.

the basis-model is 4221 and so between them. Hence, the boosting models are used only in the further analysis.

Lindenlaub and Wickler [2012] made a similar analysis with the help of Bayesian methods. The same model for Bavaria is used. In addition, Lindenlaub and Wickler [2012] use prior information from an Europe-model to stabilize the estimated effects of the Bavaria-model. The estimated effects look similar to the effects from the %ll%-model. Thus, the %ll%-operator and the prior information have seemingly the same effect.

The estimated probability for the growth of a Spruce is shown in figure 5.14. In addition, the observations are located as a plausibility check. The left part of figure 5.14 shows the estimated probability for the growth of a Spruce for the basis model. The highest probability (with values about 90%) can be found in the Bavarian Forest and Upper Palatinate. The lowest probability is observed in Lower- and Middle Franconia. The right part of the figure presents the estimated probability for the %ll%-model. Like the basis-model, the %ll%-model registered the same highest and lowest probability areas. The estimated probabilities by the two boosting models differ only slightly in parts of Upper Bavaria and Swabia. There, the %ll%-model provides a low chance.

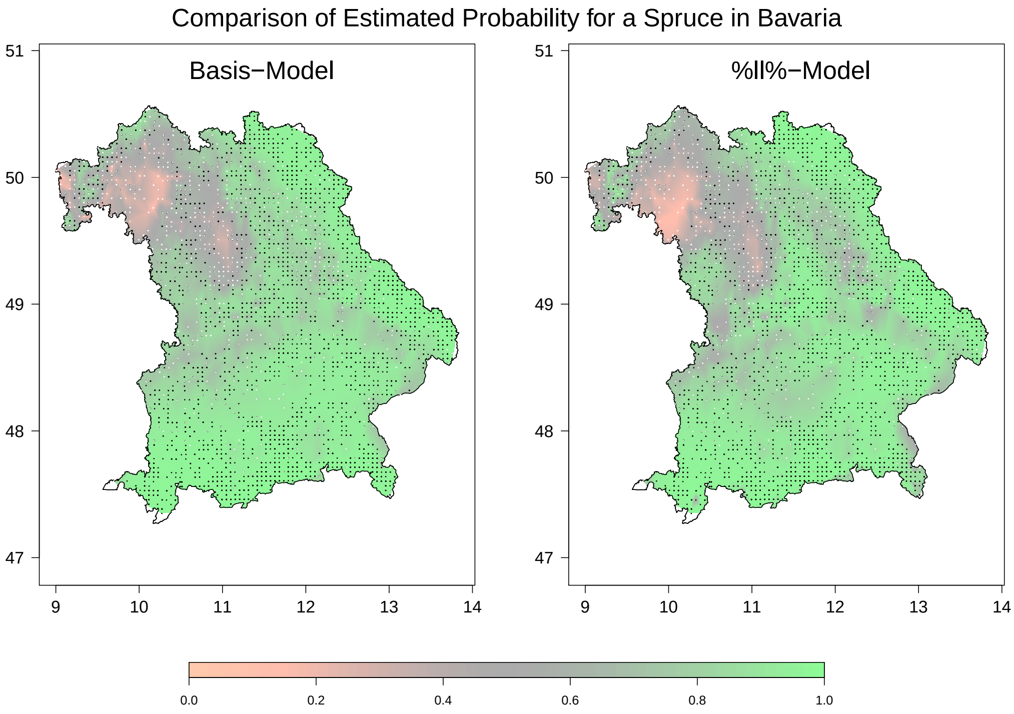

Figure 5.14.: Comparison of Estimated Probability of the Boosting-Models: Basis-Model (left) and %ll%-Model (right).

5.4. Climate Scenario 2071 - 2100

During the past 100 years, the average annual temperature increased by about 0.8 degree in Germany. All previous years of the 21st Century were warmer than the long-term average temperature of 8.3 degree [Umweltbundesamt, 2006]. How the climate will develop in the future there are only forecasts. The tree growth should be projected into the future with the help of the comparatively "mild" scenario B1. This scenario assumes an increase in temperature of "only" 1.8 degree and a decline in the total precipitation of 20 mm in average in Bavaria. More details are shown in section 5.2.1. The climate change confronts the forestry with a great challenge. The models shall serve as a support for decision of climate-friendly forestry. Especially adapted to the cold climate, there are visible consequences for the Spruce. Figure 5.15 compares the estimated probability of the basis model for the presence of a Spruce for the present and the future. The left part of the figure provides the

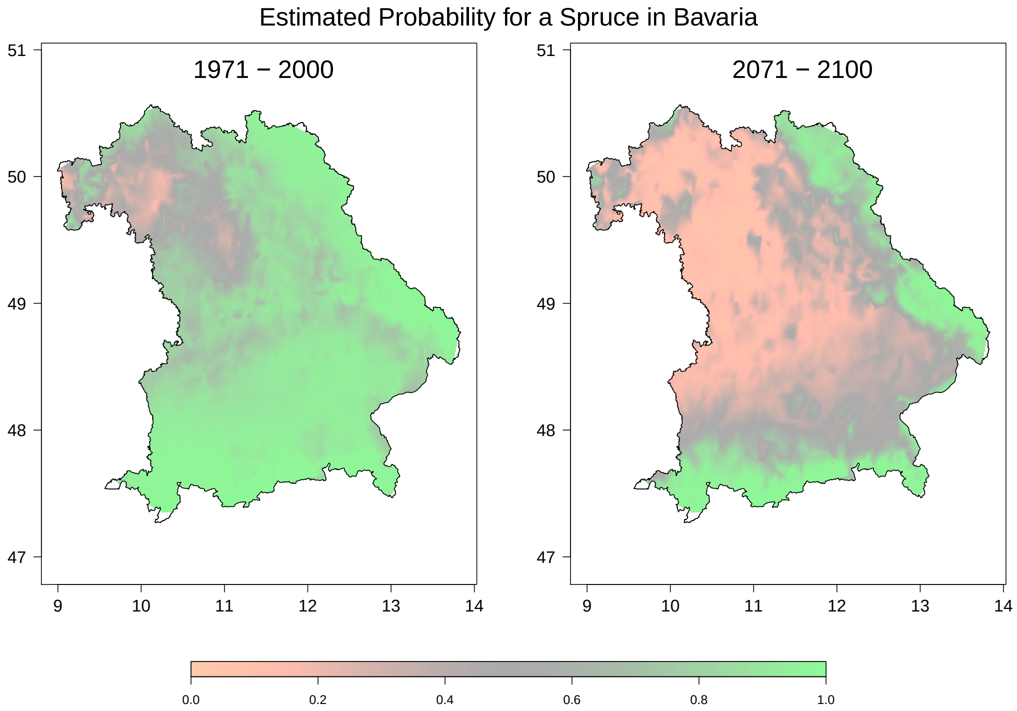

Figure 5.15.: Comparison of Estimated Probability and Forecast of the Basis-Model.

estimation as shown above. The right part shows the forecast in the future. At first glance, great changes can be seen. With the exception of parts of Lower Franconia, the Spruce is currently available in the entire state of Bavaria with an occurrence probability of more than 50%. In the north of the Danube the Spruce is expected only in the high altitudes along the Czech border and the Alps in the years 2071-2100. The probability decreases to less than 20% nationwide in Lower and Middle Franconia. Merely, Spruce growth can still be expected in Rhön and Spessart.

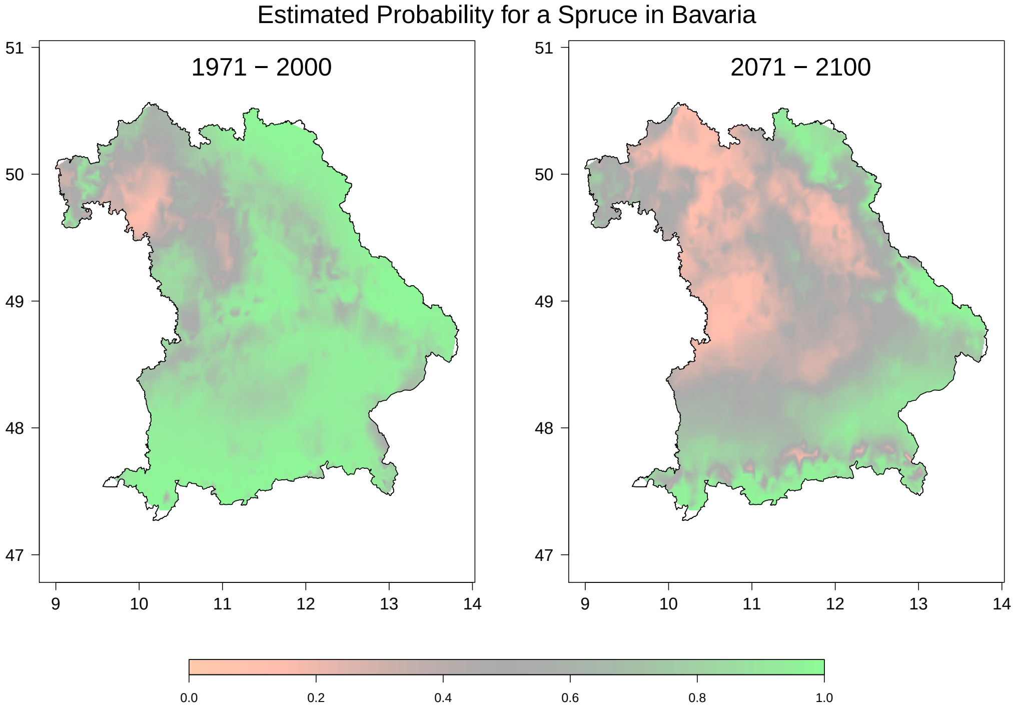

Figure 5.16.: Comparison of Estimated Probability and Forecast %ll%-Model.

Figure 5.16 also compares the estimated probability of the %ll%-model for the present and the future. The left part of the figure represents the estimation as shown above again. The right part presents the forecast for the years 2071-2100. At first glance, great changes can be seen again. The %ll%-model also predicts a strong decrease of the Spruce inventory. The greatest chance for the presence of a Spruce is in the Alps, Prealps and the Bavarian Forest in the future. The presence of a Spruce with a probability of about 50% can be expected in Swabia, Oberbayern and parts of Lower Franconia. Figure 5.17

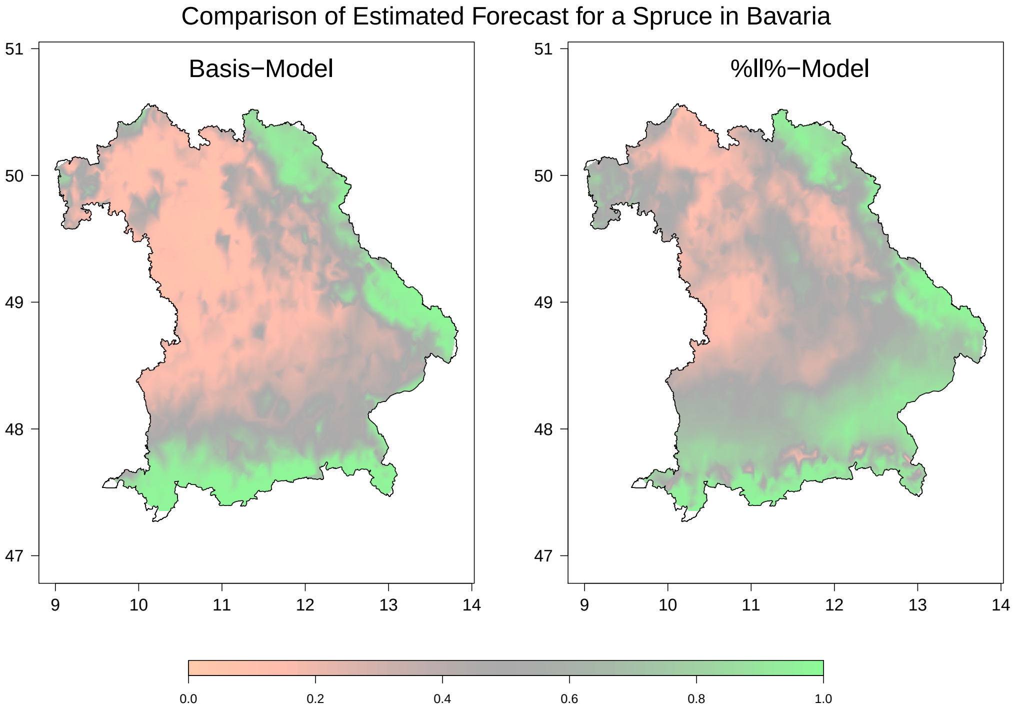

Figure 5.17.: Comparison of Forecast of the Boosting-Models: Basis-Model (left) and %ll%-Model (right).

compares the forecast of the basis-model and the %ll%-model. Here, slight differences can be seen between the two forecasts. The forecasts differ mainly in the foothills of the Alps and Lower Franconia.

6. Summary and perspectives

If covariate and spatial effects are modeled at the same time in order to cover spatial autocorrelation and unobserved heterogeneity, it will lead to wrong or attenuated effects in the presence of "concurvity". However, the %ll%-operator succeeds to correct these estimates by making the basis functions, used for the spatial effect, orthogonal to the basis functions of the covariate effect. The simulations show that the overall model fit does change hardly. Admittedly, the %ll%-operator changes the fit of each included covariate of a model. Therefore, the %ll%-operator actually manages separation of the spatial autocorrelation between spatial and covariate effect.

Thus, the result of the simulations indicate that if more than one covariate and the spatial effect modified by the %ll%-operator is included to a possible model, then the fit of a single covariate is not affected by a certain covariance type. Additionally, in models with the %ll%-operator the selections frequencies are totally independent from the covariance type and the strength of concurvity at all. The operator has also obvious strengths when a large part of the explainable variances corresponds to the covariate effect. In this case, a model with the %ll%-operator is clearly superior compared to models without this modification of the spatial effect.

A possible weakness of the methodology could also be revealed by the simulations. One simulation chooses a too complicated base-learner. The basis-model is still able to produce a good fit. In contrary, the %ll%-model is not able to recognize this. However, the %ll%-model still provides a suitable fit independent of the strength of concurvity. This point definitely should be studied with further researches.

The simulations have also managed to test the limits of the procedure. None of the presented methods are able to provide good estimations if the settings are extreme. If the strength of concurvity (SNRc) increases to 0.1, the results of the three models are relatively similarly bad. No model can capture the combination of strong noise (SNRe $= 0.1$) and strong concurvity (SNRc $= 0.1$) well. Especially, the pGAM-model is not able to get a result at all.

Generally, the modeling of a spatial effect to cover spatial autocorrelation is highly discussed by users [Franklin, 2009].
Returning to the quote from Niels Bohr, statistical models should enable prediction. The main criticism of prediction concerns the spatial effect. The spatial effect is predicted to stay totally unchanged, while for the other covariates, changes are usually assumed. However, it can be assumed for sure that not only the covariates will change in the future. There is much more to be assumed that the spatial effect, as a surrogate for all unobserved will change, too. For this reason, Franklin [2009] proposes to model only a spatial effect, if there is an explicit focus on the appearance of the response curves. In contrary, the focus is primary on the prediction that the spatial effect should rather not be included in the model [Franklin, 2009]. Unfortunately, this problem cannot be solved directly with the help of the %ll%-operator.

A further generalization of the idea, making one variable orthogonal to another variable, leads to the general case of two random correlated variables $\boldsymbol{A}$ and $\boldsymbol{B}$. In the models, presented in this thesis, it is obvious that the basis functions, used for the spatial effect needs to be orthogonal to the basis functions to the covariate effect. Admittedly, in the case with two variables $\boldsymbol{A}$ and $\boldsymbol{B}$ it is not clear at all if $\boldsymbol{A}$ should be orthogonal to $\boldsymbol{B}$ or $\boldsymbol{B}$ should be orthogonal to $\boldsymbol{A}$. More than that, it makes a great difference for the result. This question requires further research.

The previous chapter 5 shows that the onset of climate change has large impacts on tree population and their future distribution areas, especially for the Spruce which has really adapted to cold climate. Based on the climate scenario “WETTREG B1” [Spekat et al., 2007] the models predict visible consequences. According to the forecasts the distribution of spruce will shrink sharply in many parts of Bavaria (figure 5.17). If the climate scenario “WETTREG B1” will be fulfilled as expected, in the north of the Danube the Spruce will only occur in higher altitudes along the Czech border and in the Rhön and Spessart. The Spruce is almost no longer found in Lower and Middle Franconia. Kölling et al. [2007] have also made researches and come to similar results.

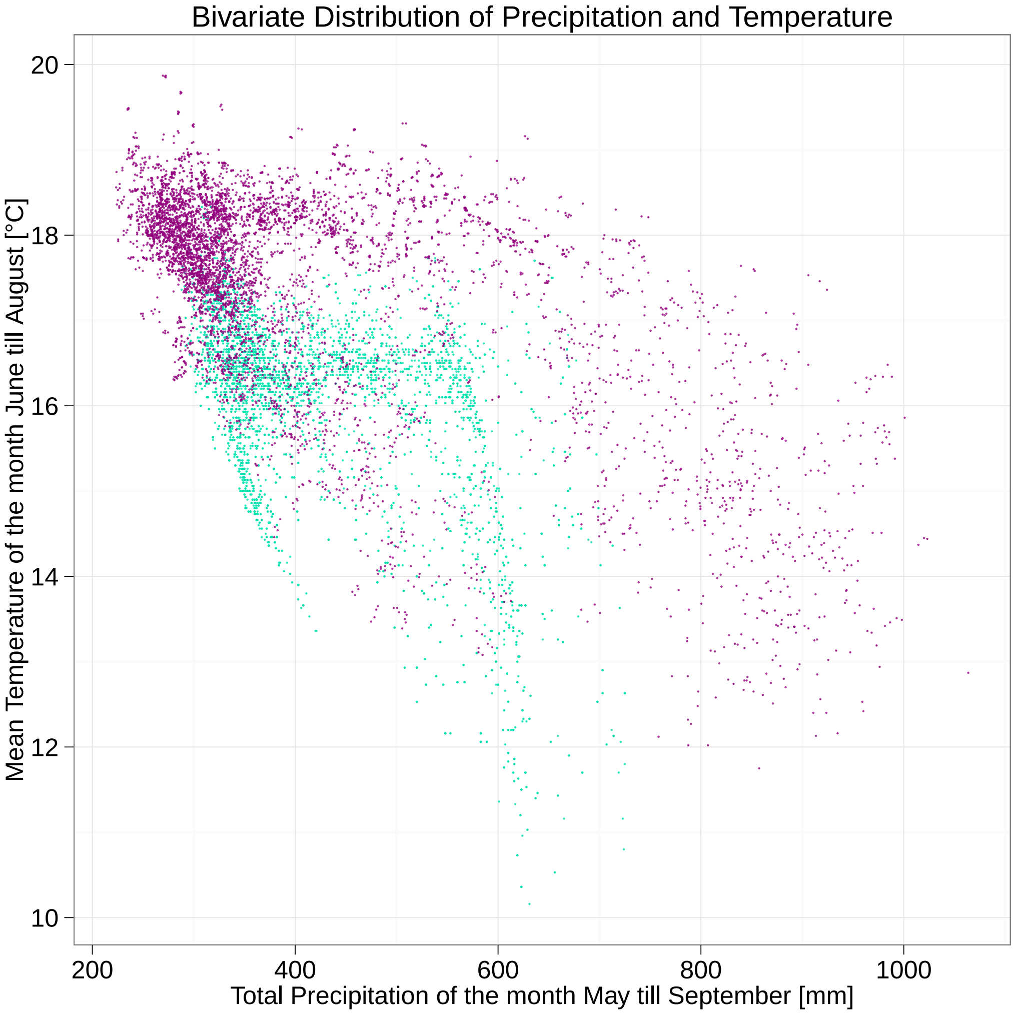

Figure 6.1.: Bivariate Distribution of Precipitation (pink) and Temperature (violet) in Bavaria.

Although the predictions of the two boosting models do not differ clearly, the response curves differ mainly at the margins. Particularly the basis-model has great difficulties to model the margins correctly. This is the result of the combination of the presence of concurvity and very few data on the edges as shown by figure 6.1. The figure 6.1 presents the bivariate Distribution of Precipitation (pink) and temperature (violet) in Bavaria. Lindenlaub and Wickler [2012] could stabilize the estimation of the response curves and especially the margins by using Bayesian Analysis. Lindenlaub and Wickler [2012] developed a model with prior information from an European model with the aim to stabilize the margins. However, the Bayesian Analysis is very computationally intensive and costly and requires much programming. In this situation, the %ll%-operator is a very good alternative. The %ll%-operator needs little processing time and is the adequate tool for this situation to get useful results.

7. List of Figures

8. Bibliography

Akaike, H. 1974. A New Look at the Statistical Model Identification. Automatic Control, IEEE Transactions on 19:716–723.

Breiman, L. 1998. Arcing Classifier (with discussion and a rejoinder by the author). The annals of statistics 26:801–849.

Breiman, L. 1999. Prediction Games and Arcing Algorithms. Neural computation 11:1493–1517.

Bühlmann, P., and T. Hothorn. 2007. Boosting Algorithms: Regularization, Prediction and Model Fitting. Statistical Science 22:477–505.

Bühlmann, P., and B. Yu. 2003. Boosting With the L 2 Loss. Journal of the American Statistical Association 98:324–339.

Buja, A., T. Hastie, and R. Tibshirani. 1989. Linear Smoothers and Additive Models. The Annals of Statistics pages 453–510.

Cover, T., and J. Thomas. 1991. Elements of Information Theory. Wiley-interscience.

Efron, B. 1979. Bootstrap Methods: Another Look at the Jackknife. The annals of Statistics 7:1–26.

Eilers, P., and B. Marx. 1996. Flexible Smoothing with B-Splines and Penalties. Statistical science pages 89–102.

Ewald, J. 2009. Experten-basierte Nischenmodelle – Wahrscheinlichkeit für Vorkommen von baumförmigen Exemplaren (relative Darstellung). Unpublished Communication .

Fahrmeir, L., T. Kneib, and S. Lang. 2009. Regression. Regression: Modelle, Methoden und Anwendungen, Statistik und ihre Anwendungen, Volume. ISBN 978-3-642-01836-7. Springer-Verlag Berlin Heidelberg, 2009 1.

Figueiras, A., J. Roca-Pardiñas, and C. Cadarso-Suárez. 2005. A Bootstrap Method to avoid the Effect of Concurvity in Generalised Additive Models in Time Series Studies of Air Pollution. Journal of Epidemiology and Community Health 59:881–884.

Fortin, M., and M. Dale. 2005. Spatial Analysis: A Guide for Ecologists. Cambridge University Press.

Franklin, J. 2009. Mapping Species Distributions (Ecology, Biodiversity and Conservation). Cambridge University Press.

Freund, Y., and R. Schapire, 1995. A Desicion-Theoretic Generalization of online Learning and an Application to Boosting. Pages 23–37 *in* Computational Learning Theory. Springer.

Freund, Y., R. Schapire, and N. Abe. 1999. A short Introduction to Boosting. Journal-Japanese Society For Artificial Intelligence 14:1612.

Friedman, J. 2001. Greedy Function Approximation: A Gradient Boosting Machine. Annals of Statistics pages 1189–1232.

Friedman, J., T. Hastie, and R. Tibshirani. 2000. Additive Logistic Regression: A Statistical View of Boosting (with discussion and a rejoinder by the authors). The annals of statistics 28:337–407.

Furrer, R., D. Nychka, and S. Sain. 2012. fields: Tools for Spatial Data URL `http://CRAN.R-project.org/package=fields`, R package version 6.6.3.

Genz, A., F. Bretz, T. Miwa, X. Mi, F. Leisch, F. Scheipl, and T. Hothorn. 2012. mvtnorm: Multivariate Normal and t Distributions URL `http://CRAN.R-project.org/package=mvtnorm`, R package version 0.9-9994.

Gu, H., T. Kenney, and M. Zhu. 2010. Partial Generalized Additive Models: An Information–Theoretic Approach for Dealing With Concurvity and Selecting Variables. Journal of Computational and Graphical Statistics 19:531–551.

Guisan, A., T. Edwards, and T. Hastie. 2002. Generalized Linear and Generalized Additive Models in Studies of Species Distributions: Setting the Scene. Ecological Modelling 157:89–100.

Hastie, T., and R. Tibshirani. 1990. Generalized Additive Models. Chapman & Hall/CRC.

He, S., 2004. Generalized Additive Models for Data with Concurvity: Statistical Issues and a novel Model Fitting Approach. Ph.D. thesis, University of Pittsburgh.

Hijmans, R. J., S. E. Cameron, J. L. Parra, P. G. Jones, and A. Jarvis. 2005. Very High Resolution Interpolated Climate Surfaces for Global Land Areas. International Journal of Climatology 25:1965–1978.

Hodges, J., and B. Reich. 2010. Adding Spatially-Correlated Errors can mess up the Fixed Effect you Love. The American Statistician 64:325–334.

Hofner, B., 2011. Boosting in Structured Additive Models. Ph.D. thesis, Ludwig-Maximilians-Universität München.

Hofner, B., T. Hothorn, T. Kneib, and M. Schmid. 2011. A Framework for Unbiased Model Selection Based on Boosting. Journal of Computational and Graphical Statistics 20:956–971.

Hothorn, T., P. Buehlmann, T. Kneib, M. Schmid, and B. Hofner. 2009. mboost: Model-Based Boosting URL `http://CRAN.R-project.org/package=mboost`, R package version 2.0.

Kearns, M., and L. Valiant. 1994. Cryptographic Limitations on learning Boolean Formulae and finite Automata. Journal of the ACM (JACM) 41:67–95.

Kölling, C. 2008. Wälder im Klimawandel: Die Forstwirtschaft muss sich anpassen. Gefahren für Menschen, Tiere und Pflanzen /Hrsg.: José L. Lozán pages 357 – 361.

Kölling, C., L. Zimmermann, and H. Walentowski. 2007. Klimawandel: Was geschieht mit Buche und Fichte? AFZ-Der Wald 62:584 – 588.

Legendre, P. 1993. Spatial Autocorrelation: Trouble or new Paradigm? Ecology 74:1659–1673.

Legendre, P., and M. Fortin. 1989. Spatial Pattern and Ecological Analysis. Plant Ecology 80:107–138.

Lennon, J. 2000. Red-shifts and Red Herrings in Geographical Ecology. Ecography 23:101–113.

Lindenlaub, C., and F. Wickler. 2012. Species Distribution Modelling von Baumarten mit Bayesianischen GAMSs. Unpublished Report .

Lumley, T., and L. Sheppard. 2003. Time Series Analyses of Air Pollution and Health: Straining at Gnats and Swallowing Camels? Epidemiology 14:13.

Maloney, K., M. Schmid, and D. Weller. 2011. Applying Additive Modelling and Gradient Boosting to assess the Effects of Watershed and reach Characteristics on Riverine Assemblages. Methods in Ecology and Evolution 3:116–128.

Miller, J., M. Turner, E. SmithWick, C. Dent, and E. Stanley. 2004. Spatial Extrapolation: The Science of Predicting Ecological Patterns and Processes. BioScience 54:310–320.

Peters, D., and J. Herrick. 2004. Strategies for Ecological Extrapolation. Oikos 106:627–636.

Petzoldt, T. 2012. akima: Interpolation of Irregularly Spaced Data URL `http://CRAN.R-project.org/package=akima`, R package version 0.5-7.

R Development Core Team, 2012. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.

Ramsay, T., R. Burnett, and D. Krewski. 2003*a*. Exploring Bias in a Generalized Additive Model for Spatial Air Pollution Data. Environmental Health Perspectives 111:1283.

Ramsay, T., R. Burnett, and D. Krewski. 2003*b*. The Effect of Concurvity in Generalized Additive Models linking Mortality to Ambient Particulate Matter. Epidemiology 14:18.

Reich, B., J. Hodges, and V. Zadnik. 2006. Effects of Residual Smoothing on the Posterior of the Fixed Effects in Disease-Mapping Models. Biometrics 62:1197–1206.

Roloff, A., and B. Grundmann. 2008. Klimawandel und Baumarten – Verwendung für Waldökosysteme. Stiftung Wald in Not .

Schapire, R. 1990. The Strength of Weak Learnability. Machine learning 5:197–227.

Schmidt-Vogt, H. 1989. Die Fichte, Band II/2. Paul Parey, Hamburg und Berlin.

Schmitz, F., H. Polley, P. Hennig, F. Schwitzgebel, and W.-U. Kriebitzsch. 2004. Die zweite Bundeswaldinventur–BWI2: Das Wichtigste in Kürze. Bundesministerium für Verbraucherschutz, Ernährung und Landwirtschaft (Hrsg.), Bonn .

Segurado, P., M. Araújo, and W. Kunin. 2006. Consequences of Spatial Autocorrelation for Niche-based Models. Journal of Applied Ecology 43:433–444.

Shannon, C., W. Weaver, R. Blahut, and B. Hajek. 1948. The Mathematical Theory of Communication. University of Illinois press Urbana.

Spekat, A., W. Enke, and F. Kreienkamp. 2007. Neuentwicklung von Regional Hoch aufgelösten Wetterlagen für Deutschland und Bereitstellung regionaler Klimaszenarios auf der Basis von globalen Klimasimulationen mit dem Regionalisierungsmodell WETTREG auf der Basis von gobalen Klimasimulationen mit ECHAM5/MPI-OM T63L31 2010 bis 2100 für die SRES-Szenarios B1, A1B und A2: Forschungsprojekt im Auftrag des Umweltbundesamtes, FuE-Vorhaben, Förderkennzeichen 20441138; Endbericht. Umweltbundesamt.

Stone, M. 1974. Cross–Validatory Choice and Assessment of Statistical Predictions. Journal of the Royal Statistical Society. Series B (Methodological) pages 111–147.

Sutmöller, J., H. Spellmann, C. Fiebiger, and M. Albert. 2008. Der Klimawandel und seine Auswirkungen auf die Buchenwälder in Deutschland. Ergebnisse angewandter Forschung zur Buche 3:135.

Tobler, W. 1970. A Computer Movie Simulating Urban Growth in the Detroit Region. Economic geography 46:234–240.

Tobler, W. 1979. Cellular Geography. Philosophy in geography 9:379–386.

Umweltbundesamt. 2006. Anpassung an Klimaänderungen in Deutschland - Regionale Szenarien und nationale Aufgaben. Hintergrundpapier .

Wagner, H., and M. Fortin. 2005. Spatial Analysis of Landscapes: Concepts and Statistics. Ecology 86:1975–1987.

Wickham, H. 2009. ggplot2: Elegant Graphics for Data Analysis. Springer New York. URL `http://had.co.nz/ggplot2/book`, R package version 0.9.0.

Wickham, H. 2012. devtools: Tools to Make Developing R Code Easier URL `http://CRAN.R-project.org/package=devtools`, R package version 0.6.

Wood, S. N. 2012. Fast Stable Restricted Maximum Likelihood and Marginal Likelihood Estimation of Semiparametric Generalized Linear Models. Journal of the Royal Statistical Society (B) 73:3–36. URL `http://CRAN.R-project.org/package=mgcv`, R package version 1.7-13.

Yee, T., and N. Mitchell. 1991. Generalized Additive Models in Plant Ecology. Journal of vegetation science 2:587–602.

A. Appendix

```
"%ll%" <- function(bl1, bl2){

        if(is.list(bl1) && !inherits(bl1, "blg"))
                return(lapply(bl1, "%ll%", bl2 = bl2))

        if(is.list(bl2) && !inherits(bl2, "blg"))
                return(lapply(bl2, "%ll%", bl1 = bl1))

        ### set baselearner name
        cll <- paste(bl1$get_call(), "%ll%",
                        bl2$get_call(), collapse = "")
        cll <- paste(bl1$get_call())


        ## test if baselearners
        stopifnot(inherits(bl1, "blg"))
        stopifnot(inherits(bl2, "blg"))

        ## build model.frame
        mf <- cbind(model.frame(bl1), model.frame(bl2))

        ## index
        index <- NULL

        ## vary
        vary <- ""

        ## return
        ret <- list(

                ## model.frame
                model.frame = function() mf,
                ## function
                get_call = function(){
                        #cll <- deparse(cll, width.cutoff = 500L)
                        if (length(cll) > 1)
                                cll <- paste(cll, collapse = "")
                        cll
                },

                ## model.frame data
                get_data = function() mf,

                ## index
                get_index = function() index,
                get_vary = function() vary,

                ## get the names of the model.frame
                get_names = function() colnames(mf),

                ## change the names of the model.frame
                set_names = function(value) attr(mf, "names") <<- value
        )
## class return
class(ret) <- "blg"


## read arguments
args1 <- environment(bl1$dpp)$args
args2 <- environment(bl2$dpp)$args

## lambda
l1 <- args1$lambda
l2 <- args2$lambda
if (!is.null(l1) && !is.null(l2)){
        args <- list(lambda = 1, df = NULL)
}
else{
        args <- list(lambda = NULL,
                df = ifelse(is.null(args1$df), 0, args1$df) +
                        ifelse(is.null(args2$df), 0, args2$df))
}
```

```
        ### Xfun
        Xfun <- function(mf, vary, args){

                ## create x and k matrices
                newX1 <- environment(bl1$dpp)$newX
                newX2 <- environment(bl2$dpp)$newX

                ## extract x and k matrices
                X1 <- newX1(mf[, bl1$get_names(), drop = FALSE])
                K1 <- X1$K
                if (!is.null(l1)) K1 <- l1 * K1
                X1 <- X1$X

                X2 <- newX2(mf[, bl2$get_names(), drop = FALSE])
                K2 <- X2$K
                if (!is.null(l2)) K2 <- l2 * K2
                X2 <- X2$X

                ## make x1 orthogonal to x2
                # qr.resid(qr, y)
                # X1orth <- qr.resid(qr(X2), X1)
                # X1orth2 <-  (I - (X2 (X2'X2)^-1 X2') X1)
                # X1 <- qr.resid(qr(X2), X1) = I - (X2 (X2'X2)^-1 X2') X1
                X1orth <- qr.resid(qr(X2), X1)

                ## new design matrix X
                X <- X1orth

                ## new penalty matrix K
                K <- K1

                ## return
                list(X = X, K = K)
        }

        ret$dpp <- bl_lin(ret, Xfun = Xfun, args = args)

        return(ret)
}
```

B. CD-ROM Content

The attached CD-ROM contains the whole R-Code used in this thesis, as well as the data-sets, the generated graphics and a digital version of thesis.
A small overview over the content of the included **folders** is given below:

- **application**: All the R-files and the raw data of the application.
- **results**: All results and generated graphics of the simulation and the application in .pdf format.
- **simulation**: All the R-files for the data generating process and the simulation.
- masterthesis.pdf
- readme.txt

Affidavit

I, Christian Lindenlaub, hereby declare that this master-thesis in question was written single-handed and no further as the denounced resources and sources were employed.

Munich, December 26, 2012

(Christian Lindenlaub)