Ludwig-Maximilians-Universität München

Department of Statistics

Master Thesis

Scalar-on-Image Regression using iterative Methods

Author:
Raphael Rehms

Supervisor:
Prof. Dr. Volker Schmid,
Working group Bioimaging

March 18, 2019

Abstract

In Scalar-on-Image regression, one often has to deal with situations where the number of regression coefficients exceeds the number of observations by far. This leads to an identification problem – the system of equations has no unique solution. To obtain a reasonable result, additional assumptions must be imposed. In a Bayesian approach, this can be done by using a Gaussian Markov random field as prior for the regression coefficients. This implies smoothness over the coefficient image, i.e. adjacent pixels or voxels are assumed to have similar values. This thesis introduces Gaussian Markov random fields in general and how they can used as prior in a full Bayesian approach for Scalar-on-Image regression. Additionally, it will be described how inference can be done by using iterative methods, also known as Markov Chain Monte Carlo. Furthermore, several simulation studies shall investigate different aspects of the described methods. The main focus lies on the influence of hyperparameters and the incorporated neighbours for a Gaussian Markov random field prior. The simulation studies are carried out for different types of coefficient images, reflecting various characteristics such as smooth or sparse structures.

Contents

List of Figures

List of Tables

1. Introduction

Over time, the progress in many different fields of science increases rapidly which leads to many highly specialized methods. Due to this fact, classical and widely used statistical techniques often are not sufficient and many new tailored approaches for particular research fields appear.

One of these approaches is *Scalar-on-Image regression*. The goal of Scalar-on-Image regression is, as the name describes, to regress an input image on a scalar target variable. This tool can be used in many different fields and applications where one wants to find structures in these images. Scalar-on-Image regression is widely used in the field of neuro- or brain imaging. See for instance Goldsmith et al. [2014], where cognitive outcomes are regressed on measures of white-matter microstructure at every voxel of a three-dimensional image of the corpus callosum.

A main advantage of Scalar-on-Image regression is the good interpretability of the estimated regressors: Since the goal to find an appropriate mapping from a higher dimensional to a one-dimensional space is done (as in common regression tasks) in a well understandably manner, the result provides a high interpretability in the shape of a *coefficient image*. This image gives the opportunity to see which areas (i.e. which pixels or voxels) of an image are associated with the scalar target variable.

To define Scalar-on-Image regression as a classical regression task, every pixel or voxel is assumed to have its own regression coefficient β. A distinction to normal linear regression problems is basically only given by the number of used regression coefficients and their implicitly given spatial arrangement on a lattice. Since every pixel or voxel is represented through its own regressor, the number of all regression coefficients grows exponentially with the resolution of an image. Consider as a simple example a two-dimensional image with a total amount of $L = m \times m,\ m \in \mathbb{N}$ pixels. If the side length m is multiplied by two, the number of pixels and therefore the number ob regressors β quadruples itself. If this is done for a three-dimensional image ($L = m \times m \times m$), the number of voxels is eight times as high. Hence, this fact can be seen as a manifestation of the 'Curse of dimensionality' (see e.g. Bishop [2006]).

Due to the fast growth in technology of generating high resolution images[1], the number of regression coefficients is in the most cases much higher than the number of observations N. This kind of problem, is also known as *large p small n problem* (Chakraborty et al. [2012]), or short $p >> n$ problem (Happ et al. [2018]).

For regression, this means mathematically that the system of equations is non-identifiable. Therefore one has to make additional assumptions for the coefficients. Many different

[1] See e.g. Penny et al. [2011] Chapter 1: 'A short history of SPM' for an overview of imaging in neuroscience.

solutions to overcome this problem exist. In the context of $p >> n$, there are some popular and useful extensions of the simple linear model which add a regularization term to the standard regression model. The first which shall be mentioned, is the well known *LASSO* (**l**east **a**bsolute **s**hrinkage and **s**election **o**perator). LASSO adds an L_1-penalty for the regression coefficients to the classical OLS problem (Tibshirani [1996]). This type of penalty has the characteristic to select several coefficients and shrinks all other to zero (Friedman et al. [2010]).
If an L_2-Penalty is chosen, the regression method is called *ridge regression* (Hoerl and Kennard [1970]). Here all regression coefficients will get shrunk towards zero, but will remain in the model. Both methods have there own advantages and disadvantages, depending on the specific situation or use case.
If one wants to use a combination of both penalties, this can be done by using a linear combination of them (Zou and Hastie [2005]). This is also known as *elastic net*.
Both, the LASSO and ridge are specific cases of the more general *bridge regression* (Frank and Friedman [1993]), which adds the more universal regularization term $J(\beta, \gamma) = \sum_{j=1}^{p} ||\beta_j||_\gamma$ to the OLS problem. LASSO can be achieved by setting $\gamma = 1$ and ridge regression by setting $\gamma = 2$.[2]
Using a Bayesian perspective, a further penalty term $J(\cdot)$ for the regressors in the OLS problem can be obtained by expanding the problem with an additional prior assumption concerning the regression coefficients. The general density of a prior, which is equivalent to the bridge regression, can be found in Fu [1998]. The density depends, just as the penalty term $J(\cdot)$ on a parameter γ. For $\gamma = 1$, one can deduce the probability density function of a Laplace distribution. If one is using this distribution as prior for the regression coefficients, one can derive the LASSO problem. For $\gamma = 2$ one can derive the probability density function of a Normal distribution. Hence, this prior corresponds to a ridge penalty.[3]

The use of the described prior distributions has one thing in common: The absolute values of the estimated β coefficients will be smaller than without the additional prior assumption. Therefore the used priors make the assumption that the true coefficients are located around zero; they get 'pushed' towards zero. In some cases, this can be a reasonable decision. In general, one *must* impose additional prior assumptions to handle the identifiability issue. In some other cases, one can choose more reasonable priors. With regard to the main topic of this thesis, it is possible to choose a prior called *Gaussian Markov random field* (GMRF), which is a Gaussian distribution with

[2] The general form of the bridge regression existed before LASSO, but γ remained there as a tuning parameter (Fu [1998]). Tibshirani [1996] introduced LASSO later as a special case of bridge regression.
[3] The elastic net can also be formulated in a Bayesian manner. The prior the is then given by $\pi(\beta) \propto \exp(-\lambda_1||\beta||_1 - \lambda_2||\beta||_2^2$ (Li and Lin [2010]).

a specific structure of its inverse covariance matrix. This type of prior distribution gives the possibility to take the spacial structure of images into account. For this one requires, that the image is *smooth* (up to a certain point). In this case, smoothness means that pixels at the same location tend to have similar values. To do inference in a regression task with a GMRF prior using a Bayesian framework, iterative methods have to be applied, which are also known as *Markov Chain Monte Carlo* (MCMC). These methods represent a popular technique among statisticians, in case the normalizing constant of a density is not available and numerical integration is not feasible due to an extremely high number of involved parameters (Brooks et al. [2011]).

A general and highly flexible approach for many different types of regression tasks are *Structured additive regression* (STAR) models (Fahrmeir et al. [2004]). STAR models provide an unified framework for a lot of different regression problems including nonlinear and spatial effects, linear and nonlinear interactions between covariates, individual-specific random intercepts and slopes (Fahrmeir et al. [2007]). Since Scalar-on-Image regression with GMRF priors is a strategy which makes use of the spatial structure of an image, a formulation as a STAR model is unproblematic. Thus, Scalar-on-Image regression tasks can be carried out by using software which is developed for STAR models.
Despite these models provide a practicable approach to Scalar-on-Image regression, one often faces situations where a large number of coefficients has to be estimated. This leads to a high computational burden. If only limited hardware resources are available, it is necessary to use strategies, to deal with this limitation, e.g. by using approximation techniques (Schmidt et al. [2017]).

The general idea of this thesis consists in the application of Scalar-on-Image regression using a full Bayesian approach by examining various aspects through different simulation studies.
The thesis is structured as follows: Section 2 describes the Scalar-on-Image regression problem, introduces Gaussian Markov random fields and connects them to each other in a Bayesian manner. In section 3, it will be described how inference in this context can be done. Afterwards, section 4 presents different simulation studies examining various aspects of the presented method. Section 5 gives a short description of alternative approaches to Scalar-on-Image regression followed by a summary with discussion in section 6.

2. Scalar-on-Image Regression

2.1. Representation of Images

In this thesis, an image is considered as a regular lattice, defined by a tuple of integers $n = (n_1, \dots n_d)$, where each element specifies the number of pixels in a direction. Hence, the length of the tuple yields the dimension of the image. The focus will be on two- and three-dimensional images on a grey scale, i.e. $d \in \{2, 3\}$.[4] Each pixel $x_l \in \mathbb{R}$ will be subscripted to define its location in the image. In total, an image has $n_1 \times \cdots \times n_d = L$ pixels. For a single subscript, x_l represents a pixel in a vectorized image (when no other information is given). If a double subscript is used, i.e. $x_{i,l}$ it defines the l-th pixel of the i-th observation. If the subscript is not separated with a comma, i.e. x_{kl} it represents the set of random variables located at i and j in the vectorized image of an arbitrary observation.[5]

2.2. Regression with Images as Covariates

The general form of Scalar-on-Image regression in this thesis is considered to have the following form (as defined in Happ et al. [2018]):

$$y_i = \sum_{j=1}^{p} w_{i,j}\alpha_j + \sum_{l=1}^{L} x_{i,l}\beta_l + \varepsilon_i, \quad i = 1, \dots, N \tag{2.1}$$

where y_i is a scalar response for each of N images. y_i is considered to be a linear combination of scalar covariates $w_i \in \mathbb{R}^p$, where each element gets multiplied with a weight α_j, the j-th element of a vector $\alpha \in \mathbb{R}^p$ [6] and the vectorized images multiplied by the vectorized coefficient image (which must have the same dimension unvectorized and therefore the same length L in vectorized form). Alternatively (2.1) can be written in matrix notation:

$$y = W\alpha + X\beta + \varepsilon \tag{2.2}$$

with the vector of responses $y = (y_1, \cdots, y_N)$, $W \in \mathbb{R}^{N \times p}$ as matrix of scalar covariates and $X \in \mathbb{R}^{N \times L}$, the row-wise vectorized images with the vectorized coefficient image $\beta = (\beta_1, \dots \beta_L)$. The error term ε is assumed to be normal distributed, i.e. $\varepsilon_i \overset{iid}{\sim} N(0, \sigma_\varepsilon^2)$ (in (2.1)) or $\varepsilon \sim N(0, \sigma_\varepsilon^2 I_N)$ (in (2.2)). Note that the first element $w_{i,1}$ or the first

[4] Images can also exist in higher dimensions. One example would be a fMRI time series where each voxel of a three-dimensional image is observed at several time points. An analysis of this kind of images can be found for example be found in Penny et al. [2005]

[5] Note that a double subscripted matrix, e.g. Q_{ij} defines the element in row i and column j.

[6] The first part is therefore writable as $w_i^T\alpha$.

column in W takes by convention a value of 1 to model an intercept.
Also note that in this thesis, there will be no other covariates than the images itself and an intercept. Therefore (2.1) reduces to

$$y_i = \alpha + \sum_{l=1}^{L} x_{i,l}\beta_l + \varepsilon_i, \quad i = 1, \ldots, N. \tag{2.3}$$

As in the first section mentioned, this is a classical regression problem. The only difference lies in the total amount of parameters which shall be estimated. Since the system of equations is not identifiable, one has to impose additional assumptions to obtain a unique solution. Using a Bayesian framework, this can be done by assuming a prior distribution for the β-coefficients.
A popular and reasonable class of priors is a *Gaussian Markov random field* (GMRF). This assumes smoothness in the image, i.e. neighboring pixels (or voxels) tend to have similar values. With a smoothness assumption, the accuracy of relevant predictors might get improved since the spatial arrangement in images can be exploited, e.g. in brain images (Reiss et al. [2015]). Next, GMRFs will be introduced and get connect to Scalar-on-Image regression.

2.3. Gaussian Markov Random Fields

In this section, GMRFs (firstly proper, then intrinsic) will be introduced. Furthermore, it will be described why they are meaningful as prior for Scalar-on-Image regression.

2.3.1. Proper GMRFs

A GMRF is basically nothing else than a multivariate Normal distribution with some properties concerning the covariance matrix Σ (Rue and Held [2005]). Given an arbitrary random vector $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$ with respect to an (undirected) graph[7] $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the density of x is given by

$$\pi(x) = (2\pi)^{-n/2}|Q|^{1/2}\exp\left(-\frac{1}{2}(x-\mu)^T Q(x-\mu)\right)$$

where $Q = \Sigma^{-1}$ is the *precision matrix* of x. For the entries of Q it holds:

$$Q_{ij} \neq 0 \quad \Longleftrightarrow \quad \{i,j\} \in \mathcal{E} \quad \forall i \neq j.$$

[7] An undirected graph $\mathcal{G}$ is a tuple $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of nodes on $\mathcal{G}$ and $\mathcal{E}$ is the set of edges $\{i,j\}$ with $i,j \in \mathcal{V}$ and $i \neq j$ (Rue and Held [2005]).

This means that a GMRF is a Normal distribution where an undirected graph defines the structure of the covariance matrix Σ, or more precisely, the precision matrix Q. Furthermore, it holds for GMRFs:

$$x_i \perp x_j | x_{-ij} \Longleftrightarrow Q_{ij} = 0$$

$\perp$ means that x_i and x_j are stochastically independent given x_{-ij}. The negative sign in the index of x_{-ij} denotes x without the elements in the subscript, i.e. x_{-ij} is the random vector $(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{j-1}, x_{j+1}, \ldots x_n)$. The precision matrix Q is in many cases, in Scalar-on-Image regression in particular, very sparse. Therefore the most entries of Q are zero and an efficient representation in memory is possible by only saving non-zero values. The property of sparsity does not transcribes to its inverse Σ (which is in general a dense matrix). From this point of view, it could be advantageous to represent GMRFs with a precision matrix.
If x is a GMRF with respect to a graph $\mathcal{G}$ with mean μ and precision matrix $Q > 0$, then the conditional expectation, precision and correlation for an element x_i is given by

$$E(x_i|x_{-i}) = \mu_i - \frac{1}{Q_{ii}} \sum_{j:j\sim i} Q_{ij}(x_j - \mu_j) \tag{2.4}$$

$$Prec(x_i|x_{-i}) = Q_{ii} \tag{2.5}$$

$$Corr(x_i, x_j)|x_{-ij}) = -\frac{Q_{ij}}{\sqrt{Q_{ii}Q_{jj}}} \quad i \neq j. \tag{2.6}$$

Here, $j \sim i$ indicates that element j and i are *neighbours*. Therefore $j : j \sim i$ can be expressed as the set $\{j \in \mathcal{V} : \{i, j\} \in \mathcal{E}\}$. Thus the set includes all elements which have an edge originating from element i (and therefore all neighbours of x_i). The proof can be found in Rue and Held [2005], pp. 23-24. Since the diagonal elements of Q represent the conditional precision and the off-diagonal elements the conditional correlation (with proper scaling through the denominator), all elements from Q have a useful interpretation which is different from Σ.[8]
It is possible to generalize the results from above (2.4) - (2.6) to more elements than one x_i. For this, the graph $\mathcal{G}$ is divided in two subgraphs $\mathcal{G}^A$ and $\mathcal{G}^B$. Then $\mathcal{V}$ can be split in two subsets: $A \subset \mathcal{V}$ and $B = \mathcal{V} \setminus A$ where $A, B \neq \emptyset$. This makes it also possible to calculate the conditional mean $\mu_{A|B}$ and precision matrix $Q_{A|B}$ (which are basically the same as above but multivariate, see Rue and Held [2005]).
Another way to define GMRFs is given by using full conditionals $\{\pi(x_i|x_{-i})\}$ as done

[8] Σ provides information about the marginal distributions.

by Besag [1974, 1975]. For this one has to specify

$$E(x_i|x_{-i}) = \mu_i - \sum_{j:j\sim i} \omega_{ij}(x_j - \mu_j) \tag{2.7}$$

$$Prec(x_i|x_{-i}) = \kappa_i > 0 \tag{2.8}$$

for $i = 1, \ldots, n$, for some $\{\omega_{ij}, i \neq j\}$ and vectors μ and κ. Since the neighbourhood between two locations is symmetric, it must hold that if $\omega_{ij} \neq 0$, then also $\omega_{ji} \neq 0$. By comparing (2.7) and (2.8) with (2.4) and (2.5), it is obvious to choose for the precision matrix Q

$$\begin{aligned} Q_{ii} &= \kappa_i \qquad \text{and} \\ Q_{ij} &= \kappa_i \omega_{ij} \end{aligned}$$

with the restriction that $\kappa_i\omega_{ij} = \kappa_j\omega_{ji},\ i \neq j, Q > 0$ (hence Q has to be symmetric). A full proof for this can also be found in Rue and Held [2005] or partially, but with some useful and intuitive implications, in Fahrmeir and Kneib [2011]. For the proof it is required that the *positivity* condition holds[9] (which is in general fulfilled in reality). Under positivity, it is possible to use *Brook's lemma* which allows a factorization – known as Brook's expansion – of a joint density by using an arbitrary fixed x' which has the same support as x. Then the full conditionals from (2.7) and (2.8) define a GMRF.

2.3.2. Intrinsic GMRFs

An intrinsic GMRF (IGMRF) of order k is a normal GMRF where the precision matrix Q has no full rank, i.e. $\text{rk}(Q) = n - k$. The density is given by

$$\pi(x) = (2\pi)^{-(n-k)/2}(|Q|^*)^{1/2}\exp\left(-\frac{1}{2}(x-\mu)^T Q(x-\mu)\right). \tag{2.9}$$

Since Q has no full rank, it does not have a normal determinant. Therefore, $|\cdot|^*$ defines a generalized determinant which is the product of all nonzero eigenvalues of Q. (2.9) is an improper density [10], but for Bayesian inference it is possible to use it in a reasonable manner. Furthermore, for an IGMRF (of first order) it must hold that $Q\mathbf{1} = 0$ where $\mathbf{1}$ is a vector of ones; the rowsums have to be 0, i.e. $\sum_j Q_{ij} = 0,\ \forall i$. The idea of intrinsic GMRFs can be motivated by a univariate random walk (firstly of

[9] The positivity condition states that if $p(x_i) > 0,\ i = 1, \ldots, n$, then also $p(x) > 0$. Therefore the support of the joint distribution of x has to be the cartesian product of the support of all individual marginals of x.

[10] Rue and Held [2005] give an interpretation of such an improper distribution in section 3.2 (pp. 89-93).

order one). For this consider a random vector $x = (x_1, \ldots, x_n)$ where the increments are normally distributed:

$$\Delta x_i \overset{iid}{\sim} N(0, \kappa^{-1}), \quad i = 1, \ldots n-1$$

where the (forward) difference operator of first order Δ is defined as $\Delta x_i = x_{i+1} - x_i$. Higher orders are defined recursively as $\Delta^k x_i = \Delta\Delta^{k-1} x_i$. The k-th differences can be interpreted as an approximation of the k-th derivative (see Rue and Held [2005], p. 87 for an intuitive explanation). The density of x is then given by

$$\begin{aligned}
\pi(x|\kappa) &\propto \kappa^{(n-1)/2} \exp\left(-\frac{\kappa}{2}\sum_{i=1}^{n-1}(\Delta x_i)^2\right) \\
&= \kappa^{(n-1)/2} \exp\left(-\frac{\kappa}{2}\sum_{i=1}^{n-1}(x_{i+1} - x_i)^2\right) \\
&= \kappa^{(n-1)/2} \exp\left(-\frac{\kappa}{2} x^T R x\right) \\
&= \kappa^{(n-1)/2} \exp\left(-\frac{1}{2} x^T Q x\right)
\end{aligned}$$

with $Q = \kappa R$ and the *structure matrix* R which is defined as

$$R = \begin{pmatrix}
1 & -1 & & & \\
-1 & 2 & -1 & & \\
& \ddots & \ddots & \ddots & \\
& & -1 & 2 & -1 \\
& & & -1 & 1
\end{pmatrix} \tag{2.10}$$

The form of R can be easily derived using the definition of the quadratic form of a matrix and the difference operator for a whole vector denoted with D:

$$\sum_{i=1}^{n-1}(\Delta x_i)^2 = (Dx)^T(Dx) = x^T D^T D x = x^T R x$$

where D is mostly zero with dimension $(n-1) \times n$:

$$D = \begin{pmatrix}
-1 & 1 & & & \\
& -1 & 1 & & \\
& & \ddots & \ddots & \\
& & & -1 & 1
\end{pmatrix}$$

The rank of Q is $n-1$ and the rows sum up to zero. Therefore an univariate random walk is an IGMRF of order one.
As in Fahrmeir and Kneib [2011] stated, the distinction between a proper GMRF and an IGMRF can be made from this random walk. The first-order random walk can be expressed as

$$x_t = x_{t-1} + \varepsilon_t, \qquad \varepsilon \sim N(0, \kappa^{-1}).$$

To obtain a proper GMRF it must be modified by multiplying x_{t-1} with a constrained factor ρ which 'pushes' x_{t-1} towards zero:

$$x_t = \rho x_{t-1} + \varepsilon_t, \qquad \varepsilon \sim N(0, \kappa^{-1}), \quad -1 < \rho < 1.$$

IGMRFs of higher orders can be constructed using higher increments. For example, a second-order IGMRF can be constructed by using Δ^2. The density is then given by

$$\begin{aligned}
\pi(x|\kappa) &\propto \kappa^{(n-1)/2} \exp\left(-\frac{\kappa}{2}\sum_{i=1}^{n-2}(\Delta^2 x_i)^2\right) \\
&= \kappa^{(n-1)/2} \exp\left(-\frac{\kappa}{2}\sum_{i=1}^{n-1}(x_i - 2x_{i+1} + x_{i+2})^2\right) \\
&= \kappa^{(n-1)/2} \exp\left(-\frac{1}{2}x^T Q x\right)
\end{aligned}$$

where again $Q = \kappa R$. The structure matrix is now given by

$$R = \begin{pmatrix}
1 & -2 & 1 & & & & \\
-2 & 5 & -4 & 1 & & & \\
1 & -4 & 6 & -4 & 1 & & \\
 & \ddots & \ddots & \ddots & \ddots & \ddots & \\
 & & 1 & -4 & 6 & -4 & 1 \\
 & & & 1 & -4 & 5 & -2 \\
 & & & & 1 & -2 & 1
\end{pmatrix} \tag{2.11}$$

R can be constructed as it is done in (2.10). The rank of Q is now $n-2$.

2.4. GMRFs in Scalar-on-Image Regression

The examples of random walks in the previous section are IGMRFs on a regular lattice with dimension one. In this section, they will be generalized to higher dimensions.

2.4.1. IGMRFs on regular Lattices in higher Dimensions

A first possibility to model structure matrices for higher dimensions is given by using the first neighbours on a regular lattice (Clayton [1995], Rue and Held [2005]). For this, one can use (2.4) and (2.5). Let $\mu = 0$, then the conditional expectation is

$$E(x_i|x_{-i}) = -\frac{1}{Q_{ii}} \sum_{j:j\sim i} Q_{ij}(x_j - \mu_j).$$

If one requires equal weights for all neighbours, the result is a precision matrix $Q = \kappa R$ with entries

$$\begin{aligned} R_{ij} &= -\mathbb{1}\{i \sim j\} \\ R_{ii} &= \sum_{i \neq j} \mathbb{1}\{i \sim j\} \end{aligned} \tag{2.12}$$

where $\mathbb{1}\{i \sim j\}$ is one, if locations i and j are adjacent and zero otherwise. Therefore, the main diagonal elements R_{ii} simply counts the number of all neighbours and the off-diagonal elements Q_{ij} indicates their neighbourhood to a location with the value -1. Comparing (2.12) with the structure matrix of a one-dimensional random walk as (2.10), it turns out that they are completely equivalent. Therefore (2.12) gives a rule to construct IGMRFs of first order for higher dimensions.
Moreover, it is possible to use one-dimensional random walks (also of higher order) to construct structure matrices for IGMRFs of any dimension. The one-dimensional random walk can easily be associated with a stochastic process observed over time (with equal distant discrete time points). A two- or three-dimensional lattice corresponds to a set of discrete points located on a plane or in a space. Therefore it would be reasonable to use the concept of the one-dimensional random walk also for higher dimensions. This can be done using interactions (Rue and Held [2005], Clayton [1995]). One can regard a regular lattice of dimension $d \in \{2, 3\}$ as an interaction of two one-dimensional grids, a vertical and an horizontal one. Following this idea, the resulting differences of differences model has the increments

$$\begin{aligned} \Delta_{(1,0)}\Delta_{(0,1)}x_{ij} &\overset{iid}{\sim} N(0, \kappa^{-1}), \\ i &= 1, \ldots, n_1 - 1 \\ j &= 1, \ldots, n_2 - 1 \end{aligned}$$

where n_1 corresponds to the number of locations in the horizontal and n_2 in the vertical direction. The subscript of Δ indicates the direction of the grid where the difference operator is acting on, i.e. $\Delta_{(1,0)}$ denotes the horizontal direction and $\Delta_{(0,1)}$ the vertical.

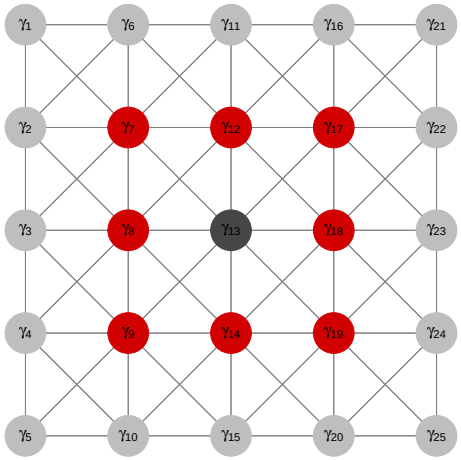

Fig. 2.1 All first neighbours for two-dimensional lattice marked in red. Note that the shown neighbourhood differs at the margins.

Hence $\Delta_{(1,0)}\Delta_{(0,1)}x_{ij}$ can also represented as $x_{i+1,j+1} - x_{i+1,j} - x_{i,j+1} + x_{i,j}$. The density of this IGMRF is then given by

$$\begin{aligned}\pi(x|\kappa) &\propto \kappa^{\frac{(n_1-1)(n_t wo-1)}{2}} \exp\left(-\frac{\kappa}{2}\sum_{i=1}^{n_1-1}\sum_{j=1}^{n_2-1}(\Delta_{(1,0)}\Delta_{(0,1)}x_{ij})^2\right) \\ &= \kappa^{\frac{(n_1-1)(n_2-1)}{2}} \exp\left(-\frac{\kappa}{2}x^T R x\right).\end{aligned} \tag{2.13}$$

The conditional mean depends on the eight nearest neighbours and is, by using (2.4) and setting $\mu = 0$, given by [11]

$$\begin{aligned}E(x_i|x_{-i}) = -\frac{1}{4\kappa}\{ &-2(x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1}) + \\ &(x_{i-1,j-1} + x_{i+1,j-1} + x_{i-1,j+1} + x_{i+1,j+1})\}.\end{aligned} \tag{2.14}$$

The conditional precision is according to (2.5) given by

$$Prec(x_i|x_{-i}) = 4\kappa.$$

Figure 2.1 illustrates the included neighbourhood. Note that these results do not hold at the margins since there are not eight adjacent neighbours. To get a better intuition why the form of the conditional expectation and precision is quite intuitive, it can be helpful to look at the structure matrix of (2.13).
The structure matrix R can be constructed from two one-dimensional random walks by

[11] Note that the comma separation in the subscript is done due to clarity and is therefore different from the definition in section 2.1: The first index defines the location in the vertical direction, the second in the horizontal.

using the Kronecker product (Rue and Held [2005]):

$$R = R_1 \otimes R_2 \tag{2.15}$$

where R_1 and R_2 are defined as in (2.10). The subscript is needed to distinguish between the length of the one-dimensional random walks. Since the ranks of the structure matrices are $\text{rk}(R_1) = n_1 - 1$ and $\text{rk}(R_2) = n_2 - 1$, it follows that $\text{rk}(R) = (n_1 - 1)(n_2 - 1)$ which is a basic property of the Kronecker product (Steeb and Shi [1997]). Comparing to other interaction structures, which will be described in the following section, this leads to a high loss in rank concerning Q.
Modeling interactions by using Kronecker products is not limited to spacial effects. It is also possible to model interactions between the time and spacial domain as it is done e.g. in Gössl et al. [2001].

Since in Scalar-on-Image regression one has to deal with two- or three-dimensional images, the construction of IGMRFs of higher order from one-dimensional IGMRFs by using Kronecker products seems to be a well-suited approach. Referring to (2.15), the Kronecker product is used to model a full interaction in each direction. This means, that all combinations in each dimension will be considered.[12]. This also leads, as stated above, to a high rank deficit in the structure matrix. Therefore one could consider modeling only the *direct* dependencies. This can be done by using a Kronecker sum:

$$R = R_1 \oplus R_2 = R_1 \otimes I_{n_2} + I_{n_1} \otimes R_2 \tag{2.16}$$

where I_{n_k} is the identity matrix with the dimensions $n_k,\ k \in 1, 2$ defining the length of the image in the k-th dimension. The eigenvalues of (2.16) are given according to Steeb and Shi [1997] by $\lambda_i^{(1)} + \lambda_j^{(2)},\ i = 1, \ldots, n_1,\ j = 1, \ldots n_2$ where $\lambda_l^{(k)}$ is the l-th eigenvalue of R_k. Since R_1 and R_2 are the structure matrices from one-dimensional random walks, the rank for the Kronecker sum is given by

$$\text{rk}(R_1 \oplus R_2) = n_1 n_2 - 1.$$

Hence, there is no additional loss in rank by using only direct interaction from one-dimensional IGMRFs. The direct interaction takes only four neighbours into account (± 1 in each dimension). The conditional mean is

$$E(x_i | x_{-i}) = -\frac{1}{4\kappa}(x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1})$$

[12] This leads to $3^d - 1$ neighbours since the dependencies are present for all combinations (direct *and* indirect).

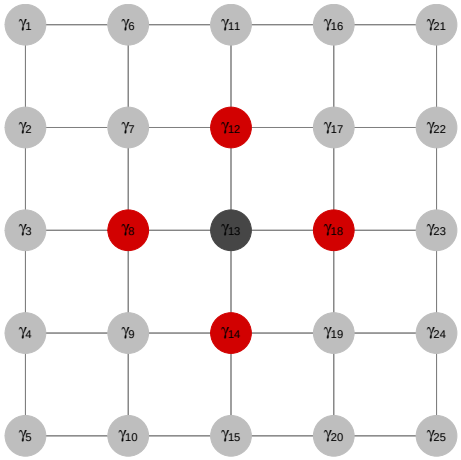

Fig. 2.2 First direct neighbours for two-dimensional lattice marked in red. Note that the shown neighbourhood differs at the margins.

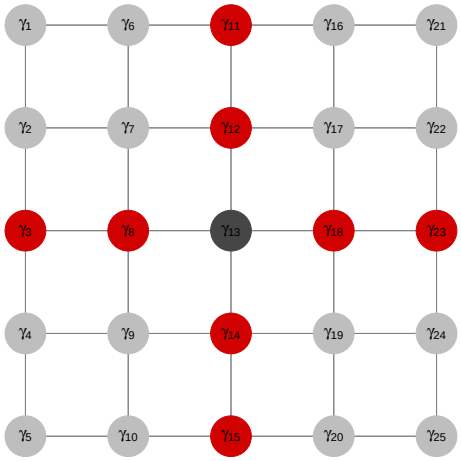

Fig. 2.3 Second direct neighbours for two-dimensional lattice marked in red. Note that the shown neighbourhood differs at the margins.

and the precision

$$Prec(x_i|x_{-i}) = 4\kappa.$$

Figure 2.2 illustrates the included neighbourhood. It can be seen easily, that now the indirect (diagonal) neighbours are no longer taken into account. The last interaction which will be presented in this thesis, is the same as (2.16), but using structure matrices from one-dimensional IGMRFs of second order as previously shown in (2.11). Figure 2.3 illustrates the considered neighbourhood. Also in this case, only the direct neighbours will be used (no indirect interactions) which preserves a higher rank.

The shown interactions are used for two-dimensional images. Since images also exist in higher dimensions, the shown interactions can also be adapted to this circumstance. By

only considering direct interaction, the three-dimensional analogue of (2.16) is

$$\begin{aligned} R &= R_1 \otimes I_{n_2} \otimes I_{n_3} + \\ & \quad I_{n_1} \otimes R_2 \otimes I_{n_3} + \\ & \quad I_{n_1} \otimes I_{n_2} \otimes R_3 \end{aligned}$$

where one loses again only the number of ranks from the one-dimensional IGMRFs (i.e. one for first order and two for second order).
If one wants to take all interactions into account, the structure matrix can be calculated by

$$R = R_1 \otimes R_2 \otimes R_3.$$

Using this neighbourhood structure, one has to bear in mind that the loss of rank is also higher than in two dimensions:

$$\mathrm{rk}(R) = (n_1 - 1)(n_2 - 1)(n_3 - 1).$$

2.4.2. IGMRFs as Prior in Scalar-on-Image Regression

Since Scalar-on-Image regression is basically a normal regression problem where one has to deal with an identification issue, additional prior assumptions are crucial to obtain useful estimates for the regression coefficients β. As already described in the first section, one can assume smoothness in the image. Therefore it will be required that adjacent pixels (or voxels) tend to have similar values. There are existing different methods to obtain smoothness. Using a Bayesian framework, this can be done by incorporating additional prior information in the regression model. Smoothness can be obtained by using IGMRFs to model the dependency between the β coefficients.
In a full Bayesian approach one has to specify priors over all variables in (2.1) or (2.2). For this, it is assumed that α, β and σ_ε^2 are independent. The distributions for y, α and σ_ε^2 can be chosen as follows. y is Normal, since the ε is Normal. α is assumed to be constant and therefore uninformative. σ_ε^2 is assumed to follow an Inverse-gamma distribution. More precisely:

$$\begin{aligned} y|\alpha, \beta, \sigma_\varepsilon^2 &\sim N(W\alpha + X\beta, \sigma_\varepsilon^2 I_N) \\ \pi(\alpha) &\propto \text{const} \\ \sigma_\varepsilon^2 &\sim IG(a_\varepsilon, b_\varepsilon) \end{aligned}$$

where the shape a_ε and rate b_ε are hyperparameters for the Inverse-gamma distribution ($IG(a_\varepsilon, b_\varepsilon)$). The density of σ_ε^2 is given by

$$\pi(\sigma_\varepsilon^2) = \frac{b_\varepsilon^{a_\varepsilon}}{\Gamma(a_\varepsilon)}(\sigma_\varepsilon^2)^{-a_\varepsilon-1}\exp\left(-\frac{b_\varepsilon}{\sigma_\varepsilon^2}\right).$$

To obtain smoothness, an IGMRF is chosen as prior for the coefficient image β:

$$\beta|\kappa \sim \text{IGMRF}(\kappa R). \tag{2.17}$$

Therefore β is Normal distributed with zero mean and the implicitly defined covariance matrix via the structure matrix R:

$$\Sigma = Q^{-1} = (\kappa R)^{-1}$$

In a full Bayesian approach one has to specify an additional distribution for the precision parameter κ, i.e. κ follows a Gamma distribution with shape a_κ and rate b_κ:

$$\kappa \sim Ga(a_\kappa, b_\kappa).$$

Therefore the density is given by

$$\pi(\kappa) = \frac{b_\kappa^{a_\kappa}}{\Gamma(a_\kappa)}\kappa^{a_\kappa-1}\exp(-b_\kappa\kappa).$$

It is also possible to formulate the prior distribution in terms of the variance $\sigma_\beta^2 = \kappa^{-1}$ using an Inverse-gamma distribution as it is done for the variance parameter of the error term σ_ε^2. Moreover, it is of course possible to use a Gamma distribution as prior for the error term $(\sigma_\varepsilon^2)^{-1}$. See e.g. Happ et al. [2018] where both, the error term and the inverse precision is defined with an Inverse-gamma distribution.
For (2.17), it is also required to impose a neighbourhood structure for $Q = \kappa R$. Three potential candidates were described in section 2.4. How the different candidates influence the estimation will be later examined in section 4.3.3 in a simulation study.

3. Inference

In a full Bayesian approach, one is interested in the joint posterior distribution of all variables given the data (scalar covariates and the images). The posterior distribution is proportional to the likelihood of the data times the prior distributions of the parameters.[13]

3.1. Gaussian Response

Since $\varepsilon_i \overset{iid}{\sim} N(0, \sigma_\varepsilon^2)$, the likelihood has the form of a multivariate Gaussian distribution. The joint posterior distribution is therefore given by

$$\begin{aligned}
\pi(\alpha, \beta, \kappa, \sigma_\varepsilon^2 | y) &\propto \mathcal{L}(y|\alpha, \beta, \kappa, \sigma_\varepsilon^2) \times \pi(\beta|\kappa) \times \pi(\kappa) \times \pi(\sigma_\varepsilon^2) \\
&= (\sigma_\varepsilon^2)^{-N/2} \exp\left(-\frac{1}{2\sigma_\varepsilon^2}(y - W\alpha - X\beta)^T(y - W\alpha - X\beta)\right) \\
&\quad \times \kappa^{\mathrm{rk}(R)/2} \exp\left(-\frac{1}{2}\beta^T Q \beta\right) \\
&\quad \times \kappa^{a_\kappa - 1} \exp(-b_\kappa \kappa) \\
&\quad \times (\sigma_\varepsilon^2)^{-a_\varepsilon - 1} \exp\left(-\frac{b_\varepsilon}{\sigma_\varepsilon^2}\right).
\end{aligned} \tag{3.1}$$

The posterior has a rather complex form. Therefore it is due to the absence of a normalizing constant necessary, to use iterative methods to draw samples from the posterior distribution. For a Gaussian response, the full conditionals for all variables can be derived in a closed form. The prior of α is assumed to be constant (hence α is a fixed effect). The full conditional of α is also Gaussian, i.e.

$$\alpha|\cdot \sim N(\tilde{\mu}_\alpha, \widetilde{\Sigma}_\alpha)$$

with

$$\tilde{\mu}_\alpha = (W^T W)^{-1} W^T (y - X\beta) \qquad \widetilde{\Sigma}_\alpha = \sigma_\varepsilon^2 (W^T W)^{-1}.$$

The full conditional for β is given by

$$\beta|\cdot \sim N(\tilde{\mu}_\beta, \widetilde{Q}^{-1})$$

with

$$\tilde{\mu}_\beta = \left(\frac{1}{\sigma_\varepsilon^2} X^T X + Q\right)^{-1} \frac{1}{\sigma_\varepsilon^2} X^T (y - W\alpha) \qquad \widetilde{Q} = \left(\frac{1}{\sigma_\varepsilon^2} X^T X + Q\right).$$

[13] Since the prior of α is assumed to be constant, there is no need for a specification of its density

Despite the fact that Q has no full rank, now $\widetilde{Q}$ has full rank since it is the sum of Q and $(X^TX)/\sigma_\varepsilon^2$. In practice, there could still be a problem concerning the numerical stability due to the imprecision of computing machines (if the data does not provide enough information).
The full conditional for κ is:

$$\kappa|\cdot \sim Ga(\tilde{a}_\kappa, \tilde{b}_\kappa)$$

with

$$\tilde{a}_\kappa = \text{rk(R)}/2 + a_\kappa \qquad \tilde{b}_\kappa = -\frac{1}{2}\beta^T R\beta + b_\kappa$$

and for σ_ε^2:

$$\sigma_\varepsilon^2|\cdot \sim IG(\tilde{a}_\varepsilon, \tilde{b}_\varepsilon)$$

with

$$\tilde{a}_\varepsilon = N/2 + a_\epsilon \qquad \tilde{b}_\varepsilon = (y - W\alpha - X\beta)^T(y - W\alpha - X\beta)/2 + b_\varepsilon.$$

The derivation of all full conditionals can be found in the appendix A.
All full conditionals are available in a closed form and are well-known distributions. Therefore a Gibbs sampler can be used to draw samples from the posterior distribution (Brooks et al. [2011]). The Gibbs sampler was originally formulated in Geman and Geman [1984], where they described the connection between Markov random fields (which was introduced for the Gaussian case in this thesis in 2.3) and the Gibbs distribution which is used in mechanical systems. The Gibbs sampler is an easy algorithm where random numbers are drawn from the constituent full conditionals in an iterative procedure (Robert and Casella [2010]).
Given an arbitrary set of random variables $\vartheta = (\vartheta_1, \ldots, \vartheta_K)$, where all full conditionals $\pi(\vartheta_1|\cdot), \ldots, \pi(\vartheta_K|\cdot)$ are known, i.e. it is possible to generate random numbers from them, a pseudo algorithm for the Gibbs sampler is stated in Algorithm 1.
The result of the iteratively drawn random numbers is a Markov chain that converges to a stationary distribution.[14] After discarding a sufficient high number M of iterations (Burn-In), the generated ϑ's can be seen as realizations of the posterior distribution.

[14] A Markov chain converges to a stationary distribution under certain circumstances which are not discussed here. A good source concerning this topic is Meyn and Tweedie [2012]. However, for the here presented methods the convergence is ensured by construction, see Chib and Greenberg [1995] for an explanation of the more general MH-Algorithm, which will be used in the next section.

Initialize all variables $\vartheta^{(0)}$

for t = *1:* `#iterations` **do**

  for k = *1:K* **do**

    Generate $\vartheta_k^{(t+1)} \sim \pi(\vartheta_k|\vartheta_1^{(t+1)}, \dots \vartheta_{k-1}^{(t+1)}, \vartheta_{k+1}^{(t)}, \dots, \vartheta_K^{(t)}))$

  end

end

Discard the first M Iterations as Burn-In.

Algorithm 1: Gibbs sampler

3.2. Non-Gaussian Response

In the previous section, and also the regression problem (2.1) stated in section 2.2, assume a Gaussian response and therefore $y \in \mathbb{R}$. In this section, the Scalar-on-Image regression problem will be adapted to a general response, for exponential families in particular. For this, it is assumed that y_i follows a distribution of the exponential family (Nelder and Wedderburn [1972], Fahrmeir et al. [2007]) with density

$$\pi(y_i|\theta_i) = \exp\left(\frac{y\theta - b(\theta)}{\phi}\omega + c(y, \phi, \omega)\right)$$

where ω is a weight factor which is assumed to be one and can therefore be ignored. ϕ is the dispersion parameter which equals in the case of a Binomial and Poisson distribution one and in the case of a Gaussian distribution σ^2. The primary focus lies on θ. The goal is to model μ_i using a linear predictor $\eta_i = \sum_{j=1}^{p} w_{i,j}\alpha_j + \sum_{l=1}^{L} x_{i,l}\beta_l$ as in (2.1) via a response function $\mu_i = h(\eta_i)$. The full likelihood for all observations is given by

$$\mathcal{L}(y_i|\theta_i) = \prod_{i=1}^{N} \pi(y_i|\theta_i).$$

The likelihood in the formulation of the posterior (3.1) now has no Gaussian form anymore (except one assumes a Normal distributed response). Hence, the posterior distribution in (3.1) is again rather complex.

But as in the Gaussian case, it is possible to use iterative methods to draw random numbers from the posterior distribution. The updates for the precision of the IGMRF κ in Algortihm 1 stay the same. For the regression coefficients, they get a quiet complex form since their shape depends strongly on the likelihood. Therefore now for *both*, α and β, GMRFs as priors are assumed. Since the prior for α has to be non-informative

the corresponding GMRF has precision $Q_\alpha = \kappa_\alpha I_p$ where $\kappa_\alpha \to 0$. This is equivalent to assume a constant prior. The set of edges $\mathcal{E}$ of the corresponding graph associated with Q_α is empty. In the remaining section, α as well as β will be represented through the random variable γ. This allows an easier notation.
The update for the regressors in Algortihm 1 must be done now by a Metropolis-Hastings (MH) step. The Metropolis algorithm was originally proposed by Metropolis et al. [1953] and later generalized by Hastings [1970]. It was first heavily used by chemists and physicists and not widely known among statisticians until 1990 (Brooks et al. [2011]). To compute updates one needs to specify a proposal distribution $\varphi(\tilde{\vartheta}^{(t+1)}; \vartheta^{(t)}|\cdot)$ to generate a candidate $\tilde{\vartheta}^{(t+1)}$ given the old state $\vartheta^{(t)}$ (conditioned on all other variables) for a new point of the Markov chain. The general MH-Algorithm is given in Algorithm 2:

Specify proposal distribution $\varphi(\cdot;\cdot)$

Initialize all variables $\vartheta^{(0)}$

for *t = 1: `#iterations`* **do**

Generate a candidate $\tilde{\vartheta}^{(t+1)} \sim \varphi(\tilde{\vartheta}^{(t+1)}; \vartheta^{(t)}|\cdot)$

Calculate acceptance rate $\psi(\tilde{\vartheta}^{(t+1)}, \vartheta^{(t)})$:

$$\psi(\tilde{\vartheta}^{(t+1)}, \vartheta^{(t)}) = \min\left[1, \frac{\pi(\tilde{\vartheta}^{(t+1)})\varphi(\tilde{\vartheta}^{(t+1)}; \vartheta^{(t)})}{\pi(\vartheta^{(t)})\varphi(\vartheta^{(t)}; \tilde{\vartheta}^{(t+1)})}\right]$$

Accept $\tilde{\vartheta}^{(t+1)}$ as new state $\vartheta^{(t+1)}$ with probability ψ

end

Discard the first M Iterations as Burn-In.

Algorithm 2: Metropolis-Hastings-Algorithm

$\pi(\vartheta)$ is the kernel of the desired distribution. Since the normalization constant of this distribution cancels out in the denominator and numerator, it is not needed to be known. The acceptance rate ψ ensures that the Markov chain converges to a stationary distribution where the states can be seen as realizations of $\pi(\vartheta)$ after discarding the first M iterations as Burn-In.
Looking at the general posterior as stated in the first line of (3.1) one can see that for a general regressor γ (which is either α or β) the full conditional is given by

$$\pi(\gamma|\cdot) \propto \mathcal{L}(y|\gamma, \phi) \times \pi(\gamma|\kappa_\gamma) \tag{3.2}$$

where $\pi(\gamma|\kappa_\gamma)$ is a GMRF as described above.

The likelihood and the GMRF can only in a Gaussian case be combined to a Normal distribution (see section 3.1). But following Rue [2001] and Rue and Held [2005], one can approximate the likelihood with a 'Gaussian-like' function. This makes it possible to bring the likelihood and the GMRF together and get an appropriate proposal for a MH-update.
The approximation can be done by a quadratic Taylor expansion of the log-likelihood $l(\gamma) = \sum_{i=1}^{N} \ln\{\pi(y_i|\gamma, \phi)\}$ around the current state $\gamma^{(t)}$. The full conditional of the likelihood and the GMRF can be written as

$$\pi(\gamma|\cdot) \propto \exp\left(l(\gamma) - \frac{1}{2}\gamma^T Q\gamma\right). \tag{3.3}$$

Now the approximation of the likelihood around the current state $\gamma^{(t)}$ is

$$l(\gamma) \approx a^{(t)} + (b^{(t)})^T\gamma - \frac{1}{2}\gamma^T C^{(t)}\gamma \tag{3.4}$$

where

$$\begin{aligned} a^{(t)} &= l(\gamma^{(t)}) - (\gamma^{(t)})^T \frac{\partial l(\gamma^{(t)})}{\partial\gamma} + \frac{1}{2}(\gamma^{(t)})^T \frac{\partial^2 l(\gamma^{(t)})}{\partial\gamma^T\gamma}\gamma^{(t)} \\ b^{(t)} &= \frac{\partial l(\gamma^{(t)})}{\partial\gamma} + C^{(t)}\gamma^{(t)} \\ C^{(t)} &= -\frac{\partial^2 l(\gamma^{(t)})}{\partial\gamma^T\gamma}. \end{aligned}$$

Since the first part of (3.4) ($a^{(t)}$) does not depend on γ (only on the current state $\gamma^{(t)}$), it can be neglected. Inserting (3.4) in (3.3), the approximated full conditional is

$$\begin{aligned} \pi(\gamma|\cdot) &\approx \exp\left(a^{(t)} + (b^{(t)})^T\gamma - \frac{1}{2}\gamma^T C^{(t)}\gamma - \frac{1}{2}\gamma^T Q\gamma\right) \\ &\propto \exp\left(-\frac{1}{2}\gamma^T(Q + C^{(t)})\gamma + (b^{(t)})^T\gamma\right). \end{aligned}$$

This is a GMRF in its canonical representation as defined in Rue and Held [2005], p.27, i.e. the proposal $\tilde{\gamma}$ is again a Gaussian:

$$\tilde{\gamma}|\cdot \sim N(\tilde{\mu}^{(t)}, (\widetilde{Q}^{(t)})^{-1}) \tag{3.5}$$

where the mean $\tilde{\mu}^{(t)}$ is given by solving the linear system

$$\widetilde{Q}^{(t)}\tilde{\mu}^{(t)} = b^{(t)}$$

and the precision matrix $\widetilde{Q}^{(t)} = Q + C^{(t)}$. Using (3.5) as proposal distribution $\varphi(\gamma^{(t)}; \cdot)$, a new point $\tilde{\gamma}^{(t+1)}$ of the Markov chain can be drawn. This proposed point will be accepted as new state $\gamma^{(t+1)}$ with probability

$$\psi(\gamma^{(t)}, \tilde{\gamma}^{(t+1)}) = \min \left[1, \frac{\pi(\tilde{\gamma}^{(t+1)}|\cdot)\varphi(\tilde{\gamma}^{(t+1)}, \gamma^{(t)})}{\pi(\gamma^{(t)}|\cdot)\varphi(\gamma^{(t)}, \tilde{\gamma}^{(t+1)})} \right]$$

where $\pi(\gamma^{(t)}|\cdot)$ is the unnormalized full conditional of γ given in (3.2).
Note that if the likelihood originates from an exponential family, (3.5) is equivalent to an iterated weighted least squares (IWLS) proposal as described in Gamerman [1997] which is extensively used in this framework (Schmidt et al. [2017], Fahrmeir and Tutz [2013]).

4. Simulation Studies

The previously presented concepts were used in different simulation studies to investigate, how different types of parameters affect the results. The main focus in lies on two-dimensional images with a Normal response. Though, there will be a short investigation for a binary response and for three-dimensional images with a Gaussian response.

4.1. Software

The Scalar-on-Image regression problem can be seen as a specific case of STAR models (Fahrmeir et al. [2004]), as shortly described in the first section). Since one formulates spatial dependencies between regression coefficients using GMRFs, they coincide very well with them.
The `R` package `Sarim` (Kuester [2018]) provides functions and utilities for fitting STAR models. Therefore the analyses were carried out with the statistical software `R` (R Core Team [2018]). Since quiet high computational effort is needed to draw from the posterior distributions, the critical parts (i.e. the simulation of the Markov chains) use the `C++` extensions `Rcpp` with `RcppEigen` (Eddelbuettel [2013], Bates and Eddelbuettel [2013], Eddelbuettel and François [2011]).
To provide a unified model frame for Scalar-on-Image regression, the `R` package `SOIR` was written, which can be seen as an extension of the `Sarim` package to Scalar-on-Image regression. The `SOIR` package is provided via GitHub [15] and can be installed and loaded within a `R` session with

```
# install SOIR from github using the devtools package
devtools::install_github("RaphaelRe/SOIR")
library(SOIR)
```

The package contains the function `SOIR` which can be incorporated within the syntax of a call of the `sarim` function and is basically a wrapper of the function `sx` from the `Sarim` package. The call can be done like this:

```
sarim(y ~ SOIR(x))
```

where `x` are N vectorized images given in the description of the full function. The full function with all arguments:

[15] Full link to repository: www.github.com/RaphaelRe/SOIR

```
SOIR(images,
     dimension = rep(sqrt(ncol(images)),2),
     neighbours = c("2dfirst", "2dsecond", "2dallfirst",
                    "3dfirst", "3dsecond", "3dallfirst"),
     solver = c("rue", "lanczos"),
     demean = FALSE,
     add_diag = NULL,
     ...)
```

The constituent arguments are described as follows:

`images`
A $N \times L$ matrix, N images with L pixels or voxels in each row (in vectorized form). Therefore each column represents the l-th pixel or voxel over all images.

`dimension`:
One has to submit the exact dimensions of the (unvectorized) images (as a vector of the side length of the original image), since this information is needed to construct the structure matrix of the GMRF. Due to convenience, a default argument is implemented which assumes two-dimensional quadratic images (as the coefficient images which are later used in the simulation studies).

`neighbours`:
This argument defines the incorporated neighbours for the construction of the IGMRF. Currently, there are three different neighbourhoods implemented for two- and three-dimensional images. `"2dfirst"` uses the first four direct neighbours (see figure 2.2), `"2dsecond"` the eight direct first and second neighbours (see figure 2.3). `"2dallfirst"` incorporates all eight first direct and indirect neighbours (see figure 2.15). The function provides the same options for three-dimensional images, as it was described in section 2.4 (same commands where '2' get replaced by '3').

`solver`:
The solver which shall be used for sampling from the normal distribution (proposal density). The standard solver is `"rue"`, which uses a Cholesky decomposition of the precision matrix. See Rue and Held [2005], p. 34 for the algorithm to sample from a Gaussian distribution with a given covariance or precision matrix and p. 35 for sampling from a Gaussian distribution in canonical representation.
The other possible solver is `"lanczos"` which is a Krylov subspace method. These methods approximate the solution $A^{-1}b$ of a large linear system $Ax = b$ (Saad [2003]). This is done by projecting the problem to a lower dimensional space in a sequential

manner. See Simpson et al. [2013] or Schmidt et al. [2017] for a more accurate discussion. It can be reasonable to use this solver if the number of pixels or voxels is extremely high and only limited hardware resources are available.

`demean`:
An option to demean the images before fitting the model. This can be reasonable to set all images on the same colour level.

`add_diag`:
An option for adding a given number to the main diagonal of the structure matrix of the (latent) IGMRF which will be constructed by `SOIR()`. This can be done to obtain numerical stability within the sampling procedure when the data do not contain enough information. Moreover, this could lead to better results (see section 4.3.2). By adding a value to the main diagonal, the constructed IGMRF will get a normal GMRF where the precision matrix has full rank. This will shrink the coefficient image towards zero depending on the value of `add_diag`. The default is `NULL`, which means that no number will be added. For `add_diag` $\to \infty$, the coefficients will be shrunk to zero. Therefore it can be seen as a penalty term as described in section 1.

`...`(further arguments):
Further arguments which will be passed to the underlying `sx()` function from the `Sarim` package. In the context of Scalar-on-Image regression, this are mainly three further arguments: `ka_start`, `ka_a` and `ka_b`. The first argument is the starting value of the Markov chain, which shall be generated for the precision parameter κ. The latter two are the hyperparameters of the Gamma distribution, which is assumed as prior for κ, i.e. the shape a_κ and rate b_κ. The hyperparameters are rather important since the result depends heavily on them (see section 4.3.2).

The `SOIR` package also provides a view utilities which makes work a little easier. An example would be the `get_beta()` function which can easily be used to extract the coefficient image from the fitted model, discarding a Burn-In and reduce the chains of all coefficients (e.g. by its mean). This allows an easy workflow within the 'Forward-Pipe' from the `magrittr` package (Bache and Wickham [2014]). A simple example would be to fit a model and directly look at the estimated coefficient image (e.g. by using the function `plot_coefficient_image()`). For this, a Burn-In of 100 will be discarded, then the generated chains get reduced by its mean and then reshaped to the original dimension of the coefficient image (here 32×32) by using the function `set_dim()`.

```
sarim(y ~ SOIR(x), intercept = "TRUE") %>%
   get_beta(intercept = TRUE, burnin = 100, reduce = TRUE,
            reduce_with = mean) %>%
   set_dim(c(32,32) %>%
   plot_coefficient_image()
```

4.2. Data

4.2.1. Coefficient Images

For the simulation, four different coefficient images, depicted in figure 4.1, were considered. The first three are taken from Happ et al. [2018].[16] Each image reflects different structures in the coefficients.

- *Smooth*: A smooth image which is constructed from three two-dimensional Gaussian densities. This image reflects the smoothness assumption in the coefficients the most. Hence, it can be expected that GMRFs as priors should work well.
- *Sparse*: This image was originally proposed by Goldsmith et al. [2014]. The majority of the pixels are zero. There are only two small and smooth spikes. This reflects the assumption for sparse GMRFs which are not covered in detail in this thesis (see section 5 for a short description).
- *Bumpy*: This image was proposed by Reiss et al. [2015]. It is a two-dimensional version of the *bump* function (Donoho and Johnstone [1994]), which is a common benchmark for wavelet-based methods.[17]
- *Circle*: A simple circle with a clear edge at its margins. Therefore it can be used to examine the influence of different model settings in a clean manner. Pixels are either 0 or 0.1.

4.2.2. Covariates

The used covariates are N = 300 observations of 32×32 subimages from 300 three-dimensional images. The first of the full images can be found in Appendix B.

[16] The four images are also integrated into the `SOIR` package. The code of the first three is originally provided as supplemental material of Happ et al. [2018].

[17] Note that the original function was created for $64 \times 64 = 4096$ pixels. To obtain an image with $32 \times 32 = 1024$ pixels, the resolution was scaled down.

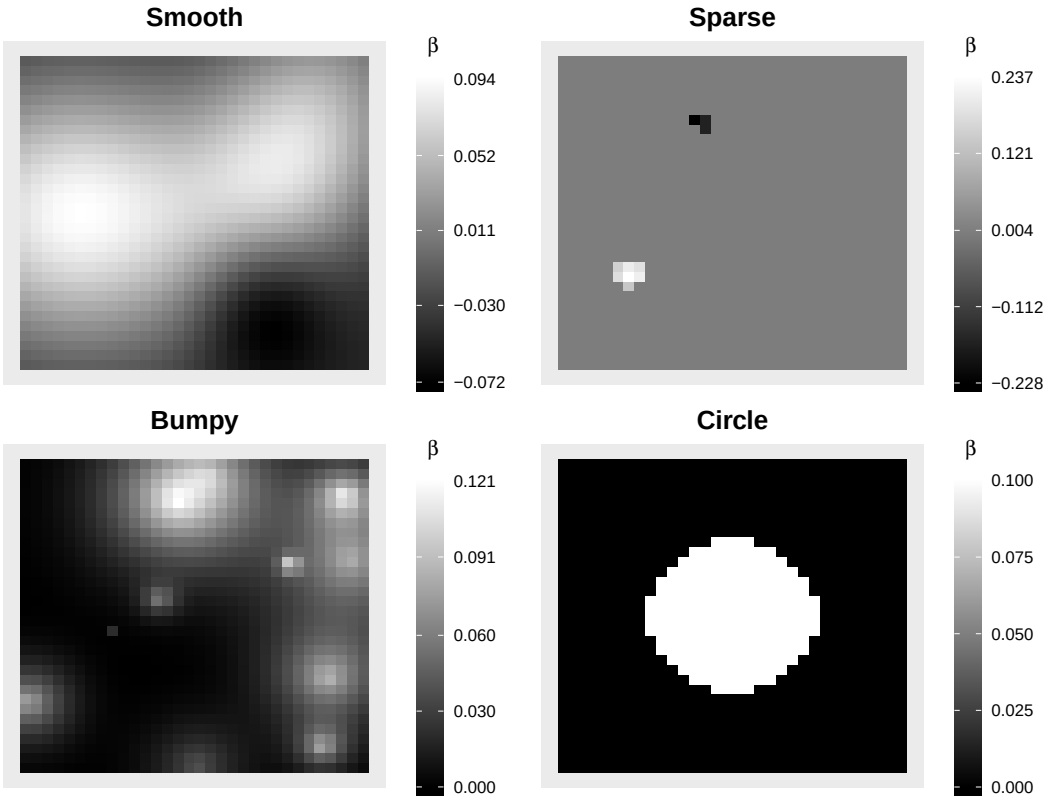

Fig. 4.1 The four coefficient images used in the simulations as β coefficients. (Note that each image has its own scale)

The image covariates stem from FDG-PET scans, which measure the glucose uptake in the brain. The original scans were co-registered to simultaneously measured MRI scans in order to reduce registration effects (Caballero et al. [2015]).
To obtain two-dimensional images, the center of the 20th slice of the three-dimensional images was extracted since it contains the most non-`NA` values. Three of the used images are depicted in figure 4.2. The `NA` values (located at the corners) were set to the lowest global value overall used images (which was roughly -0.6013). All images were demeaned before using them in the simulation (as in Happ et al. [2018]).

4.2.3. Construction of Response

As already mentioned, there were no further fixed effects α except an intercept and of course the covariate images. The relation is described in (2.3) with intercept $\alpha = -1$. For a Gaussian response, the added error terms ε_i were constructed with 5% noise on the original response, i.e.

$$\varepsilon_i \overset{iid}{\sim} N(0,\ \widehat{\mathrm{sd}}(X\beta) \times 0.05)$$

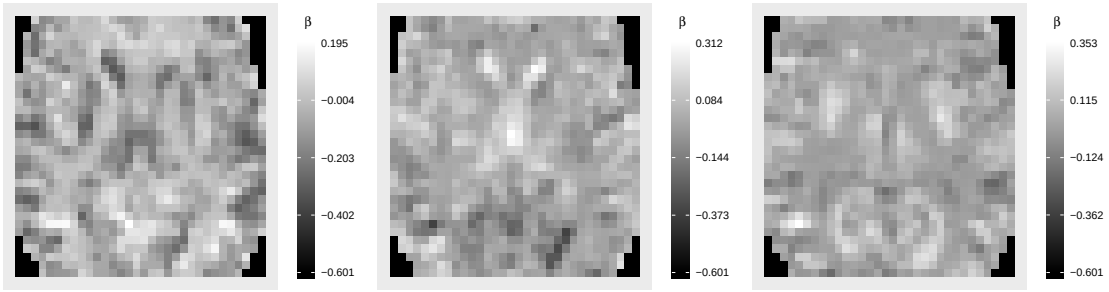

Fig. 4.2 Three of 300 used subimages with dimension 32×32. Note the different scale in each image. `NA` values (at the corners) were set to the lowest global value of -0.6013 (round to four decimal digits).

For a binomial response, (2.3) was used to to construct η_i (without ε_i). An individual probability p_i was afterwards calculated via

$$p_i = \frac{\exp(\eta_i)}{1 + \exp(\eta_i)}.$$

y_i was then constructed with threshold 0.5, therefore

$$y_i = \begin{cases} 1 & p_i > 0.5 \\ 0 & p_i \leq 0.5 \end{cases}$$

4.3. Studies

4.3.1. General Settings

All Markov chains run with 5000 iterations (except for the three-dimensional images). A point estimation was done by calculating the mean over the generated Markov chain. The first 500 iterations were always discarded as Burn-In. For a Normal response, the hyperparameters (shape and rate of the Inverse-gamma distribution) for the variance σ^2, were chosen to $a_\varepsilon = b_\varepsilon = 10^{-4}$ with a starting value of $(\sigma^2)^{(0)} = 0.1$. Each study was carried out with 28 replications.

4.3.2. Influence of Hyperparameters

In a first simulation study, it was examined how important a proper choice of the hyperparameters a_κ and b_κ are. For this, two different parameter settings were tested:

- $a_\kappa = b_\kappa = 1$, which is considered as rather uninformative (Happ et al. [2018])
- $a_\kappa = 10,\ b_\kappa = 10^{-3}$, highly informative

For both settings, two different values of `add_diag` were used:

- `add_diag` $= 10^{-4}$, which has no real influence and is used to obtain numerical stability.
- `add_diag` $= 10^{-1}$, pushes the the coefficient towards zero which leads to a lower variance in the posterior (see results).

Hence there are four models for four coefficient images. This leads to 16 models in total.[18]
To evaluate the quality of the estimated coefficient images, the mean squared error (MSE) between the true and the estimated image was calculated.

Adaption to coefficient images with many zero values:
Since all coefficient images except *Smooth*, contain a relatively high proportion of (nearly) zero values, the MSE was not calculated over all pixels, but by using a set $\widetilde{\mathrm{L}}$. Therefore the used MSE formula is

$$MSE = \frac{1}{|\widetilde{\mathrm{L}}|} \sum_{l=1}^{L} \mathbb{1}\{l \in \widetilde{\mathrm{L}}\}(\widehat{\beta}_l - \beta_l)^2.$$

$|\widetilde{\mathrm{L}}|$ is the cardinality of the set $\widetilde{\mathrm{L}}$. $\widehat{\beta}_l$ is the point estimation (mean over all iterations without Burn-In) of the simulated Markov chain for the l-th coefficient. β_l is the true coefficient and $\mathbb{1}\{l \in \widetilde{\mathrm{L}}\}$ an indicator function which is one, if l is in the set $\widetilde{\mathrm{L}}$ and zero otherwise.
$\widetilde{\mathrm{L}}$ was constructed as the union of all non-zero coefficients of the true coefficient image and all coefficients, were the 95%-credible interval of the simulated posterior distribution contains the zero. The union of both, the estimated and the true non-zeros is necessary since the model fails to identify all non-zero values. For an illustration see figure 4.3, where the estimated coefficient image (left) and the true (right) are depicted for only one replication. The non-zero values, which were identified by the model are coloured in red and superimposed over the estimated and true coefficient image. It can be seen in the true image, that *not* all non-zero pixels get identified. Another important observation is, that the estimated image fails to estimate the values to a substantial extent (note the high difference in the scale of the 2 images). See the following full results of the study for further details. For the depicted estimation, the four first direct neighbours were used. The hyperparameters were set to $a_\kappa = 10$ and $b_\kappa = 10^{-3}$. Furthermore, a

[18] Note that many other parameter settings were tested using the `R` package *mlrMBO* (Bischl et al. [2017]), which provides a modular framework for model-based optimization of expensive black-box functions. It was used to examine how reasonable the stated hyperparameters are. As surrogate model, a Gaussian process was used with a Matérn covariance function. The results gave no more insight and will not be discussed any further in this thesis.

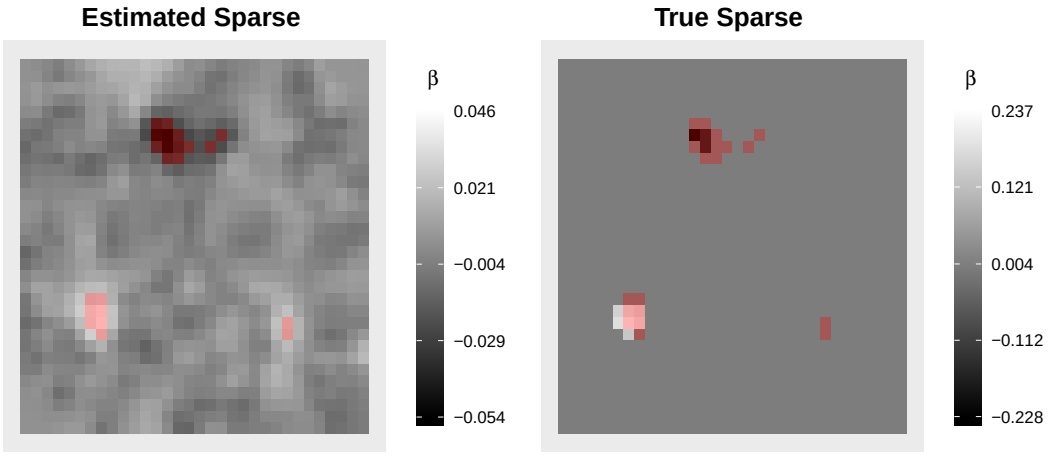

Fig. 4.3 The estimated (with four neighbours, $a_\kappa = 10, b_\kappa = 10^{-3}$, `add_diag` $= 0.1$) and true *Sparse* coefficient image (right). Depicted in red are the non zero values (95%-credible interval)

value of 0.1 was added to the main diagonal of the structure matrix of the underlying GMRF.

Results:
The estimated coefficient images are given in Appendix C. All model settings provide no useful coefficients for the images *Sparse* and *Bumpy*. Even though the models are able to find a sort of right structures of the true coefficient images, they completely fail to estimate their magnitude (note the scale of the estimated images in comparison with the original). Since smoothness over the coefficients is imposed, only smooth patterns can be estimated accurately. High steep peaks are getting smoothed out. The assumption also leads to a blurring of sharp edges as it can be observed in the estimated images for *Circle*.
For a comparison of the parameter settings, one can look at the resulting MSEs. The mean over all resulting MSEs of the four models, for the four coefficient images, are given in table 4.1. Each row shows the results for a specific parameter combination over all considered coefficient images. It can be seen, that the highest MSEs are given for $a_\kappa = b_\kappa = 1$ and `add_diag` $= 10^{-4}$. Therefore, it seems meaningful to choose a highly informative prior. By comparing the MSEs of the different values of `add_diag` by equal a_κ and b_κ, it can be seen that adding an additional value to the diagonal seems to lead to better results. For a better understanding concerning the variance, the MSEs over all 28 replications are given as boxplots in figure 4.4.[19]

[19]For a better visualization, the upper limit of the y-axis was set to 0.5. For the first parameter configuration ($a_\kappa = b_\kappa = 1$, `add_diag` $= 10^{-4}$), the highest outliers had a value around 1.5.

Parameters			Coefficient image			
a_κ	b_κ	`add_diag`	*Smooth*	*Sparse*	*Bumpy*	*Circle*
1	1	10^{-4}	0.2586	0.2172	0.1689	0.2644
1	1	10^{-1}	0.0089	0.0287	0.0080	0.0092
10	10^{-3}	10^{-4}	0.0106	0.0260	0.0059	0.0125
10	10^{-3}	10^{-1}	**0.0017**	**0.0101**	**0.0012**	**0.0025**

Tab. 4.1 Mean of 28 replications of the calculated MSEs over all models with Normal response (round to four decimal digits). The minimal value of each column is printed in bold.

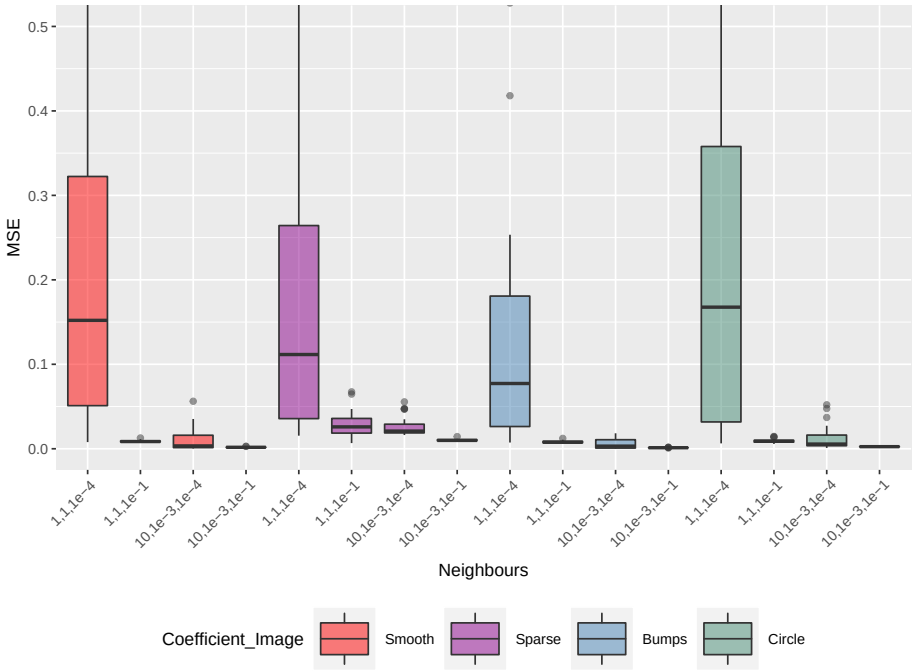

Fig. 4.4 MSEs for different parameter configurations as given in table 4.1. Note that the y-scale was limited to obtain a better comparison between the different hyperparameter configurations.

Adding an additional value to the main diagonal leads to a much lower variance (or higher precision) of the GMRF (see the moments of a GMRF, (2.5) in section 2). This also affects the calculated credible intervals. For low values of `add_diag`, the intervals are much greater than for higher values. A major consequence of this is, that the calculated intervals contain much more often the zero. For an illustration see figure 4.5. Depicted are two single (only one replication) generated Markov chains for β_l, $l = 328$ [20] of the coefficient image *Sparse* for both values of `add_diag` (10^{-4} and 10^{-1}) with $a_\kappa = 10, b_\kappa = 10^{-3}$. The red lines are the 2.5% and 97.5% quantiles and therefore gives the 95%-credible interval. The blue line indicates the zero. It can be seen that for `add_diag` $= 10^{-4}$, the variance of the chain is much higher and the credible interval

[20] $l = 328$ was chosen, since it has the highest mean of all generated chains.

includes the zero. For the second case, the variance is much smaller and the zero is not included in the credible interval.[21]

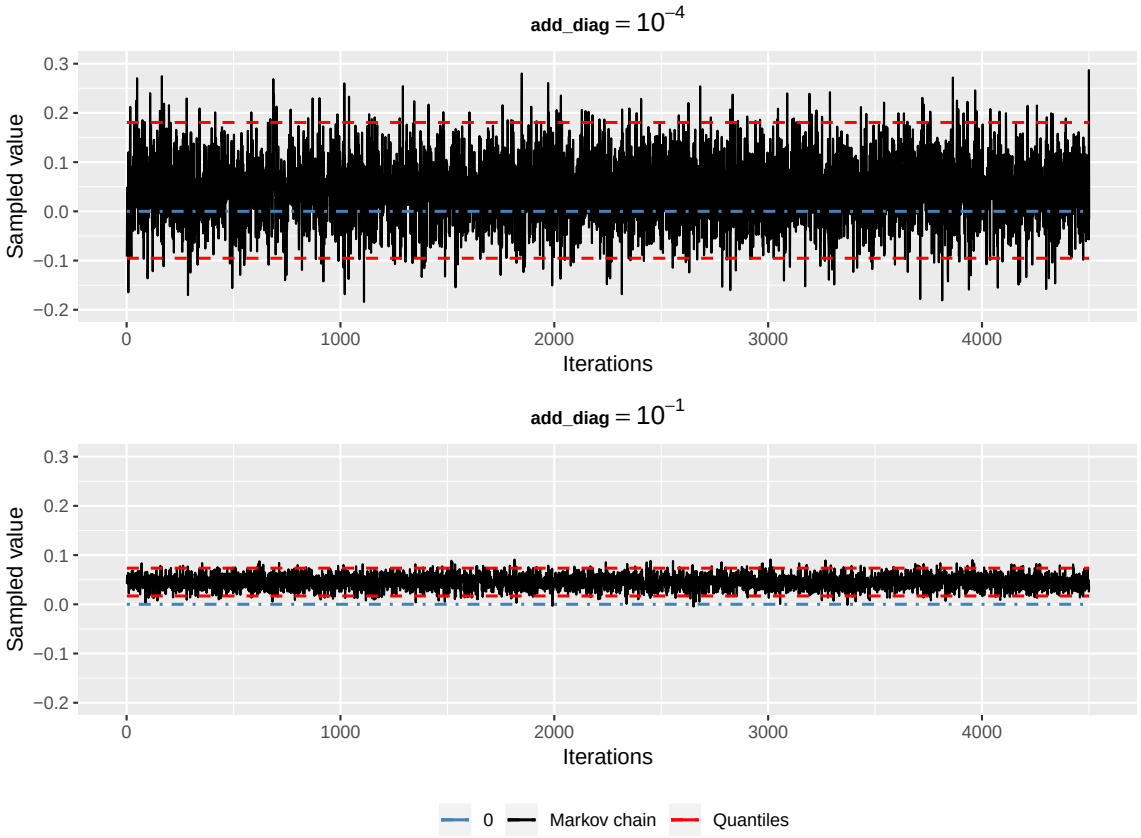

Fig. 4.5 Generated Markov chains of β_l, $l = 328$ for $a_\kappa = 10, b_\kappa = 10^{-3}$ and the two different values of `add_diag`(10^{-4} and 10^{-1}). The red lines indicate the 2.5% and 97.5% quantiles and the blue line the zero.

By looking at all credible intervals for all coefficient images and all parameter configurations, it was observable that the first three models were not able to identify any non-zero values (see table 4.2, which shows the sum of all credible intervals that excludes the zero). Only the model with $a_\kappa = 10, b_\kappa = 10^{-3}$ and `add_diag` $= 10^{-4}$ was able to identify non-zero values.

4.3.3. Influence of incorporated Neighbours

In this study, the influence of the incorporated neighbours of the GMRF was examined. For this only the images *Smooth* and *Circle* were used. *Smooth* was chosen since it suits the smoothness assumption very well. *Circle* was chosen to examine how the incorporated neighbourhood influences sharp edges in images. For both coefficient

[21] Note that depicted chain for `add_diag` $= 10^{-1}$ corresponds to the 328-th pixel of figure 4.3.

Parameters			Number of non-zero coefficients			
a_κ	b_κ	`add_diag`	*Smooth*	*Sparse*	*Bumpy*	*Circle*
1	1	10^{-4}	0	0	0	0
1	1	10^{-1}	0	0	0	0
10	10^{-3}	10^{-4}	0	0	0	0
10	10^{-3}	10^{-1}	160	21	110	126

Tab. 4.2 Number of non-zero values using a 95%−credible interval. Note that the counts stem from only one replication.

images the hyperparameters were set to $a_\kappa = 10,\ b_\kappa = 10^{-3}$ and `add_diag` $= 10^{-1}$. The evaluation was again done by using the MSE. The results in form of the mean over 28 replications are given in table 4.3.

`Neighbours`	MSEs	
	Smooth	*Circle*
`"2dfirst"`	0.0017	0.0025
`"2dsecond"`	**0.0015**	**0.0019**
`"2dallfirst"`	0.0035	0.0042

Tab. 4.3 Mean of 28 replications of the calculated MSEs over all models with Normal response for incorporated neighbours (round to four decimal digits). The minimal value for each coefficient image is printed in bold.

For a better understanding, the MSEs of the 28 replications are depicted again as boxplots in figure 4.6.
It turns out, that for the here examined coefficient images, incorporating eight neighbours (first direct and indirect) leads to much higher MSEs. In Addition, it seems like the MSEs tend to be smaller if not only the first, but the first *and* second direct neighbours are taken into account. To get more insight, how the neighbourhood affects the coefficient images, see figure 4.7, which shows their estimations for all neighbourhoods. Each pixel was calculated as the mean over all point estimates of all 28 replications.
The image shows, that there is only a very slight visual difference between the two direct neighbourhoods. It can also be seen, that the third option leads to a rather bad result which coincides very well with the results of the calculated MSEs. The bad results are plausible since the structure matrix has a much lower rank as in the other considered neighbourhoods.

4.3.4. Lanczos Solver

This simulation was carried out to test, if the second possible solver `"lanczos"` delivers equal results. This was only done for the coefficient image *smooth* and *circle*. The

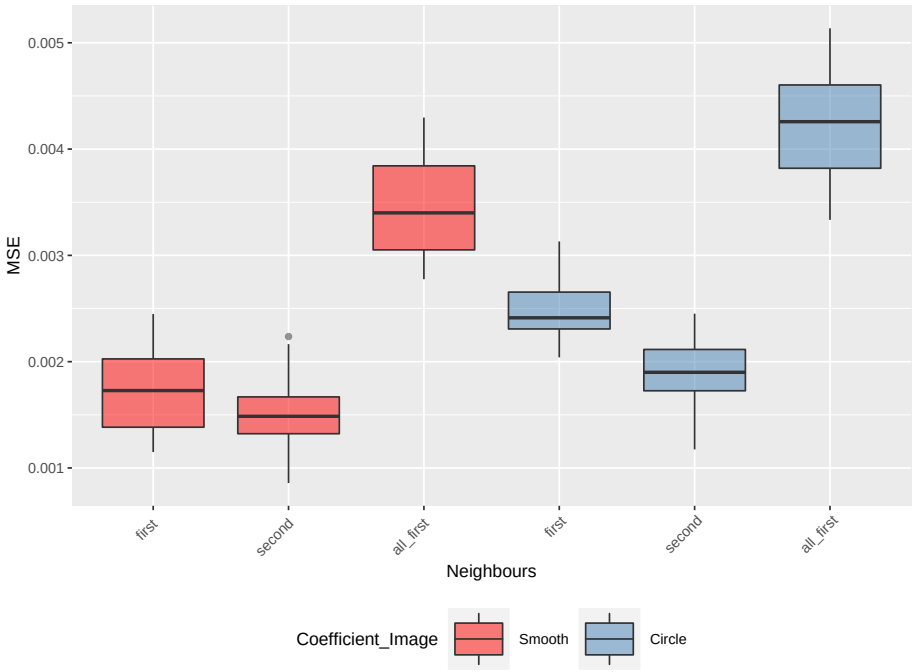

Fig. 4.6 MSEs for the three different types of incorporated neighbours for 28 replications.

convergence threshold for the solver was set to 10^{-4}. The experiment was set up like for the influence of the hyperparameters, but only with $a_\kappa = 10$, $b_\kappa = 10^{-3}$ and `add_diag` = 0.1. The simulation was again repeated 28 times. The resulting MSEs are given in table 4.4 (round on 4 decimal digits). For both coefficient images (*Smooth* and *Circle*), the mean MSE over 28 replications was nearly the same. Round on four decimal digits, the results are actually equal.

For a visual comparison between the two solvers see figure 4.8. Depicted are the estimated coefficient images *Smooth* (left) and *Circle* (right). The first row shows the results for `"lanczos"`, the second for `"rue"`. It can be seen that there is only a very small difference between the two solvers.

`solver`	MSEs	
	Smooth	*Circle*
`"rue"`	0.0017	0.0027.
`"lanczos"`	0.0017	0.0027

Tab. 4.4 Different MSEs for *Smooth* and *Circle* estimated with solvers `"rue"` and `"lanczos"` for comparison. `"lanczos"` was calculated with a convergence threshold of 10^{-4}.

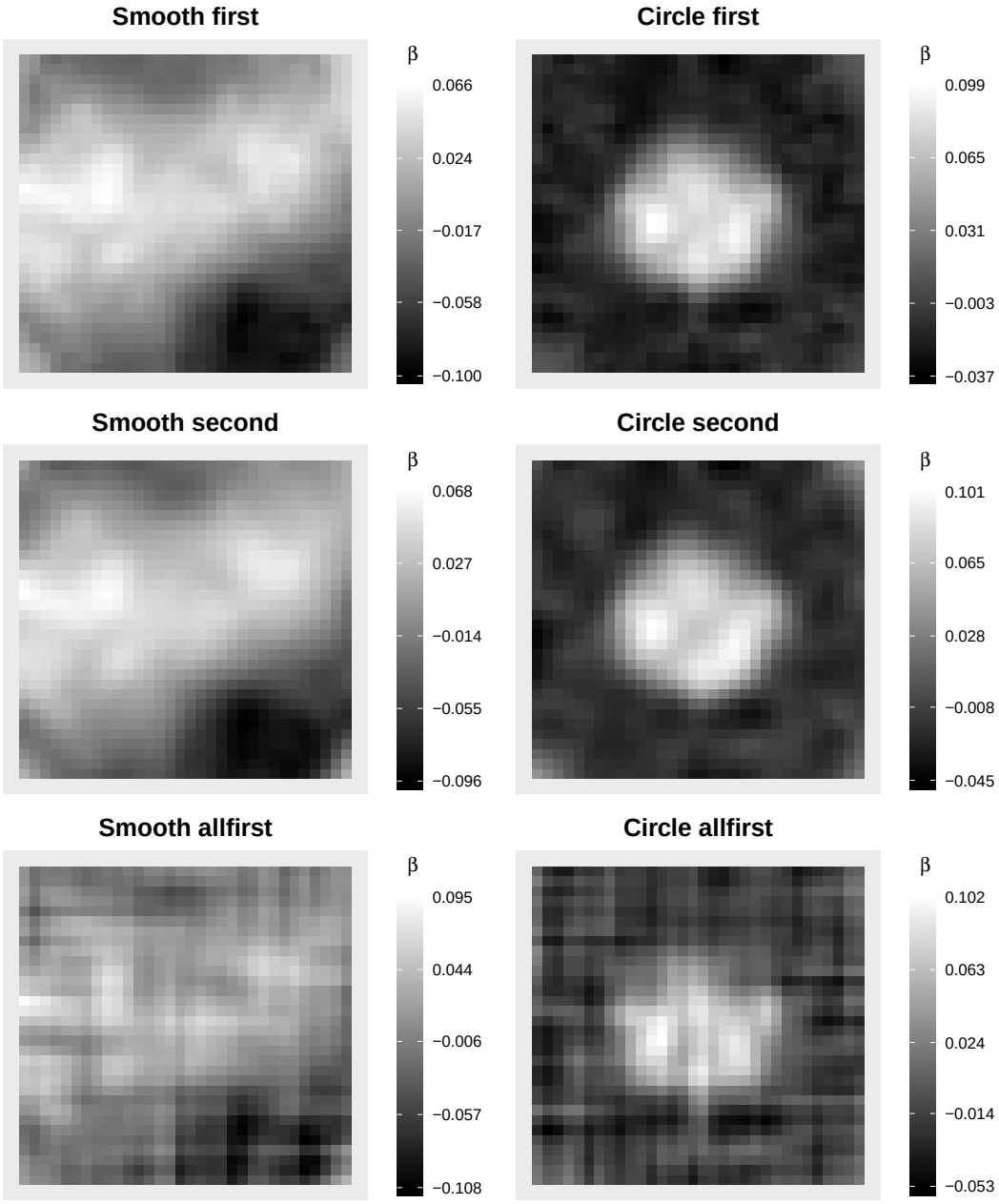

Fig. 4.7 Influence on the estimated Coefficient images for *Smooth* (left) and *Circle* (right) using three different incorporated neighbourhoods as described in section 4.1

4.3.5. Binary Target Variable

An additional short simulation study for a binary y was conducted to examine, whether and how well iterative methods are working (again for 28 replications).
The construction of the response is described in section 4.2.3. The simulation was only done for the coefficient image *Smooth* with the parameter configuration $a_\kappa = 10$, $b_\kappa = 10^{-3}$ and two different values of `add_diag`. The estimated coefficient images are presented in figure 4.9, left for `add_diag` $= 10^{-4}$ and right for `add_diag` $= 10^{-1}$.

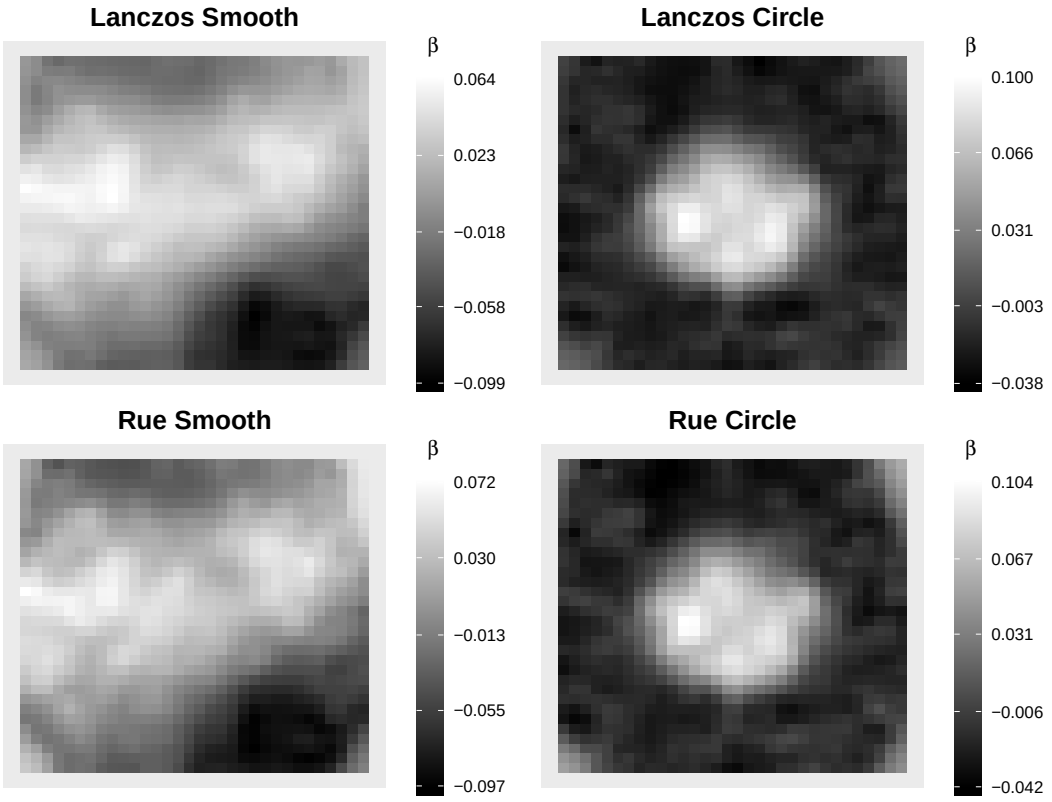

Fig. 4.8 Comparison lanczos (first row) and rue (second row) for the coefficient image *Smooth* (left) and *Circle* (right), calculated over 28 replications.

Each pixel is again calculated as the mean over the point estimates of 28 replications. The results for both are rather bad: By looking at the scale, it can be seen that the estimated coefficients take rather extreme values (in the positive direction as well as in the negative). Nevertheless, it can also be seen that some structures in the images were found, especially for a lower value of `add_diag`. Despite the found structure, the mean MSEs over 28 replications were 1.3469 for $\texttt{add\_diag} = 10^{-4}$ and 0.2623 for $\texttt{add\_diag} = 10^{-4}$ (which are both much higher than for a Gaussian response). Since the estimated coefficient image for $\texttt{add\_diag} = 10^{-4}$ is more smooth, the regions with extreme values are greater. See e.g. the bottom right in both images. This leads to a much higher MSE for the lower value of `add_diag`.

An inspection of the realized Markov chains had shown, that a lower value of `add_diag` leads again to a much lower variance. Therefore the difference in the MSEs should not be overrated. Moreover, the mean acceptance rates for the MH-updates were only 50.5 % for the $\texttt{add\_diag} = 10^{-4}$ and 45.2% for $\texttt{add\_diag} = 10^{-1}$.

Therefore it can be concluded, that the estimation for a binary target variable delivers no satisfactory results in this simulation. A deeper analysis for a binary y is needed but will be not covered in this thesis.

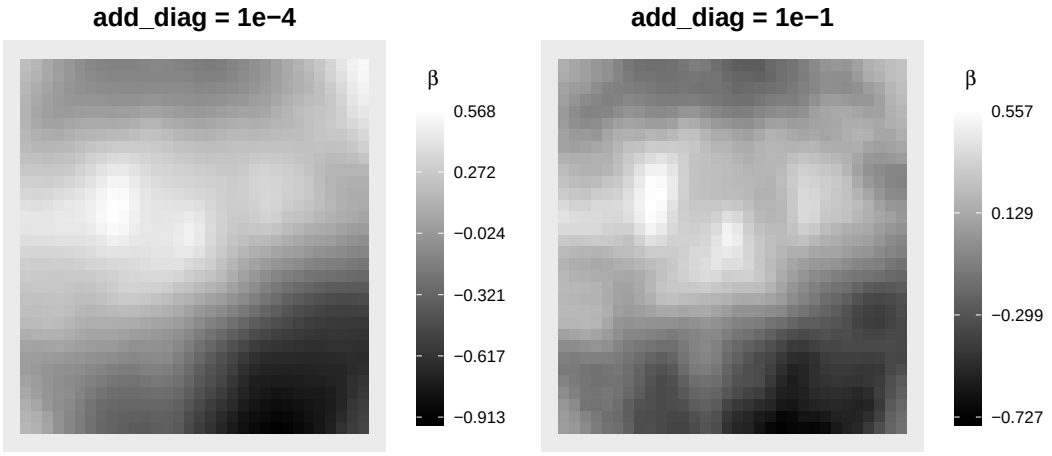

Fig. 4.9 Estimated coefficient image (over 28 replications) for a binary target variable. The left image shows the estimation with `add_diag` $= 10^{-4}$, right with `add_diag` $= 10^{-1}$.

4.3.6. Three-dimensional Images

The last simulation study was done for a three-dimensional coefficient image. As in the first section described, the number of coefficients grows in three dimensions much faster. This also leads to a higher number of coefficients even though the side lengths $n_k,\ k = 1, 2, 3$ of the used images were sharply reduced. The image was constructed from the coefficient image *Circle* and forms a cylinder: The total dimension is $16 \times 16 \times 10$. The slices three to eight are two-dimensional *Circles* with dimension 16×16. Figure 4.10 shows the original coefficient image.[22] The used covariates are three-dimensional subimages from the center of the same original covariates as used before.
Since the number of coefficients is more than double as high as for the used two-dimensional image, the calculated chains run only over 3000 iterations. The simulation was again repeated 28 times. The hyperparameters for the prior Gamma distribution were set informatively to $a_\kappa = 10, b_\kappa = 10^{-3}$ and an additional value for `add_diag` was again set, to 10^{-1} to get a lower variance in the estimation.
Figure 4.11 shows the estimated coefficient image.
The estimated image shows relatively similar results as for its two-dimensional analogue. The circles in the center of the image get recognized and estimated correctly. But the smoothness assumption leads to a blurring of the edges.
The mean over 28 MSEs was 0.0036 (round to 4 decimal digits).

[22] Note that the range of the colour scale was aligned to the scale of the estimated coefficient image to make both images comparable.

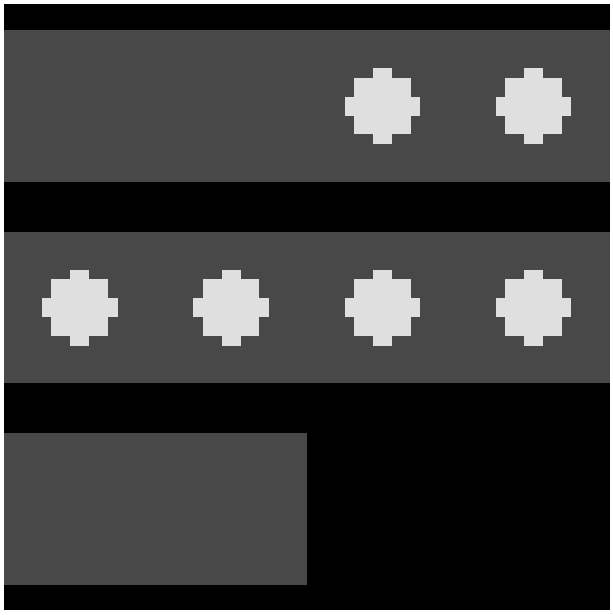

Fig. 4.10 Original three-dimensional coefficient image with dimension $16 \times 16 \times 10$ depicted in 10 slices.

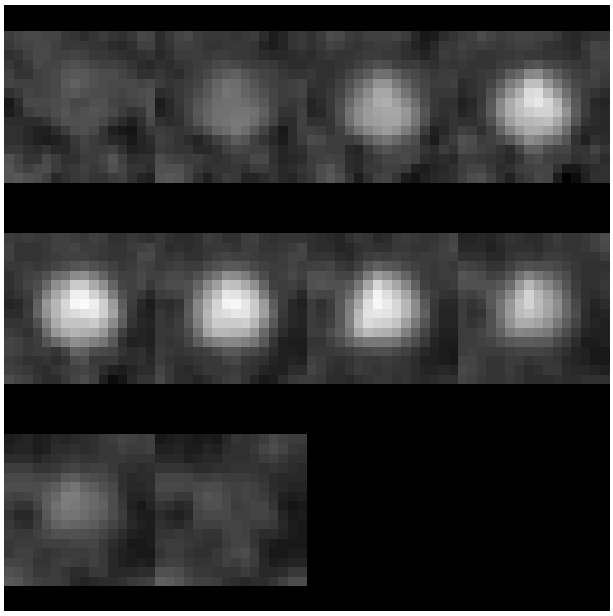

Fig. 4.11 Estimated three-dimensional coefficient image with dimension $16 \times 16 \times 10$ depicted in 10 slices.

5. Other Models

This thesis described and used different settings of GMRFs within a full Bayesian approach for Scalar-on-Image regression. Besides this, other approaches exist, which can be considered.

The smoothness assumption formulated through a GMRF is maybe not always present in the true coefficient image (or only to a rather small magnitude, see e.g. the coefficient image *Sparse*). To overcome this problem within a Bayesian framework, one can use an additional type of prior which imposes sparsity (as it is present in the coefficient *Sparse*). Goldsmith et al. [2014] add this type of prior to get a variable selection aspect to the GMRF. This is supposed to combine the smoothness and sparsity assumption. The idea of this is, that major parts of the true coefficient image are not associated with the target variable y and are therefore assumed to be zero. Therefore they use a latent binary indicator image that designates coefficients as either predictive or nonpredictive. This is done by a latent Ising prior in a hierarchical Bayesian formulation.
A coefficient β_l is then either zero or follows a GMRF. If an Ising prior is used within the hierarchical Bayes, one has to deal with two additional parameters coming from the Ising model. These two parameters control the overall sparsity of the coefficient image. Therefore the results depend heavily on them. The estimation for these two parameters can be done with a cross-validation approach, which consequently leads to a much higher computational burden in comparison to a standard GMRF (see Happ et al. [2018]).

Functional approaches
Apart from iterative methods, other approaches for Scalar-on-Image regression exist. According to Happ et al. [2018], these methods can be concluded as *basis function approaches*. The idea behind these methods is, that the coefficient image is generated by a function $\beta(\cdot) : \mathcal{T} \to \mathbb{R}$ with $\mathcal{T} \subset \mathbb{R}^d$. $\beta(\cdot)$ is evaluated at the points t_l of a finite lattice. Therefore $\beta_l = \beta(t_l)$ represents the values of the unknown coefficient image. It is assumed that β lies in the span of K known basis functions $B_1, \dots B_K$, which is a K-dimensional space. The original regression problem (2.1) can therefore represented in terms of these basis functions (the fixed effects remain unchanged):

$$y_i = \sum_{j=1}^{p} w_{i,j}\alpha_j + \sum_{l=1}^{L} x_{i,l} \sum_{k=1}^{K} b_k B_k(t_l) + \varepsilon_i. \tag{5.1}$$

Now, the number of coefficients, denoted with $b_k, \quad k = 1, \dots K$, which shall be estimated is K. Usual K is much smaller than L. As long as $p + K < N$, the system of equations is identifiable. If N is too high, it is possible, to impose additional assumptions for the

coefficients b_k as described before.
If one uses orthonormal basis functions, it can be useful to interpret the observed images also as functions and to expand them in the same basis functions as $\beta(\cdot)$ with coefficients $\xi_{i,m}$. Then the regression problem (2.1) can be represented as:

$$\begin{aligned} y_i &= \sum_{j=1}^{p} w_{i,j}\alpha_j + \sum_{l=1}^{L} x_{i,l}\beta_l + \varepsilon_i \\ &\approx \sum_{j=1}^{p} w_{i,j}\alpha_j + \sum_{m=1}^{K}\sum_{k=1}^{K} \xi_{i,m} b_k \sum_{l=1}^{L} B_m(t_l)B_k(t_l) + \varepsilon_i \\ &\approx \sum_{j=1}^{p} w_{i,j}\alpha_j + \sum_{m=1}^{K}\sum_{k=1}^{K} \xi_{i,m} b_k \int_{\mathcal{T}} B_m(t)B_k(t)\mathrm{d}t + \varepsilon_i \\ &= w_i^T\alpha + \xi^T b + \varepsilon_i \end{aligned}$$

where $\xi = (\xi_{i,1}, \ldots, \xi_{i,K})$ and $b = (b_1, \ldots b_K)$. Note that for a non-equidistant lattice, integration wights would be needed to be valid.
In general, there are different types of basis functions which can be used for the regression problem. An overview of some examples (taken from Happ et al. [2018]):

- (Penalized) B-Splines (see Marx and Eilers [2005])
- Wavelets (see e.g. Reiss et al. [2015])
- Principal component regression (see e.g. Allen [2013])
- Combination of the above variants. (See e.g. Reiss and Ogden [2010] for principal component regression based on splines or Reiss et al. [2015] for principal component regression and partial least squares in wavelet space.)

Each of these basis function approaches has its own assumptions concerning the unknown coefficient image which leads to different results. For a comparison of all methods mentioned above, see Happ et al. [2018] where a simulation study for the different methods is conducted.

6. Summary and Discussion

This thesis introduced Scalar-on-Image regression by using iterative methods, i.e. Markov Chain Monte Carlo and examined different aspects in various simulation studies.

Firstly, Scalar-on-Image regression and Gaussian Markov random fields (GMRFs) were introduced. A GMRF is a Normal distribution, which can be used to model dependencies between adjacent locations. Therefore it seems like a reasonable prior for Scalar-on-Image regression to overcome the $p >> n$ problem. Furthermore, it exploits the spacial structure in images, which could lead to better results – as long as the smoothness assumption for the underlying coefficient image holds.
Afterwards, it was described how inference in this context can be done. In a full Bayesian approach, one has to use iterative methods which are also known as Markov Chain Monte Carlo. For a Gaussian response, all full conditionals are known and a Gibbs sampler can be used to draw from the posterior distribution. For a non-Gaussian response, one has to use a Metropolis-Hastings update within the generated Markov chain for the regression coefficients.

Following the presented concepts, different simulation studies were carried out. The first study examined the influence of chosen hyperparmeters. All parameter configurations were able to find underlying structures in the images. But due to the smoothness assumption, the models fail to estimate accurate values for high peaks and blur sharp edges. An informative prior led to better estimations for all underlying coefficient images. This seems plausible: Since the number of coefficients was much higher than the number of observations, one has to subjoin a lot of prior information, to obtain satisfactory results. Furthermore, it was tested if it is useful to add an additional value to the main diagonal of the structure matrix. This leads to a lower variance in the generated Markov chains and the calculated MSEs. This has implications for the realized study: For a lower value, a higher number of replications that 28 could be appropriate. Adding an additional value also shrinks the coefficients towards zero. One has to be aware of these circumstances and must choose all parameters with caution.
In a second study, the influence of the incorporated neighbourhood was examined (for the coefficient images *Smooth* and *Circle*). The results revealed, that incorporating the first direct and indirect neighbours led to the worst results. The reason for this is probably the high loss in rank of the precision matrix. Incorporating the first and second direct neighbours yielded slightly better results than only using the first four. This could be plausible: By using eight direct neighbours, one can utilize the spacial structure a little more than by using only four, without suffering from a much higher

loss in rank.
For the same coefficient images, it was shown that a Lanczos approximation works very well and leads to nearly equal results. Therefore one can use this option if the number of coefficients is quite high and only limited hardware resources are available. However, the question whether this result is applicable to a general coefficient image remains.
An additional study for the coefficient image *Smooth* with a binary target variable was conducted. The resulting MSEs were rather bad. Nevertheless, the models were able to find structures in the images, where a lower value of `add_diag` led to more smoothness in the estimated image, but also to a higher MSE. Though, this also leads again to a higher variance in the generated chains and therefore in the reliability of the results. It would also be interesting how different settings affect the acceptance rate of the MH-update. Therefore much deeper investigations are needed to get a better understanding of this kind of problem.
A last study for three-dimensional images with a Gaussian response was done. As in the first section described, the number of coefficients grows very fast with the dimension. Therefore the study was only done for a smaller image (regarding the side length). The results are roughly similar to a two-dimensional image. The difference lies in the needed computing resources.
The simulation studies had shown, that in general an informative prior with the first and second direct neighbours seems to be meaningful. It can also be useful to add an additional value to the structure matrix to obtain more reliable results. Though a deeper analysis of non-Gaussian responses is needed.

The last section presented other methods for Scalar-on-Image regression. To overcome the enforced smoothness over the full coefficient image, one can extend the GMRF prior for the β coefficients with a further Ising prior, to additionally include a sparseness assumption. This leads to additional tuning parameters and therefore to a much higher computational effort.
Furthermore, other approaches exist. Here, one assumes that the coefficient image is generated by a latent function which maps from a higher dimensional space to a one-dimensional. The function is evaluated for each pixel or voxel at its location. Furthermore, it is assumed that this latent function can be represented by a set of basis functions. Using this basis functions, one can reduce the number of regressors until the problem gets a feasible solution. The outcome of the estimation strongly depends on the used basis functions. Each basis function imposes its own assumption for the underlying coefficient image.

This has, according to Happ et al. [2018], some important implications: If one imposes assumptions for a model, the found results are not only containing information from

the data, but also from the assumptions made in the model. Therefore if one assumes smoothness, as it is done in this thesis by using GMRFs, the result will be a smooth image. Therefore the found results will always reflect the imposed structures. Or as stated by Coombs [1964]: *"We buy information with assumptions"*.
This means for $p >> n$ problems, for Scalar-on-Image regression in particular, that one needs a deep understanding of the application field and the used methods. It is important to evaluate if a specific assumption is justifiable and inferred results are valid. It is also important to understand the impact of model settings as hyperparameters and choose them appropriately.
With respect to the presented simulation studies, this means that additional investigations would give more insight. All studies were carried out with a constant number of observations. It is questionable, how many observations are required and how this number interacts with the strength of the chosen prior information. Furthermore, it could be interesting if and how the used neighbourhood interacts with the resolution of images. Potential anisotropy in images is an additional point which, however, was not covered by this thesis.

Bibliography

Allen, Genevera I. (2013). "Multi-way functional principal components analysis". In: *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. IEEE, pp. 220–223.

Bache, Stefan Milton and Hadley Wickham (2014). *magrittr: A Forward-Pipe Operator for R*. R package version 1.5. URL: `https://CRAN.R-project.org/package=magrittr`.

Bates, Douglas, Dirk Eddelbuettel, et al. (2013). *Fast and elegant numerical linear algebra using the RcppEigen package*. In: Journal of Statistical Software 52.5, pp. 1–24.

Besag, Julian (1974). *Spatial interaction and the statistical analysis of lattice systems*. In: Journal of the Royal Statistical Society. Series B (Methodological), pp. 192–236.

Besag, Julian (1975). *Statistical analysis of non-lattice data*. In: The statistician, pp. 179–195.

Bischl, Bernd, Jakob Richter, Jakob Bossek, Daniel Horn, Janek Thomas, and Michel Lang (2017). *mlrMBO: A Modular Framework for Model-Based Optimization of Expensive Black-Box Functions*. In: arXiv. URL: `http://arxiv.org/abs/1703.03373`.

Bishop, Christopher M. (2006). *Pattern recognition and machine learning*. springer.

Brooks, Steve, Andrew Gelman, Galin Jones, and Xiao-Li Meng (2011). *Handbook of Markov Chain Monte Carlo*. CRC press.

Caballero, Miguel Ángel Araque, Matthias Brendel, Andreas Delker, Jinyi Ren, Axel Rominger, Peter Bartenstein, Martin Dichgans, Michael W. Weiner, Michael Ewers, et al. (2015). *Mapping 3-year changes in gray matter and metabolism in Aβ-positive nondemented subjects*. In: Neurobiology of aging 36.11, pp. 2913–2924.

Chakraborty, Sounak, Malay Ghosh, and Bani K. Mallick (2012). *Bayesian nonlinear regression for large p small n problems*. In: Journal of Multivariate Analysis 108, pp. 28–40.

Chib, Siddhartha and Edward Greenberg (1995). *Understanding the metropolis-hastings algorithm*. In: The american statistician 49.4, pp. 327–335.

Clayton, D. G. (1995). *Generalized linear mixed models*. In: *Markov chain Monte Carlo in practice*. Ed. by Walter R. Gilks, Sylvia Richardson, and David Spiegelhalter. London: Chapman and Hall/CRC, pp. 275–301.

Coombs, Clyde H. (1964). *A theory of data*. New York: Wiley.

Donoho, David L. and Jain M. Johnstone (1994). *Ideal spatial adaptation by wavelet shrinkage*. In: biometrika 81.3, pp. 425–455.

Eddelbuettel, Dirk (2013). *Seamless R and C++ Integration with Rcpp*. ISBN 978-1-4614-6867-7. New York: Springer. DOI: 10.1007/978-1-4614-6868-4.

Eddelbuettel, Dirk and Romain François (2011). *Rcpp: Seamless R and C++ Integration*. In: Journal of Statistical Software 40.8, pp. 1–18. DOI: 10.18637/jss.v040.i08. URL: http://www.jstatsoft.org/v40/i08/.

Fahrmeir, Ludwig, Thomas Kneib, et al. (2011). *Bayesian smoothing and regression for longitudinal, spatial and event history data*. In: OUP Catalogue.

Fahrmeir, Ludwig and Gerhard Tutz (2013). *Multivariate statistical modelling based on generalized linear models*. Springer Science & Business Media.

Fahrmeir, Ludwig, Thomas Kneib, and Stefan Lang (2004). *Penalized structured additive regression for space-time data: a Bayesian perspective*. In: Statistica Sinica, pp. 731–761.

Fahrmeir, Ludwig, Thomas Kneib, Stefan Lang, and Brian Marx (2007). *Regression*. Springer.

Frank, LLdiko E. and Jerome H Friedman (1993). *A statistical view of some chemometrics regression tools*. In: Technometrics 35.2, pp. 109–135.

Friedman, Jerome, Trevor Hastie, and Rob Tibshirani (2010). *Regularization paths for generalized linear models via coordinate descent*. In: Journal of statistical software 33.1, p. 1.

Fu, Wenjiang J. (1998). *Penalized regressions: the bridge versus the lasso*. In: Journal of computational and graphical statistics 7.3, pp. 397–416.

Gamerman, Dani (1997). *Sampling from the posterior distribution in generalized linear mixed models*. In: Statistics and Computing 7.1, pp. 57–68.

Geman, Stuart and Donald Geman (1984). *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*. In: IEEE Transactions on pattern analysis and machine intelligence 6, pp. 721–741.

Goldsmith, Jeff, Lei Huang, and Ciprian M. Crainiceanu (2014). *Smooth scalar-on-image regression via spatial Bayesian variable selection*. In: Journal of Computational and Graphical Statistics 23.1, pp. 46–64.

Gössl, Christoff, Dorothee P. Auer, and Ludwig Fahrmeir (2001). *Bayesian spatiotemporal inference in functional magnetic resonance imaging*. In: Biometrics 57.2, pp. 554–562.

Happ, Clara, Sonja Greven, and Volker J. Schmid (2018). *The impact of model assumptions in scalar-on-image regression*. In: Statistics in medicine 37.28, pp. 4298–4317.

Hastings, W. Keith (1970). *Monte Carlo sampling methods using Markov chains and their applications*. In: Biometrika 57.1, pp. 97–109.

Hoerl, Arthur E. and Robert W. Kennard (1970). *Ridge regression: Biased estimation for nonorthogonal problems*. In: Technometrics 12.1, pp. 55–67.

Kuester, Christopher (2018). *Sarim: Structured additive regression model using interative methods*. `https://github.com/bioimaginggroup/Sarim`.

Li, Qing and Nan Lin (2010). *The Bayesian elastic net*. In: Bayesian Analaysis 5.1, pp. 151–170.

Marx, Brian D. and Paul H.C. Eilers (2005). *Multidimensional penalized signal regression*. In: Technometrics 47.1, pp. 13–22.

Metropolis, Nicholas, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller (1953). *Equation of state calculations by fast computing machines*. In: The journal of chemical physics 21.6, pp. 1087–1092.

Meyn, Sean P. and Richard L. Tweedie (2012). *Markov chains and stochastic stability*. Springer Science & Business Media.

Nelder, John Ashworth and Robert W.M. Wedderburn (1972). *Generalized linear models*. In: Journal of the Royal Statistical Society: Series A (General) 135.3, pp. 370–384.

Penny, William D., Nelson J. Trujillo-Barreto, and Karl J. Friston (2005). *Bayesian fMRI time series analysis with spatial priors*. In: NeuroImage 24.2, pp. 350–362.

Penny, William D., Karl J. Friston, John T. Ashburner, Stefan J. Kiebel, and Thomas E. Nichols (2011). *Statistical parametric mapping: the analysis of functional brain images*. Elsevier.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: `https://www.R-project.org/`.

Reiss, Philip T. and R. Todd Ogden (2010). *Functional generalized linear models with images as predictors*. In: Biometrics 66.1, pp. 61–69.

Reiss, Philip T., Lan Huo, Yihong Zhao, Clare Kelly, and R. Todd Ogden (2015). *Wavelet-domain regression and predictive inference in psychiatric neuroimaging*. In: The annals of applied statistics 9.2, p. 1076.

Robert, C. and G. Casella (2010). *Introducing Monte Carlo Methods with R*. Use R! Springer. URL: `https://books.google.co.il/books?id=WIjMyiEiHCsC`.

Rue, Håvard (2001). *Fast sampling of Gaussian Markov random fields*. In: Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63.2, pp. 325–338.

Rue, Havard and Leonhard Held (2005). *Gaussian Markov random fields: theory and applications*. CRC press.

Saad, Yousef (2003). *Iterative methods for sparse linear systems*. Vol. 82. siam.

Schmidt, Paul, Mark Muehlau, and Volker Schmid (2017). *Fitting large-scale structured additive regression models using Krylov subspace methods*. In: Computational Statistics & Data Analysis 105, pp. 59–75.

Simpson, Daniel P., Ian W. Turner, Christopher M. Strickland, and Anthony N. Pettitt (2013). *Scalable iterative methods for sampling from massive Gaussian random vectors*. In: arXiv preprint arXiv:1312.1476.

Steeb, Willi-Hans and Tan Kiat Shi (1997). *Matrix calculus and Kronecker product with applications and C++ programs*. World Scientific.

Tibshirani, Robert (1996). *Regression shrinkage and selection via the lasso*. In: Journal of the Royal Statistical Society. Series B (Methodological), pp. 267–288.

Zou, Hui and Trevor Hastie (2005). *Regularization and variable selection via the elastic net*. In: Journal of the royal statistical society: series B (statistical methodology) 67.2, pp. 301–320.

Appendices

A. Derivation of the Full Conditionals for Gaussian Response

All full conditionals can be derived from the joint posterior (3.1).

Full conditional for $\boldsymbol{\alpha}$

Let $\tilde{y} = y - X\beta$:

$$\begin{aligned}
\alpha|\cdot &\propto \exp\left(-\frac{1}{2\sigma_\varepsilon^2}(y - W\alpha - X\beta)^T(y - W\alpha - X\beta)\right) \\
&= \exp\left(-\frac{1}{2\sigma_\varepsilon^2}(\tilde{y}^T\tilde{y} - 2\alpha^T W^T\tilde{y} + \alpha^T W^T W\alpha)\right) \\
&\propto \exp\left(-\frac{1}{2\sigma_\varepsilon^2}(\alpha^T W^T W\alpha - 2\alpha^T W^T\tilde{y}\right) \\
&\propto \exp\left(-\frac{1}{2}\left(\alpha^T\frac{1}{\sigma_\varepsilon^2}W^T W\alpha - 2\alpha^T\frac{1}{\sigma_\varepsilon^2}W^T W(W^T W)^{-1}W^T\tilde{y}\right)\right) \\
&= \exp\left(-\frac{1}{2}\left(\alpha^T\widetilde{\Sigma}_\alpha^{-1}\alpha - 2\alpha^T\widetilde{\Sigma}_\alpha^{-1}\tilde{\mu}_\alpha\right)\right) \\
&\propto \exp\left(-\frac{1}{2}(\alpha^T - \tilde{\mu}_\alpha)^T\widetilde{\Sigma}_\alpha^{-1}(\alpha^T - \tilde{\mu}_\alpha)\right)
\end{aligned}$$

This is the kernel of a Gaussian distribution with mean and variance

$$\tilde{\mu}_\alpha = (W^T W)^{-1}W^T\tilde{y}\ , \qquad \widetilde{\Sigma}_\alpha = \sigma_\varepsilon^2(W^T W)^{-1}$$

Full conditional for $\boldsymbol{\beta}$

Now let $\tilde{y} = y - W\alpha$:

$$\begin{aligned}
\beta|\cdot &\propto \exp\left(-\frac{1}{2\sigma_\varepsilon^2}(y - W\alpha - X\beta)^T(y - W\alpha - X\beta)\right) \times \exp\left(-\frac{1}{2}\beta^T Q\beta\right) \\
&= \exp\left(-\frac{1}{2\sigma_\varepsilon^2}\left(\tilde{y}^T\tilde{y} - 2\beta^T X^T\tilde{y} + \beta^T X^T X\beta\right) - \frac{1}{2}\beta^T Q\beta\right) \\
&\propto \exp\left(-\frac{1}{2}\left(\beta^T\frac{1}{\sigma_\varepsilon^2}X^T X\beta + \beta^T Q\beta - 2\beta^T\frac{1}{\sigma_\varepsilon^2}X^T\tilde{y}\right)\right)
\end{aligned}$$

$$
\begin{aligned}
&\propto \exp\left(-\frac{1}{2}\left(\beta^T\left(\frac{1}{\sigma_\varepsilon^2}X^TX+Q\right)\beta\right.\right.\\
&\qquad\left.\left.-2\beta^T\left(\frac{1}{\sigma_\varepsilon^2}X^TX+Q\right)\left(\frac{1}{\sigma_\varepsilon^2}X^TX+Q\right)^{-1}\frac{1}{\sigma_\varepsilon^2}X^T\tilde{y}\right)\right)\\
&= \exp\left(-\frac{1}{2}(\beta^T\widetilde{Q}\beta-2\beta^T\widetilde{Q}\tilde{\mu}_\beta)\right)\\
&\propto \exp\left(-\frac{1}{2}(\beta-\tilde{\mu}_\beta)^T\widetilde{Q}(\beta-\tilde{\mu}_\beta\right)
\end{aligned}
$$

This is the kernel of Gaussian distribution with mean and precision

$$
\tilde{\mu}_\beta=\left(\frac{1}{\sigma_\varepsilon^2}X^TX+Q\right)^{-1}\frac{1}{\sigma_\varepsilon^2}X^T\tilde{y}\ ,\qquad \widetilde{Q}=\left(\frac{1}{\sigma_\varepsilon^2}X^TX+Q\right)
$$

Full conditional for $\boldsymbol{\kappa}$

Remind that $Q=\kappa R$. Then

$$
\begin{aligned}
\kappa|\cdot &\propto \kappa^{\mathrm{rk}(R)/2}\exp\left(-\frac{1}{2}\kappa\beta^TR\beta\right)\times\kappa^{a_\kappa-1}\exp\left(-b_\kappa\kappa\right)\\
&= \kappa^{\mathrm{rk}(R)/2+a_\kappa-1}\exp\left(-\kappa\left(\frac{1}{2}\beta^TR\beta+b_\kappa\right)\right)\\
&= \kappa^{\tilde{a}_\kappa-1}\exp(-\kappa\tilde{b}_\kappa)
\end{aligned}
$$

This is the kernel of a Gamma distribution with shape and rate

$$
\tilde{a}_\kappa=\mathrm{rk(R)}/2+a_\kappa\ ,\qquad \tilde{b}_\kappa=\frac{1}{2}\beta^TR\beta+b_\kappa
$$

Full conditional for $\boldsymbol{\sigma_\varepsilon^2}$

The full conditional can also derived from the joint posterior:

$$
\begin{aligned}
\sigma_\varepsilon^2|\cdot &\propto (\sigma_\varepsilon^2)^{-N/2}\exp\left(-\frac{1}{2\sigma_\varepsilon^2}(y-W\alpha-X\beta)^T(y-W\alpha-X\beta)\right)\\
&\qquad\times(\sigma_\varepsilon^2)^{-a_\varepsilon-1}\exp\left(-\frac{b_\varepsilon}{\sigma_\varepsilon^2}\right)\\
&= (\sigma_\varepsilon^2)^{-(N/2+a_\varepsilon)-1}\exp\left(-\frac{1}{\sigma_\varepsilon^2}\left((y-W\alpha-X\beta)^T(y-W\alpha-X\beta)/2+b_\varepsilon\right)\right)
\end{aligned}
$$

which corresponds to the kernel of an Inverse-gamma distribution with parameters

$$
\tilde{a}_\varepsilon=N/2+a_\varepsilon\ ,\qquad \tilde{b}_\varepsilon=(y-W\alpha-X\beta)^T(y-W\alpha-X\beta)/2+b_\varepsilon
$$

B. Image of a three-dimensional Covariate

Figure B.1 shows the first of 300 three-dimensional images used for the simulations in 36 slices.

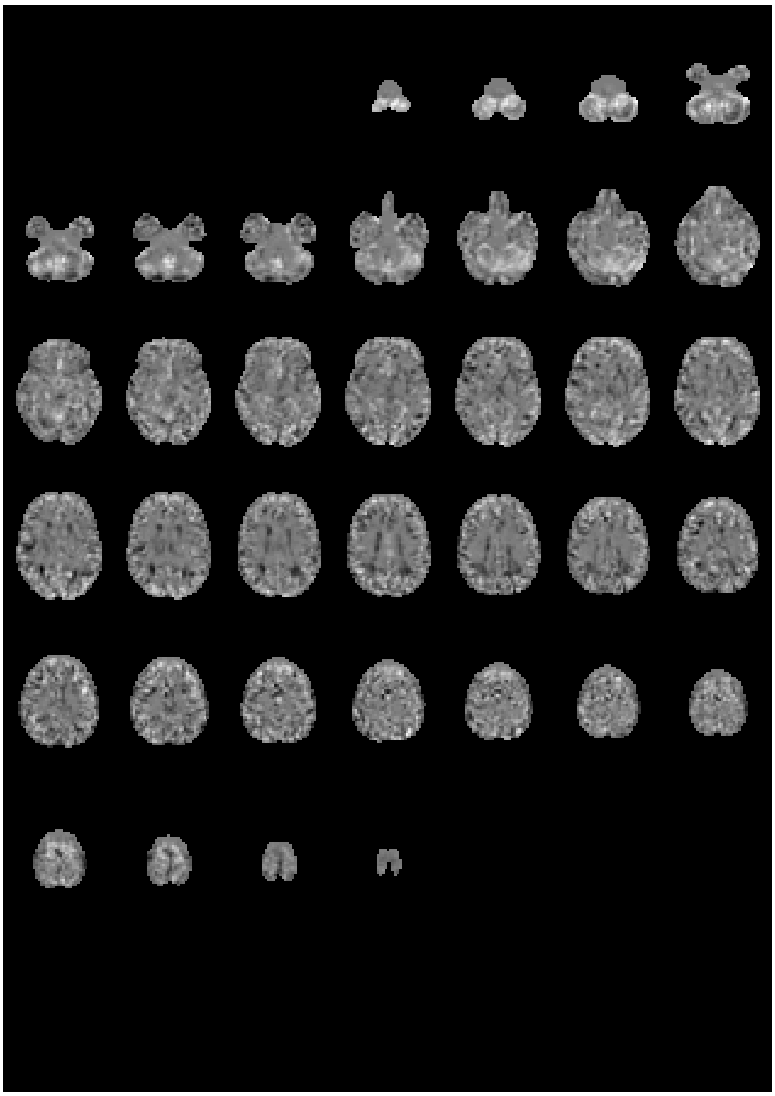

Fig. B.1 The first of 300 three-dimensional image used for simulations

C. Estimated Coefficient Images

The following graphics show the estimated coefficient images with different parameter settings.

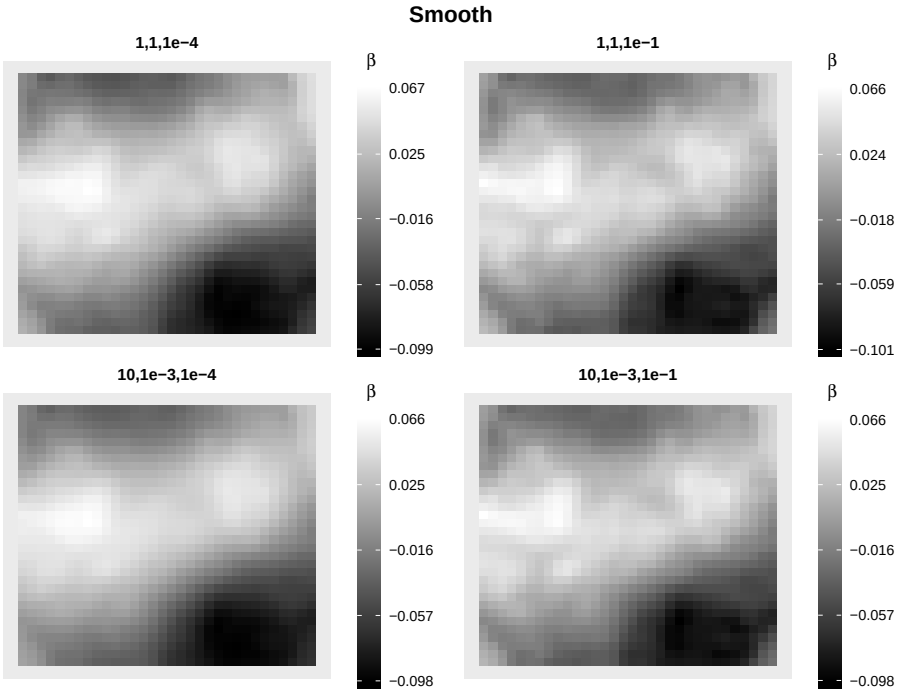

Fig. C.1 Estimated image *Smooth* (different parameter settings)

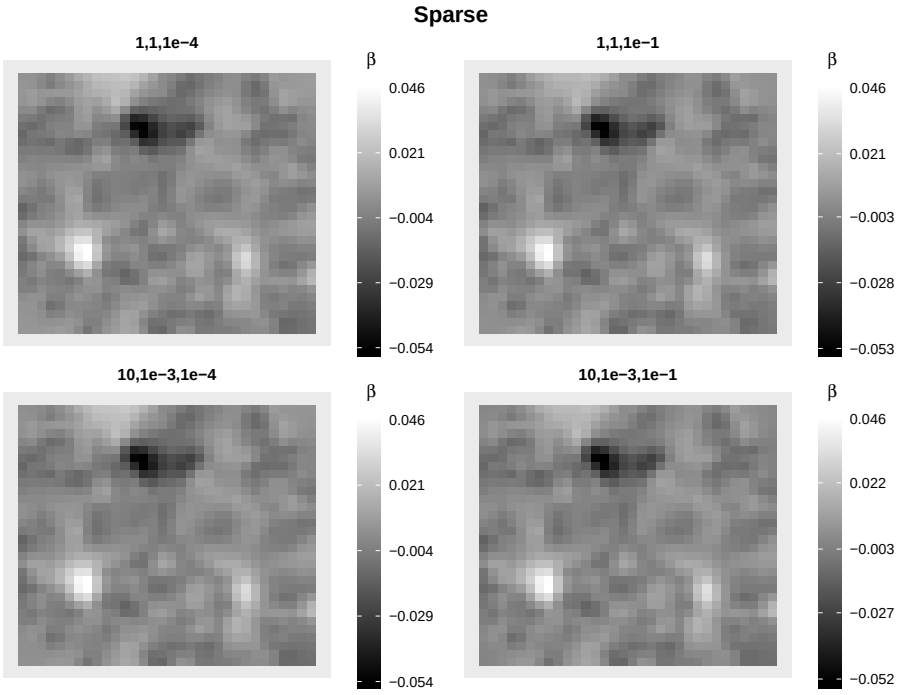

Fig. C.2 Estimated image *Sparse* (different parameter settings)

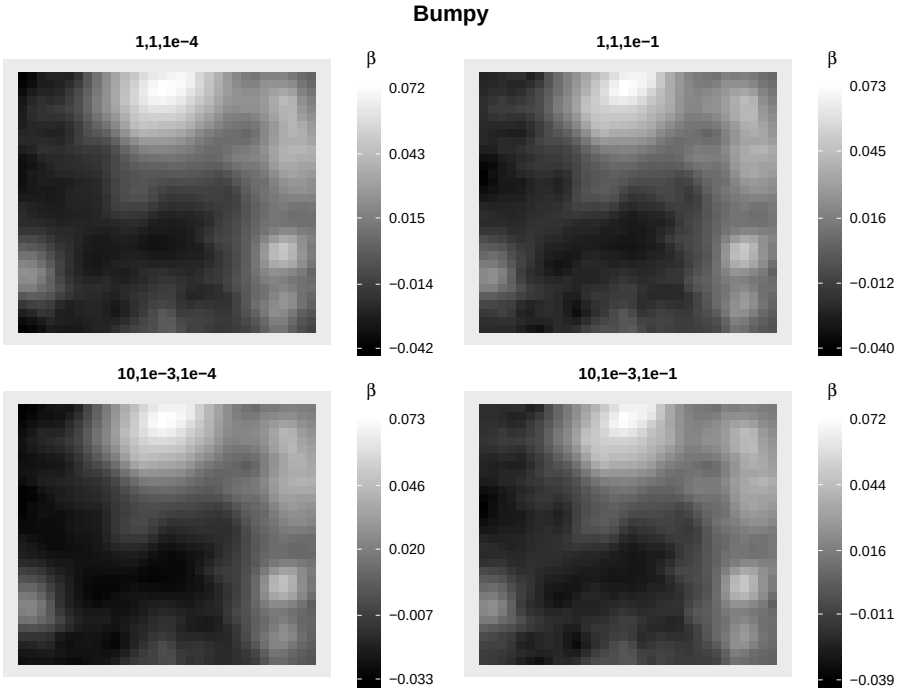

Fig. C.3 Estimated image *Bumpy* (different parameter settings)

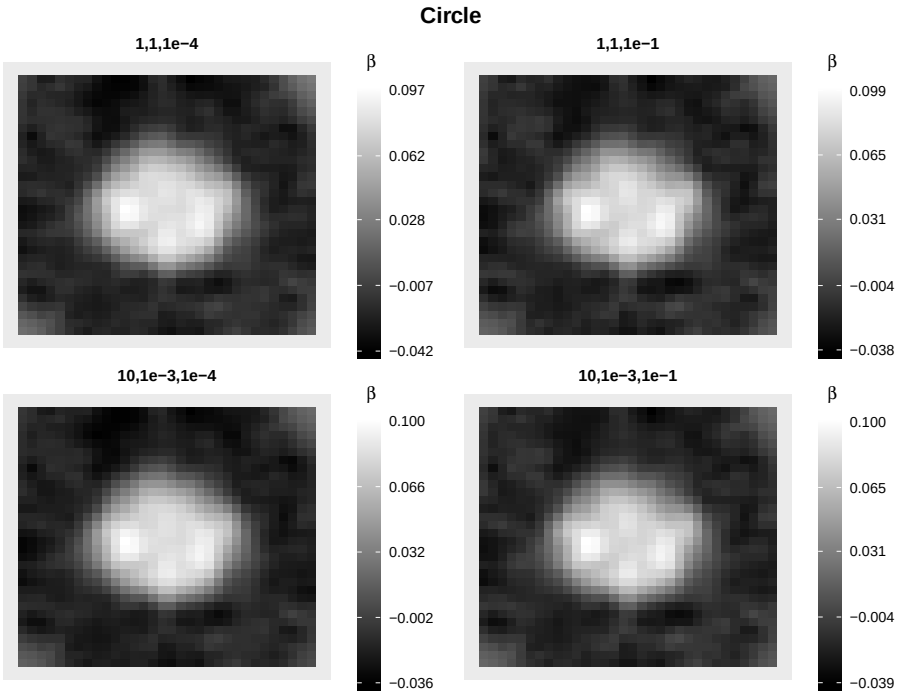

Fig. C.4 Estimated image *Circle* (different parameter settings)

Statutory Declaration

I declare that this thesis was carried out by my own for the degree Master of Science under the guidance and supervision of Prof. Dr. Volker Schmid, Department of Statistics, Ludwig Maximilian University of Munich. I have not used other than the declared sources and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Munich, March 18, 2019

_____