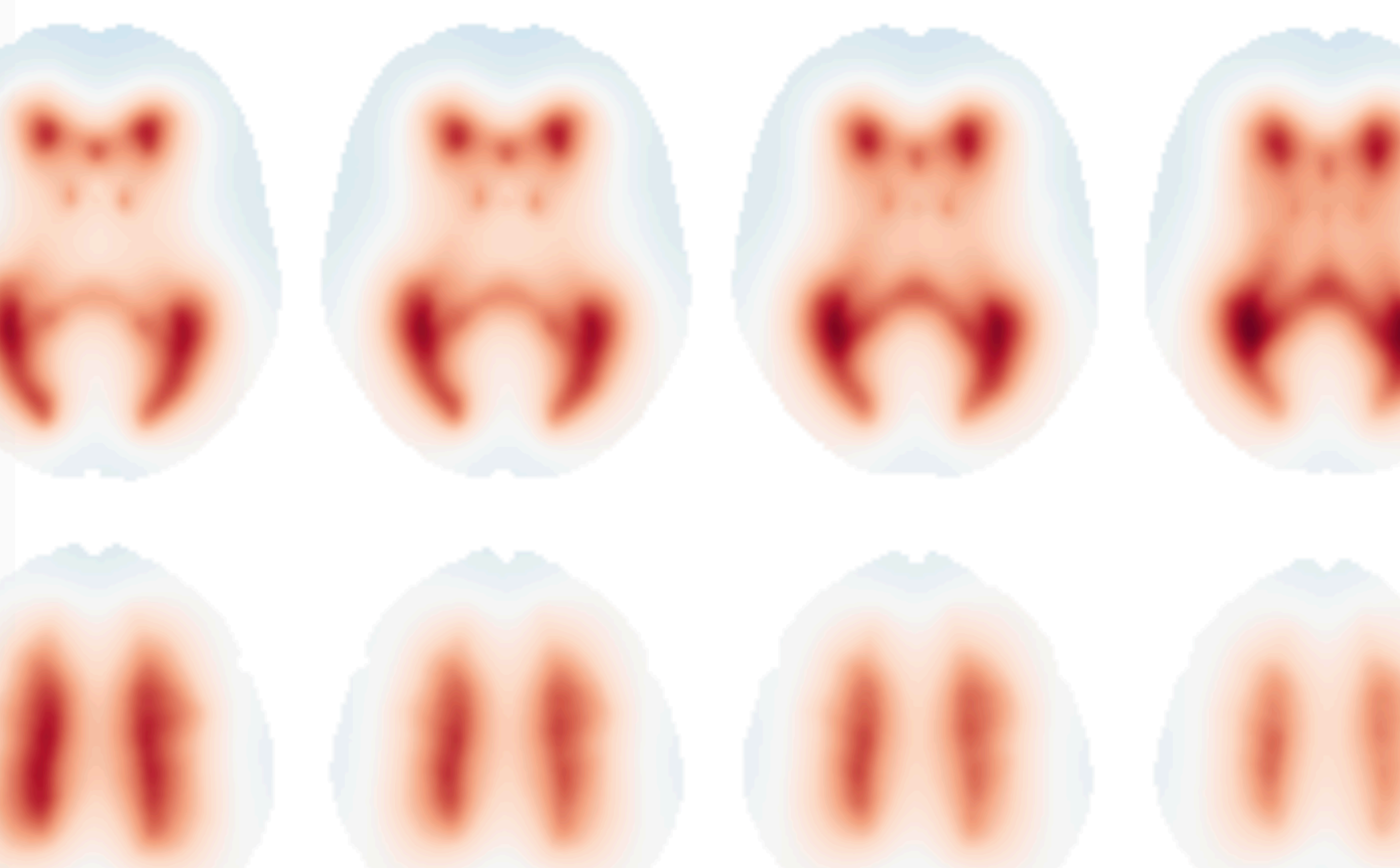
Paul Schmidt

# Bayesian inference for structured additive regression models for large-scale problems with applications to medical imaging

Dissertation an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

Eingereicht am 07. November 2016



Paul Schmidt

# Bayesian inference for structured additive regression models for large-scale problems with applications to medical imaging

Dissertation an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

Eingereicht am 07. November 2016

Erster Berichterstatter: Prof. Dr. Volker Schmid

Zweiter Berichterstatter: Prof. Dr. Thomas Kneib

Dritter Berichterstatter: Prof. Dr. Mark Mühlau

Tag der Disputation: 19. Januar 2017

# Danksagung

Diese Arbeit würde ohne die Unterstützung vieler Menschen nicht existieren. Allen voran möchte ich mich ganz herzlich bei meinem Doktorvater Prof. Dr. Volker Schmid für die Möglichkeit der Promotion sowie die tolle Betreuung trotz räumlicher Entfernung bedanken.

Ein herzlicher Dank geht auch an Prof. Dr. Mark Mühlau für die exzellente Betreuung und Zusammenarbeit sowie die zahlreichen Diskussionen. Durch sein Vertrauen in meine Arbeit war es mir möglich, auch fernab von München für seine Arbeitsgruppe zu arbeiten, woraus überhaupt die Idee für diese Arbeit entstand. In diesem Rahmen möchte ich mich auch bei der Arbeitsgruppe Morphometrie des Neuroimaging Center der TU München bedanken, insbesondere bei Dr. Viola Biberacher für die angenehme und reibungslose Zusammenarbeit. Bei der Arbeitsgruppe Schmerz bedanke ich mich bei Dr. Elisabeth May, Dr. Laura Thiemann sowie Moritz Nickel für die tolle Atmosphäre im und die schöne Zeit außerhalb des Büros.

Des Weiteren möchte ich mich bei Prof. Dr. Thomas Kneib für den aufschlussreichen E-Mail-Verkehr sowie bei Dr. Stephanie Thiemichen für die hilfreichen Erläuterungen zum Promotionsvorgang bedanken.

Ein sehr großer Dank geht an meine Eltern, Monika und Hans-Joachim Schmidt. Ohne ihre Unterstützung wäre ich nicht dort, wo ich jetzt bin. Ebenfalls möchte ich mich bei Dr. Coralie Wink und Prof. Dr. Michael Wink für die nützlichen Hinweise bedanken. Charlotte Wink und Laura Menz danke ich ganz herzlich für die angenehme Unterbringung sowie Fürsorge während meiner zahlreichen Besuche in München.

Besonders bedanken möchte ich mich bei meiner Partnerin Lucie Wink. Durch ihr Vertrauen, ihren Rückhalt und Beistand hatte sie indirekt einen großen Anteil an der Fertigstellung dieser Arbeit. Und schließlich möchte ich mich bei unserem Sohn Jakob dafür bedanken, dass er die letzten fünf Monate zu den schönsten meines Lebens gemacht hat.

# Zusammenfassung

In der angewandten Statistik können Regressionsmodelle mit hochdimensionalen Koeffizienten auftreten, die sich nicht mit gewöhnlichen Computersystemen schätzen lassen. Dies betrifft unter anderem die Analyse digitaler Bilder unter Berücksichtigung räumlich-zeitlicher Abhängigkeiten, wie sie innerhalb der medizinisch-biologischen Forschung häufig vorkommen.

In der vorliegenden Arbeit wird ein Verfahren formuliert, das in der Lage ist, Regressionsmodelle mit hochdimensionalen Koeffizienten und nicht-normalverteilten Zielgrößen unter moderaten Anforderungen an die benötigte Hardware zu schätzen. Hierzu wird zunächst im Rahmen strukturiert additiver Regressionsmodelle aufgezeigt, worin die Limitationen aktueller Inferenzansätze bei der Anwendung auf hochdimensionale Problemstellungen liegen, sowie Möglichkeiten diskutiert, diese zu umgehen. Darauf basierend wird ein Algorithmus formuliert, dessen Stärken und Schwächen anhand von Simulationsstudien analysiert werden. Darüber hinaus findet das Verfahren Anwendung in drei verschiedenen Bereichen der medizinisch-biologischen Bildgebung und zeigt dadurch, dass es ein vielversprechender Kandidat für die Beantwortung hochdimensionaler Fragestellungen ist.

# Summary

In applied statistics regression models with high-dimensional coefficients can occur which cannot be estimated using ordinary computers. Amongst others, this applies to the analysis of digital images taking spatio-temporal dependencies into account as they commonly occur within bio-medical research.

In this thesis a procedure is formulated which allows to fit regression models with high-dimensional coefficients and non-normal response values requiring only moderate computational equipment. To this end, limitations of different inference strategies for structured additive regression models are demonstrated when applied to high-dimensional problems and possible solutions are discussed. Based thereon an algorithm is formulated whose strengths and weaknesses are subsequently analyzed using simulation studies. Furthermore, the procedure is applied to three different fields of bio-medical imaging from which can be concluded that the algorithm is a promising candidate for answering high-dimensional problems.

# Contents

# 1 Introduction

## 1.1 Large-scale problems

### 1.1.1 The problem

The concept of regression analysis is the most common statistical problem applied statisticians are faced with today. Here, the main goal is to describe the relationship between a number of covariates on a response variable of interest. Despite all methodological and technical progress the statistical community has witnessed in the last decades there are still situations where it is not able to fit even simple regression models due to the size of the problem. Consider for example a regression model whose linear predictor contains a random intercept to account for heterogeneity among certain statistical units:

$$\eta_i = \cdots + \gamma_{\mathrm{Unit}[i]} + \cdots .$$

If the number of units, $m$, exceeds a certain limit, i.e. if the dimension of $\boldsymbol{\gamma} = (\gamma_{\mathrm{Unit}[1]}, \ldots, \gamma_{\mathrm{Unit}[m]})'$ is too large, it may not be possible to obtain a solution for this relatively simple regression problem due to restricted computational resources. The situation worsens if, in addition, special dependency structures are assumed between units. For example, the units may represent days of the year which may require the inclusion of some form of temporal dependency. Similarly, if the units are spatially aligned, it may be useful to account for spatial information by appropriate model extensions. The statistical analysis of digital images (Besag, 1989) represents an example of the latter. Here, the number of units, i.e. the number of picture elements (pixels) or volume elements (voxels), is directly related to the resolution of the image. The higher the resolution, the more pixels are included into the analysis which leads to larger regression coefficients. Furthermore, the spatial alignment over a regular lattice represents a natural source of dependency which needs to be accounted for.

In this thesis, a method is provided that allows to fit models with *high-dimensional* or *large-scale* regression coefficients on moderate working stations. It is assumed that the problem can be formulated as a regression problem, that is, a well defined response vector as well as certain explanatory variables are available. The problem of interest arises if the dimension of one or more regression coefficients are too large. Usually, when estimating the coefficients of regression models, it is required to solve systems of linear equations. If regression coefficients are high-dimensional, solving these systems may become impossible given only moderate computational equipment. Additional problems arise when non-Gaussian response variables are considered such as in the framework of generalized linear models (GLMs, Nelder and Wedderburn, 1972).

The necessity to provide a solution to the above problem may come as a surprise as we are constantly faced with technological progress (Moore et al., 1975). In particular, one could argue that it will not be long until these problems can be solved using affordable hardware. The following points hold against this argument: First, in order to keep pace with the newest technological achievements a certain financial basis is required which may not be available to all applied statisticians. Second, regression problems that are on the edge of current computational abilities will always exist. Thus, in providing a way to solve such problems using available hardware means that the critical limit for which no more solutions can be obtained is raised even for future problems. Thus, the method introduced in this thesis allows one to encounter problems that are difficult to handle even with future equipment. This will increase the ability to solve large-scale regression problems not only for today, but also for the forthcoming years and decades.

### 1.1.2 Differentiation from other big data problems

Recently, the term “big data” has gained considerable attention in the public discourse. As a result, many resources went into the development of strategies that are able to solve such big data problems. Usually, methods that are designed to work with big data can be identified as data mining tools that are able to handle huge amounts of observations. In contrast, the present thesis puts the focus on valid statistical inference by the use of regression analyses. Thus, the expression “big data” is explicitly avoided in this thesis. Instead, the terms “large-scale” or “high-dimensional” are used. However, for these expressions no clear definition exists in the statistical literature. Often, “high-dimensional” refers to so-called large $p$ small $n$ problems, that is, situations where there are more unknown parameters $p$ than data points $n$ (Johnstone and Titterington, 2009). Such problems can be found, for example, in the analysis of microarrays and related fields in genomics (Bickel et al., 2009). Handling large $p$ small $n$ situations usually means to apply inference to a set of relevant predictors, i.e. to perform some kind of variable selection. Shrinkage or penalization approaches such as Ridge regression (Hoerl and Kennard, 1970) or the Lasso method (Tibshirani, 1996) can be applied to these kind of problems. While, under certain conditions, large $p$ small $n$ problems can also be treated with the methods presented in this thesis, they are not of primary interest here.

### 1.1.3 How big is too large?

Due to new technological achievements the critical size of the dimension of regression coefficients is a dynamic variable rather than a constant. Today, statisticians are able to fit models in a few minutes that would have taken days or weeks one or two decades ago. Similarly, situations that are problematic today may be considered as moderate tomorrow. Thus, technological progress will always ensure that the term “too big” does not constitute a fixed definition. Obviously, this discussion also applies to the term „moderate“ as in „moderate working station“. Therefore, in this thesis data situations are considered as large-scale or high-dimensional if standard software packages fail to execute the corresponding analysis on ordinary working stations. Of course, the methods presented in this thesis also have their limit. However, it is expected that this limit lies several factors above the one that is processable by standard software packages.

### 1.1.4 Previous work on high-dimensional regression models

In recent years different approaches for fitting regression models with large-scale coefficients have been published. Due to its option of working on each coefficient vector separately, most methods perform the fitting process via Gibbs sampling within a fully Bayesian setup based on Markov chain Monte Carlo (MCMC, see Section 3.1). The main computational bottleneck when applying the Gibbs sampler to large-scale regression problems is the generation of samples from high-dimensional Gaussians (Section 4.2.1). Thus, the focus of most approaches lies on techniques to overcome this problem. For example, in the context of high-dimensional inverse problems with regard to the analysis of digital images, Bardsley (2012) utilized iterative methods for sparse linear systems as discussed in Section 4.2.2. Zhang et al. (2013) sampled a proposal from a high-dimensional Gaussian using the ordinary Cholesky decomposition with special permutation techniques (Section 4.2.1).

They were able to successfully apply their algorithm to seismic tomography data with more than 11,000 parameters. However, this number is not comparable to the parameters in later applications of this thesis which all include more than 500,000 components to estimate. In order to estimate spatially varying effects, Ge et al. (2014) sampled from large Gaussians utilizing a checkerboard-like conditional independence structure which occurs naturally on a regular lattice. Although easy to implement, this approach is limited to special situations and further adaptations are needed to guarantee good mixing of Markov chains for all parameters (Section 4.2.1). Furthermore, all of the approaches mentioned above are restricted to Gaussian response models – Ge et al. utilized the auxiliary variable method by Albert and Chib (1993) to encounter the non-normal nature of their data. However, a declared aim of this thesis is to provide a framework that is able to fit models with general responses. On this subject, only a few works have been published. The most promising approach is the one recently proposed by Wood et al. (2015). They presented a relatively fast method for fitting generalized additive models to large data sets within a frequentist setting by utilizing iterative updating schemes for the factorization of model matrices. However, their approach still fails with respect to computational requirements when applied to the applications presented in later chapters of this thesis.

## 1.2 Applications in medical imaging

In the last half of the 20th century new medical imaging devices were developed. Subsequently, new statistical methods for the analysis of the corresponding medical images emerged which also includes different forms of regression analysis. In the following, three applications of regression analysis to bio-medical images are presented.

### 1.2.1 Tissue segmentation

Medical images play an important part in the process of diagnosis of various diseases. Consider, for example, the identification of cancerous tissue in digital mammography, visualization of lung tumors by computed tomography, or the recognition of inflammatory plaques in three-dimensional magnetic resonance (MR) images of the brain of patients with multiple sclerosis (MS). In all these cases it is required to classify (segment) the depicted parts of the body into normal appearing and suspicious tissue. In addition, non-diagnostic applications of tissue segmentation exist as well. For example, in neuroimaging classification of head MR images into the three major components of the brain, i.e. cerebrospinal fluid, gray matter (GM), and white matter (WM), is required in order to measure and compare GM atrophy along different cohorts or patient groups. While manual segmentation by trained experts is a valid option, it is advantageous to use automatic segmentation algorithms in order to minimize rater specific biases as well as costs and operator time.

In general, tissue segmentation methods can be divided into supervised and unsupervised approaches (Bezdek et al., 1992). The most popular methods among unsupervised approaches are cluster algorithms. Hierarchical clustering (Ohkura et al., 2000) as well as more sophisticated model-based clustering techniques (Wells III et al., 1996) are commonly used. Within the latter class finite mixtures of Gaussians take a special position. The advantage of finite mixture models is that they can easily be expanded in order to include spatial dependencies by the use of Markov random fields (Winkler, 2003). Supervised approaches, on the other hand, mostly use methods from the machine learning community such as classifiers based on $k$-nearest neighbors (Anbeek et al., 2004), neural networks (Ghosh et al., 1991), and classifiers that are trained by support vector machines (Wang et al., 2001). One approach that has received very little attention in the past is image segmentation based on binary regression models. This is due to the fact that individual training of voxels is often not possible as not all voxels show the tissue of interest and, thus, no valid results can be obtained for these voxels. A possible solution to this would be to consider all voxels jointly when training classifiers based on binary regression models. In this context it may be helpful to consider spatial dependencies between voxels. Such an approach is taken in Section 6.1 of this thesis in order to formulate a supervised MS lesion segmentation algorithm.

### 1.2.2 Pixel- and voxel-wise regression models

Another application of regression models within the analysis of medical images are pixel- or voxel-wise regression models. Such models are widely used in neuroimaging. For example, when identifying activated brain regions by the use of functional magnetic resonance imaging (fMRI) time series models are fitted to each voxel in order to obtain activation maps of the brain (Lindquist et al., 2008). Once these activation maps for individual subjects have been computed they are further compared among groups or experimental conditions by a so-called second-level analysis. Here, normalized activation maps are analyzed by a general linear model (Worsley and Friston, 1995), that is, simple linear models are fitted for all voxels separately. Application of this approach to the analysis of local GM volume obtained from structural MR images is known as voxel-based morphometry (Ashburner and Friston, 2000). With respect to non-brain medical images pixel-wise regression models are commonly seen in quantitative analysis of dynamic contrast-enhanced MR images. Here, nonlinear regression models are fitted to each pixel of a time series of images in order to estimate kinetic parameters within pharmacokinetic models (Tofts, 1997).

Although pixel- and voxel-wise regression models appear in many different applications, the main modeling approach is nevertheless rather similar. For example, for Gaussian responses the following procedure can often be observed. First, in order to account for spatial dependencies among voxels the images of all subjects are smoothed by a special filter which most probably depends on a pre-chosen smoothing parameter. Subsequently, a linear regression model is fitted to each voxel separately. Although this approach yields useful results its disadvantage is that spatial information can only be accounted for by modifying the actual data using a smoothing parameter that is chosen by the user. Instead, it is favorable to estimate the amount of smoothness automatically from the original data, as this does not only increases the signal to noise ratio (Penny et al., 2005) but also leads to more powerful analyses (Schmidt et al., 2013). However, estimating the smoothing parameter adaptively from the data requires to model all voxels jointly. Such models have been extensively studied with respect to fMRI time-series analysis. For example, Gössl et al. (2000) fitted Gaussian response models within a fully Bayesian setup based on MCMC. Their procedure allows to derive the full conditionals for each voxel and, thus, to work on each voxel separately by still considering the joint model formulation. However, updating the parameters for each voxel independently may result in insufficient mixing as well as slow convergence of Markov chains to their corresponding stationary distributions (Knorr-Held and Rue, 2002). In contrast, the methods derived in this thesis allow updating these parameters jointly. For example, in Section 6.2 a large-scale voxel-wise regression model is formulated for images of local GM volume obtained from structural MR images over 565,475 voxels. In addition, the presented approach allows fitting models with non-Gaussian responses as well.

### 1.2.3 Spatial information in object-based co-localization

Statistical methods play an important role in the analysis of digital microscopy images. Early applications are the identification – detection as well as segmentation – of objects of interest, such as cells or sub-cellular structures. For this task a broad range of methods exist, see for example Xing and Yang (2016) for an extensive review. Besides object identification and labeling the analysis of spatial relationships between sub-cellular structures and their relation to cell processes are of particular interest. This kind of analysis is known as co-localization (Dunn et al., 2011). In particular, by using fluorescence microscopy specially marked cell components can be identified and visualized in high resolution. This allows to draw conclusions about the appearance of certain cellular objects in the presence of other structures within their neighborhood. Early methods in co-localization analyze intensity values of pixels by the use of correlation coefficients (Manders et al., 1992) or overlap measures (Manders et al., 1993). In object-based co-localization, on the other hand, the available data is limited to identified (segmented) objects of interest for which the distances to other structures are recorded. By thresholding these distances the effect of neighboring structures on the objects of interest can be revealed. Recently, methods that do not rely on thresholding but explore interaction of cellular structures as a function of their distances instead, have been proposed (Helmuth et al., 2010). While these approaches allow more flexibility in modeling the relation between neighboring structures, they miss the potential effect of the location of the objects of interest within the cell itself. In Section 6.3, a Poisson process which is able to account for this spatial information is fitted to the data of a 3D fluorescence microscopy image of 514,442 voxels.

## 1.3 Outline

### 1.3.1 Thesis objectives

This work has four main objectives. The first goal is the identification of potential vulnerabilities of popular inference strategies when applied to high-dimensional problems. Here, the focus lies on methods that are able to fit structured additive regression (STAR, Fahrmeir et al., 2004) models. Once these vulnerabilities have been identified and discussed, an inference strategy has to be selected for which solutions to these problems are presented in the second step. These solutions should be designed in a way so that they can be applied with low to moderate computational equipment. At the end, a complete framework for the estimation of large-scale STAR models should be formulated. The third objective involves the analysis of the performance of the proposed methods, that is, it should be shown in which way these methods have an influence on the final inference. Of particular importance is the identification of situations where one can expect to obtain good results and situations where applying these methods is not suitable. Finally, by applying the complete framework to different real world problems its practical relevance will be demonstrated.

### 1.3.2 Structure of thesis

The rest of this thesis is organized as follows. First, the theoretical foundations of STAR models are presented. To this end, Chapter 2 provides the conceptual bases by introducing all necessary model components and reviewing certain effect types. In Chapter 3, three currently used approaches for fitting STAR models are reviewed. This includes a fully Bayesian approach based on MCMC, an empirical Bayes method utilizing a mixed model representation, and the integrated nested Laplace approximation approach for approximate inference. Chapter 4 analyzes these approaches first in terms of their usefulness with respect to high-dimensional problems. As a result, one approach is then selected which is subsequently adapted to large-scale problems. The performance of these adaptations and the effect on the final results are extensively analyzed in Chapter 5 by simulation studies. These insights are used to formulate a complete framework which is applied to different examples from the field of bio-medical imaging in the following chapter. The thesis concludes with a summary chapter and outlook (Chapter 7), as well as an appendix with implementation details.

### 1.3.3 Contributed Manuscript

Some parts of this thesis can be found in the following work which has been published in the preparation of this thesis:

> Schmidt, P., Mühlau, M., and Schmid, V. Fitting large-scale structured additive regression models using Krylov subspace methods. *Computational Statistics & Data Analysis*, 105:59 – 75, 2017.

The individual contributions of the authors to this article are as follows: The main idea of using Krylov subspace methods emerged from discussions between Volker Schmid and Paul Schmidt. In addition, Volker Schmid supervised the overall structure of the article and all statistical aspects. He also initiated larger simplifications and more detailed explanations of the manuscript. Furthermore, he was involved in the formulation of answers to the reviewers within the review process. Mark Mühlau provided the data set for the first application and proposed choosing data from the ADNI data base for the second application. He also provided information about Alzheimer's disease and formulated the corresponding part in the introduction of the second application. Most of the manuscript was written by Paul Schmidt who also implemented the algorithm, and conducted the simulation as well as data analysis. All authors participated in proof-reading the manuscript.

Detailed information about the content of this thesis which has already been published in a modified form in the above article will be provided at the beginning of each chapter.

# 2 Structured additive regression models

This chapter reviews STAR models. Adequate modeling of a large variety of effect types, including temporal, spatial and spatio-temporal effects, as well as nonlinear effects for continuous covariates (Lang and Brezger, 2004) using this class of models will be demonstrated. In addition, STAR models are appropriate for modeling large-scale problems as discussed further in Chapter 4. For a more comprehensive introduction of STAR and related models see, for example, Kneib (2006) and Fahrmeir and Kneib (2011).

The concept of regression considered here follows the idea of classical regression analysis, namely that the conditional expectation of the $i$th response $y_i$ given the values of $p$ explanatory variables $z_{i1}, \dots, z_{ip}$ can be expressed as a function of these variables:

$$\mathrm{E}(y_i | z_{i1}, \dots, z_{ip}) \approx f(z_{i1}, \dots, z_{ip}). \tag{2.1}$$

Objective of this chapter is to specify all components of (2.1) that are required to fully describe the functional relationship between the explanatory variables and the observed response. Chapter 3 then presents ways to perform inference on all parameters of interest.

The information given in Section 2.1, 2.2.1, and 2.2.2 below can also be found in shortened form in Schmidt et al. (2017, Section 2.1 and 2.2).

## 2.1 Observation model

Throughout the following chapters, the probability density function of the vector of response variables $\boldsymbol{y} = (y_1, \dots, y_n)'$ is denoted by $p(\boldsymbol{y}|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a collection of all unknown parameters. It is further assumed that the response variables are conditionally independent given explanatory variables $\boldsymbol{z} = (\boldsymbol{z}'_1, \dots, \boldsymbol{z}'_n)'$ with $\boldsymbol{z}_i = (z_{i1}, \dots, z_{ip})'$, and unknown parameters $\boldsymbol{\theta}$, so that $p(\boldsymbol{y}|\boldsymbol{\theta}) = \prod_{i=1}^{n} p(y_i|\boldsymbol{z}_i, \boldsymbol{\theta})$. The nonlinear relation given in (2.1) is divided into the following structural assumptions: First, the conditional expectation

$\mu_i = \mathrm{E}(y_i|\boldsymbol{z}_i, \boldsymbol{\theta})$ is linked to the linear predictor $\eta_i$ by a known link function $g$, i.e. $g(\mu_i) = \eta_i$. Alternatively, one can specify this relation equivalently by the inverse link function, i.e. the response function $h = g^{-1}$ and $\mu_i = h(\eta_i)$. Second, it is assumed that the general nonlinear relationship between $\eta_i$ and $\boldsymbol{z}_i$ can be approximated by a linear combination of nonparametric functions $f_1, \dots, f_p$ of $\boldsymbol{z}$:

$$\eta_i = f_1(z_{i1}) + \cdots + f_p(z_{ip}). \tag{2.2}$$

It is further assumed that the function evaluations $\boldsymbol{f}_k = (f_k(z_{1k}), \dots, f_k(z_{nk}))'$ can be written as a combination of a $n \times m_k$ design matrix $\boldsymbol{Z}_k$ and a $m_k \times 1$ vector of unknown coefficients $\boldsymbol{\gamma}_k$ as

$$\boldsymbol{f}_k = \boldsymbol{Z}_k \boldsymbol{\gamma}_k. \tag{2.3}$$

This yields the linear predictor in compact matrix notation as

$$\boldsymbol{\eta} = \boldsymbol{Z}_1 \boldsymbol{\gamma}_1 + \cdots + \boldsymbol{Z}_p \boldsymbol{\gamma}_p.$$

With this definition, the set of unknown parameters $\boldsymbol{\theta}$ consists of the regression coefficients $\boldsymbol{\gamma}_k, k = 1, \dots, p$, and an additional dispersion parameter $\phi$ that may be included in $p(\boldsymbol{y}|\boldsymbol{\theta})$. If seen as a function of $\boldsymbol{\theta}$, $p(\boldsymbol{y}|\boldsymbol{\theta})$ is referred to as the *likelihood function*.

The distributional assumption given above allows the consideration of a broad range of distributions for the response variable. Other authors, however, put more emphasis on distributions that are exponential families (Fahrmeir and Tutz, 2001). This has the advantage that well known results within the framework of generalized linear models (GLMs, Nelder and Wedderburn, 1972) can easily be integrated into the estimation concepts presented in Chapter 3. For exponential families the conditional distribution of $y_i$ given $\boldsymbol{z}_i$ can be written as

$$p(y_i|\boldsymbol{z}_i) \propto \exp\left(\frac{y_i \vartheta(\mu_i) - b(\vartheta(\mu_i))}{\phi}\omega_i + c(y_i, \phi, \omega_i)\right). \tag{2.4}$$

Here, $\vartheta$ is a function of the expectation $\mu_i$ and is called natural parameter, and $\phi$ is the additional scale or dispersion parameter that was introduced before. The functions $b(\cdot)$ and $c(\cdot)$ depend on the choice of the specific exponential family. Similar to the structural assumptions above, the expectation $\mu_i$ is linked to the linear predictor $\eta_i$ by a known response function $h$, $\mu_i = h(\eta_i)$.

## 2.2 Prior specification

So far, $p + 1$ unknown parameters have been introduced: the regression coefficients $\boldsymbol{\gamma}_k, k = 1, \ldots, p$, and the dispersion parameter $\phi$. To complete the model formulation, prior distributions for these parameters and for additional hyperparameters need to be set up. In order to simplify notation the subscript $k$ is suppressed for $\boldsymbol{\gamma}_k$ in the remaining of this chapter. Thus, the regression coefficient is represented by the $m$-dimensional vector $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_m)'$.

### 2.2.1 Regression coefficients

Within the concept of STAR models, priors for the regression coefficients are usually multivariate normals (Fahrmeir et al., 2004):

$$\boldsymbol{\gamma}|\kappa \sim \mathrm{N}(\mathbf{0}, \boldsymbol{Q}^{-1}). \tag{2.5}$$

Here, $\mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes for the multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance-matrix $\boldsymbol{\Sigma} = \boldsymbol{Q}^{-1}$ or, alternatively, precision matrix $\boldsymbol{Q} = \boldsymbol{\Sigma}^{-1}$ which usually depends on a precision parameter $\kappa$, i.e. $\boldsymbol{Q} = \boldsymbol{Q}(\kappa)$. As it is shown in subsequent sections, prior information is typically induced by specific choices of the precision matrix: the elements in $\boldsymbol{\gamma}$ can be directly related to each other in various ways by imposing different assumptions on the structure of $\boldsymbol{Q}$, which, in turn, may result in appropriate prior configurations for modeling temporal or spatial information. In order to further clarify the role of the elements in $\boldsymbol{Q}$ it is useful to point out the connection of prior (2.5) to Gaussian Markov random fields (GMRFs) and therefore to undirected graphs. According to Rue and Held (2005), a vector $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_m)'$ is called a GMRF with respect to a labelled graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertices or nodes $\mathcal{V} = \{1, \ldots, m\}$ and edges $\mathcal{E}$ if and only if its density has the form

$$p(\boldsymbol{\gamma}) = (2\pi)^{-m/2} |\boldsymbol{Q}|^{1/2} \exp\left(-\frac{1}{2}(\boldsymbol{\gamma} - \boldsymbol{\mu})' \boldsymbol{Q} (\boldsymbol{\gamma} - \boldsymbol{\mu})\right). \tag{2.6}$$

In this notation, $\boldsymbol{\mu}$ is the mean and $\boldsymbol{Q}$ is a symmetric positive-definite precision matrix whose nonzero pattern defines the dependence structure of the vertices in $\mathcal{G}$. From this definition it can be seen that every multivariate normal distributed vector forms a GMRF. Thus, the non-zero pattern of $\boldsymbol{Q}$ of prior (2.5) directly induce the neighborhood structure of an undirected graph. Usually, this neighborhood structure is used to impose some kind of smoothness-penalty on the elements of $\boldsymbol{\gamma}$: too rough deviations from neighboring elements are penalized in order to obtain a fairly smooth effect of interest. Thus, by (2.5) prior information is only provided for differences of parameters, not for their expectations.

For some specifications discussed below, the precision matrices are not of full rank which leads to (partially) improper prior distributions. Improper GMRFs are known as intrinsic GMRFs (IGMRFs, Rue and Held, 2005). The rank deficiency is equivalent to the dimension of the null space of $\boldsymbol{Q}$, which consists of all vectors $\boldsymbol{x} \neq \boldsymbol{0}$ for which $\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{0}$ holds. It is important to note that, although (2.6) may be improper, the corresponding posterior is usually proper, thus, the rank deficiency yields no restriction with regard to inference. However, since $\boldsymbol{Q}$ is singular the determinant in (2.6) does not exist. If it is required the determinant can be replaced by the generalized determinant which can be calculated by the product of all non-zero eigenvalues of $\boldsymbol{Q}$ (Rue and Held, 2005, p. 93).

In the remainder of this section it is shown that a large variety of effect types can be represented as a GMRF and written as (2.3). With regard to the adjustments for large-scale data an additional focus is put on the sparsity structure of design and precision matrices.

### Fixed effects

A non-informative prior for a $m$-dimensional vector of parametric or fixed effects can be achieved by setting

$$\boldsymbol{Q} = \kappa \boldsymbol{I}_m \tag{2.7}$$

and choosing a fixed small value for $\kappa$. In this notation, $\boldsymbol{I}_m$ stands for the $m$-dimensional identity matrix. For the limiting case of $\kappa \to 0$ this choice leads to the improper prior $p(\gamma) \propto 1$ which usually gives similar results and is preferred by other authors, for example Fahrmeir et al. (2004).

While the $m \times m$ precision matrix of such a fixed effect is diagonal and can therefore be considered as sparse, the $n \times m$ design matrix $\boldsymbol{Z}$ is usually dense. Most STAR models include at least one fixed effect, namely the overall intercept.

The corresponding graph that is induced by (2.7) is trivial: all $m$ nodes are independent, i.e. there are no connections between any nodes in $\mathcal{G}$. Since $\boldsymbol{Q}$ is diagonal, its determinant can be calculated by the product of its diagonal elements.

### Unstructured random effects

As there is no unique definition of a "random effect" in statistics (Gelman and Hill, 2007, Section 11.4), it may be necessary to first clarify this term's meaning within the context of STAR models, especially as in a fully Bayesian setup the phrase "random effect" maybe misleading since all unknown parameters are considered as random. However, here, the term "unstructured random effect" refers to effects that are able to account for heterogeneity among groups or clusters of observational units or that account for subject specific deviations from population effects in longitudinal data. This corresponds to effects as they are included along parametric or fixed effects in common mixed models (Pinheiro and Bates, 2000). Often, these effects are further divided into effects that account for changes in the intercept (random intercept) and for changes in the coefficients of certain explanatory variables (random slope). The term "independent and identically distributed (i.i.d.) random effect" is also used within this context.

An appropriate prior for modeling unstructured random effects can be obtained from (2.7) by considering $\kappa$ as an additional hyperparameter rather than being fixed. This prior specification implies that all elements in $\boldsymbol{\gamma}$ are drawn from the same normal distribution with equal precision $\kappa$. Often, the desired effect of *shrinkage* or *partial pooling* along the elements in $\boldsymbol{\gamma}$ is a direct consequence of this assumption. This behavior is evident from the full conditional prior distribution for $\gamma_j$ given all other parameters:

$$\gamma_j | \boldsymbol{\gamma}_{-j}, \kappa \sim \mathrm{N}(0, \kappa^{-1}).$$

While the overall mean is zero the precision $\kappa$ plays the role of a shrinkage parameter: the higher $\kappa$ the more the elements in $\boldsymbol{\gamma}$ are pooled towards their overall mean.

As for the fixed effect definition above, the $m \times m$ precision matrix is sparse with $m$ non-zero elements, that is $m/(m \cdot m) \cdot 100\% = 1/m \cdot 100\%$ of its entries are non-zero elements. However, in contrast to fixed effects, the $n \times m$ design matrices associated with unstructured random effects are sparse as well. For a random intercept $\boldsymbol{Z}$ is an indicator matrix which allocates each observation to its corresponding group, cluster or subject. Therefore, in most cases, $\boldsymbol{Z}$ only has $n/(m \cdot n) \cdot 100\% = 1/m \cdot 100\%$ non-zero elements. For random slopes this design matrix needs to be multiplied in a column-wise manner by the values of the explanatory variable $\boldsymbol{z}$. Thus, the sparsity structure remains identical. An exception is the usage of this shrinkage prior in the context of ridge regression (Hoerl and Kennard, 1970). Here, the design matrix is a concatenation of different explanatory variables and, thus, usually dense.

The graph that is induced by this prior is analogous to the graph for the fixed effect setup above, that is, there is no connection between any vertices in $\mathcal{G}$. Again, the determinant of $\boldsymbol{Q}$ can be obtained by the product of its diagonal elements.

### Temporal effects

Temporal dependence between the elements of $\boldsymbol{\gamma}_k$ can be induced by prior (2.5) by random walk priors of order one (RW1) or two (RW2). In practice this is often used for modeling smooth temporal effects in a nonparametric fashion, for example when fitting autoregressive and dynamic models (West and Harrison, 1997). In this setup, it is assumed that the elements of $\boldsymbol{\gamma}$ are defined over a set of equidistant nodes $\{1,\dots,m\}$. The sequential conditional prior distributions for $\gamma_j$ given previous time points is then given by

$$\gamma_j|\gamma_{j-1},\kappa \sim \mathrm{N}(\gamma_{j-1},\kappa^{-1}), \qquad j=2,\dots,m \tag{2.8}$$

for the RW1 and

$$\gamma_j|\gamma_{j-1},\gamma_{j-2},\kappa \sim \mathrm{N}(2\gamma_{j-1}-\gamma_{j-2},\kappa^{-1}), \qquad j=3,\dots,m \tag{2.9}$$

for the RW2 prior. By using the Hammersley-Clifford theorem (Hammersley and Clifford, 1971) it can be shown that, under appropriate corrections for the first and first and second nodes, respectively, the corresponding joint priors of (2.8) and (2.9) can be written as

$$p(\boldsymbol{\gamma}|\kappa) \propto \exp\left(-\frac{\kappa}{2}\sum_j(\triangle\gamma_j)^2\right) \qquad \text{and} \qquad p(\boldsymbol{\gamma}|\kappa) \propto \exp\left(-\frac{\kappa}{2}\sum_j(\triangle^2\gamma_j)^2\right)$$

respectively, with $\triangle\gamma_j = \gamma_j - \gamma_{j-1}$ and $\triangle^2 = \gamma_j - 2\gamma_{j-1} + \gamma_{j-2}$. To derive the joint priors in matrix notation the $(m-1)\times m$ and $(m-2)\times m$ difference matrices

$$\boldsymbol{D}_1 = \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{D}_2 = \begin{pmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \end{pmatrix}$$

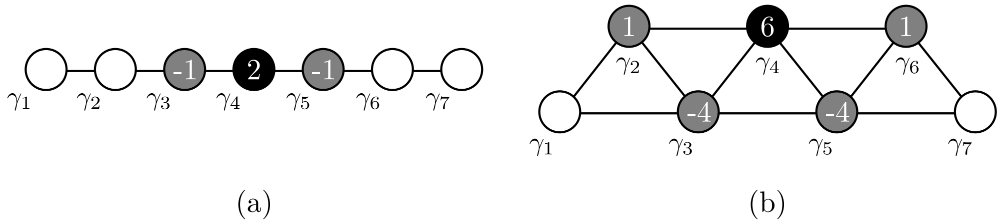


Figure 2.1 Visualization of undirected graphs according to the RW1 (a) and RW2 (b) prior specification. Only nodes with a direct connection are considered as neighbors. Neighboring nodes of the black nodes are colored gray. Numbers refer to the entries of the fourth row of the corresponding structure matrices.

are defined, respectively. The joint priors can then be written as

$$p(\boldsymbol{\gamma}|\kappa) \propto \exp\left(-\frac{\kappa}{2}\boldsymbol{\gamma}'\boldsymbol{K}_d\boldsymbol{\gamma}\right) \tag{2.10}$$

with structure or penalty matrices $\boldsymbol{K}_d = \boldsymbol{D}'_d\boldsymbol{D}_d, d = \{1, 2\}$ of dimension $m \times m$. Hence, a suitable prior for modeling temporal effects by using random walks can be written as (2.5) with precision $\boldsymbol{Q} = \kappa\boldsymbol{K}_d$. The graphs that correspond to these random walk priors are defined on regular locations as depicted in Figure 2.1. Panel (a) of this figure shows the pattern of the RW1 prior where only the direct neighbors (gray) to the left and to the right have a direct influence on node four (black) through prior (2.10). The numbers on the nodes refer directly to the non-zero entries of the fourth row in the corresponding structure matrix $\boldsymbol{K}_1$. In Panel (b) the structure of a RW2 is shown. Here, in contrast to the RW1, the fourth node does also depend on the second neighbors. Similar to Panel (a), the numbers refer to the non-zero values in the fourth row of $\boldsymbol{K}_2$.

In accordance to unstructured random effects, the $n \times m$ design matrix is an indicator matrix which assigns each observation to its correspondent element in $\boldsymbol{\gamma}$. However, rather than being diagonal, the structure matrix $\boldsymbol{K}_d$ is a band matrix with $2 + m(1 + 2d) - 4d$ non-zero elements. Compared to $m \cdot m$ elements in total, this can usually be considered as sparse. From its construction it can also be seen that $\boldsymbol{K}_d$ is rank deficient: For $d = 1$, the joint prior $p(\boldsymbol{\gamma}|\kappa)$ is invariant to the addition of a constant vector $(c, \dots, c)$' of dimension $m$, thus $\text{rk}(\boldsymbol{K}_1) = m - 1$. For $d = 2$, the prior is also invariant to the addition of any linear trend, that is, the the null space also includes the vector $c \cdot (1, \dots, m)$', thus $\text{rk}(\boldsymbol{K}_2) = m - 2$. Hence, (2.10) defines an IGMRF with $\text{rk}(\boldsymbol{K}_d) = m - d$. If $|\boldsymbol{Q}|$ is required, its generalized version can be computed by the product of all non-zero eigenvalues.

The simple random walk priors can be extended in different ways, see Fahrmeir and Kneib (2011, Section 2.1.2) and the references therein. For example, support for unequally spaced locations can be achieved by weighting the precision of the conditional priors in (2.8) and (2.9) by the difference between time point $j$ and $j-1$, $\delta_j$. This leads to $\boldsymbol{K}_d = \boldsymbol{D}'_d \boldsymbol{\Delta}_d \boldsymbol{D}_d$ in (2.10) with $\boldsymbol{\Delta}_d = \text{diag}(\delta_{d+1}, \dots, \delta_m)$. A detailed discussion of continuous random walks can be found in Rue and Held (2005, Section 3.5). Further extensions include the modeling of seasonal variation (Rue and Held, 2005, Section 3.4.3) and the incorporation of local adaptivity in order to account for unsmooth elements or abrupt changes in time-series (Fahrmeir and Kneib, 2011).

### Nonlinear effects for continuous covariates

Since its introduction by Eilers and Marx (1996), penalized splines (P-splines) have been applied successfully for the approximation of smooth functions of continuous covariates in a wide range of applications. The Bayesian approach to P-splines discussed here has been proposed by Lang and Brezger (2004). A more thorough derivation can be found in Fahrmeir and Kneib (2011, Chapter 2).

By using Bayesian P-splines, the elements in $\boldsymbol{\gamma}$ are placed on a set of knots $\{\tau_1, \dots, \tau_m\}$, $m < n$, which covers the range of an explanatory variable $\boldsymbol{z} = (z_1, \dots, z_n)$'. The smooth function $f$ in (2.2) is then modeled by a linear combination of B-spline basis functions of degree $D$:

$$f(z_i) = \sum_{j=1}^{m} \gamma_j B_j^D(z_i) \tag{2.11}$$

where the basis functions can be recursively defined (de Boor, 1978) by

$$B_j^D(z_i) = \frac{z_i - \tau_j}{\tau_{j+D} - \tau_j} B_j^{D-1}(z_i) + \frac{\tau_{j+D+1} - z_i}{\tau_{j+D+1} - \tau_{j+1}} B_{j+1}^{D-1}(z_i) \tag{2.12}$$

with

$$B_j^0(z_i) = \begin{cases} 1 & \tau_j \leq z_i < \tau_{j+1}, \\ 0 & , \text{ otherwise,} \end{cases} \qquad j = 1, \dots, m-1.$$

Due to this transformation of $\boldsymbol{z}$, the corresponding $n \times m$ design matrix $\boldsymbol{Z}$ is no longer an indicator matrix. Instead, it consists of the evaluations of (2.12) at each observation $z_i$:

$$\boldsymbol{Z} = \begin{pmatrix} B_1^D(z_1) & \cdots & B_m^D(z_1) \\ \vdots & & \vdots \\ B_1^D(z_n) & \cdots & B_m^D(z_n) \end{pmatrix}. \tag{2.13}$$

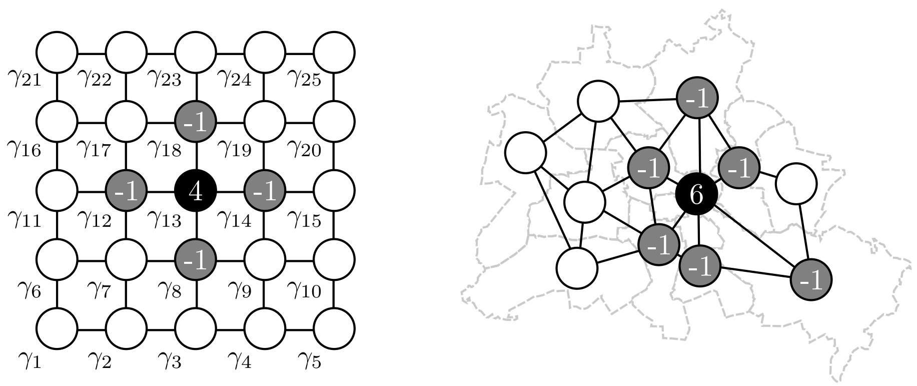
With this definition, the function evaluation in (2.11) can be compactly written in matrix notation as in (2.3).

Smoothness for $f$ is achieved by imposing similar priors on $\boldsymbol{\gamma}$ as for temporal effects, in particular the RW2 prior is used for penalizing to rough deviations between the elements of $\boldsymbol{\gamma}$. Hence, a similar graph as depicted in Figure 2.1 (b) is considered. In practice, this prior is usually combined with a moderate number of equidistant knots $m$ (20–40, Eilers and Marx, 1996) and cubic ($D = 3$) B-spline basis functions which yields attractive performance among different applications (Brezger and Lang, 2006).

While the $m \times m$ precision matrix has the same sparsity structure as a RW2 prior for temporal effects, the amount of sparseness for $\boldsymbol{Z}$ depends on the number of knots as well as on the degree of the basis functions. However, usually upper limits can be given for special cases of $D$ and $m$. For example, choosing 30 equidistant knots over the range of $\boldsymbol{z}$ and $D = 3$ usually results in a design matrix with a maximum of four non-zero elements per row. Less non-zero elements result for data points that lie on the outside of the range of $\boldsymbol{z}$. Hence, an upper bound for the amount of non-zero elements in $\boldsymbol{Z}$ for this particular case can be given as $4 \cdot n/(n \cdot m) \cdot 100\% = 4/m \cdot 100\% \approx 13\%$. Thus, the design matrix for this common specification can be considered as sparse.

### Spatial effects

GMRF priors are often used to model discrete spatial information where data is observed on a regular or irregular lattice, see Rue and Held (2005, Section 1.3) for an extensive review of applications. Examples for the occurrence of regular lattices can be found in the analysis of digital images where each observation can be assigned to one of many picture elements (pixels) that are arranged in an equidistant two-dimensional grid structure. Here, each element in $\boldsymbol{\gamma}$ is assigned to one image pixel. The corresponding graph of such a regular lattice is depicted in Figure 2.2, Panel (a), along with a visualization of the first order neighborhood for a randomly chosen node, that is, the first neighboring pixels in $x$- and



$\gamma_{21}$ $\gamma_{22}$ $\gamma_{23}$ $\gamma_{24}$ $\gamma_{25}$
$\gamma_{16}$ $\gamma_{17}$ $\gamma_{18}$ $\gamma_{19}$ $\gamma_{20}$
$\gamma_{11}$ $\gamma_{12}$ $\gamma_{13}$ $\gamma_{14}$ $\gamma_{15}$
$\gamma_{6}$ $\gamma_{7}$ $\gamma_{8}$ $\gamma_{9}$ $\gamma_{10}$
$\gamma_{1}$ $\gamma_{2}$ $\gamma_{3}$ $\gamma_{4}$ $\gamma_{5}$
-1
-1
4
-1
-1
-1
-1
-1
6
-1
-1
-1
-1

(a)

(b)

Figure 2.2 Visualization of two-dimensional undirected graphs on (a) a 5 × 5 regular lattice and (b) an irregular lattice corresponding to the districts of the city of Berlin, Germany. In both graphs, a first order neighborhood structure is imposed, that is, only nodes that share a common border are considered as neighbors. Numbers refer to the non-zero entries of the black nodes rows of (2.15).

$y$-direction are considered to be neighbors. Thus, this graph can be seen as an extension of a RW1 in two dimensions. An example for a graph on an irregular lattice is shown in Panel (b) of Figure 2.2. Here, a first order neighborhood structure is imposed over the districts of the city of Berlin, Germany, that is, districts that share a common border are considered as neighbors. Both types of lattices can be modeled by a GMRF that imposes the following conditional prior for $\gamma_j$ given all other elements in $\boldsymbol{\gamma}$:

$$\gamma_j | \boldsymbol{\gamma}_{-j}, \kappa \sim \mathrm{N}\left(\frac{1}{n_j} \sum_{l \in \mathcal{N}_j} \gamma_l, \frac{1}{n_j \kappa}\right). \tag{2.14}$$

Here, the set $\mathcal{N}_j$ includes the indices of all neighboring sites of node $j$, and $n_j$ is the number of neighbors of node $j$. The joint prior of (2.14) is a zero-mean multivariate normal with precision matrix $\boldsymbol{Q}(\kappa) = \kappa \boldsymbol{K}$, thus of form (2.5). The structure matrix has elements

$$K_{jl} = \begin{cases} n_j & \text{if } j = l, \\ -1 & \text{if } l \in \mathcal{N}_j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.15}$$

Figure 2.2 shows the non-zero entries in $\boldsymbol{K}$ for the black nodes.

As already mentioned, the neighborhood structure on the graph depicted in panel (a) of Figure 2.2 can be seen as an extension of the RW1 in two dimensions. It turns out that, for regular lattices of size $n_x$ and $n_y$, the corresponding structure matrix for this graph can be composed by the structure matrices $\boldsymbol{K}_x$ and $\boldsymbol{K}_y$ of two independent RW1 using the Kronecker sum:

$$\begin{aligned}\boldsymbol{K}_{xy} &= \boldsymbol{K}_y \oplus \boldsymbol{K}_x \\ &= \boldsymbol{K}_y \otimes \boldsymbol{I}_{n_x} + \boldsymbol{I}_{n_y} \otimes \boldsymbol{K}_x. \end{aligned} \tag{2.16}$$

Here, $\oplus$ denotes the Kronecker sum and $\otimes$ denotes the Kronecker product. Higher order neighborhood structures on regular lattices in two dimensions are discussed and compared in Schmid (2004, Section 4.3). Much interest has been paid to approaches that do not only account for dependencies along the $x$- and $y$ directions but also for dependencies along the diagonals, in particular approximations to the biharmonic differential operator as discussed in Rue and Held (2005, Section 3.4) and Fahrmeir and Kneib (2011, Section 5.3).

Of particular interest for this thesis are extensions of two-dimensional regular lattices in three dimensions. Let $\boldsymbol{K}_z$ be the structure matrix of a RW1 with length $n_z$. Then, the structure matrix of a three-dimensional random walk on a regular lattice with dimension $n_x \times n_y \times n_z$ can be obtained by

$$\boldsymbol{K}_{xyz} = \boldsymbol{K}_z \oplus \boldsymbol{K}_{xy}. \tag{2.17}$$

The design matrix $\boldsymbol{Z}$ for modeling discrete spatial information is again an indicator matrix, hence it can be seen as sparse. For irregular graphs, no general statement for the structure of the precision matrix can be given. However, this matrix can usually be considered as sparse as well if the amount of neighbors that is induced by the neighborhood order is small compared to the number of sites $m$. For regular lattices, general statements can be made for precision matrices due to its deterministic structure. For example, (2.16) has $5n_xn_y - 2(n_x + n_y)$ non-zero elements while the three-dimensional version in (2.17) has $7 \cdot n_xn_yn_z - 2 \cdot (n_xn_y + n_xn_z + n_yn_z)$ non-zero elements, compared to $(n_xn_y) \cdot (n_xn_y)$ and $(n_xn_yn_z) \cdot (n_xn_yn_z)$ elements in total, respectively. Similar to the one-dimensional random walk these matrices are invariant to the addition of a constant vector, hence, $\text{rk}(\boldsymbol{K}_{xy}) = n_xn_y - 1$ and $\text{rk}(\boldsymbol{K}_{xyz}) = n_xn_yn_z - 1$. The $n_x \cdot n_y - 1$ non-zero eigenvalues of (2.16) can be obtained from the diagonal elements of $\text{diag}(\boldsymbol{\lambda}_y) \oplus \text{diag}(\boldsymbol{\lambda}_x)$, where $\boldsymbol{\lambda}_x = (\lambda_{x,1}, \ldots, \lambda_{x,n_x})'$ and $\boldsymbol{\lambda}_y = (\lambda_{y,1}, \ldots, \lambda_{y,n_y})'$ are the vectors of eigenvalues of $\boldsymbol{K}_x$

and $\boldsymbol{K}_y$, respectively, see Besag and Higdon (1999). A similar statement can be made for (2.17).

A popular approach for modeling spatial effects has been proposed by Besag et al. (1991). Here, the spatial effect is split into two components, one unstructured effect as presented in Section 2.2.1 and one spatially structured effect. This combination is known as the Besag-York-Mollié (BYM)-model.

While GMRF priors are suitable for modeling discrete spatial information, they are not directly applicable in the case of continuous spatial information. Here, Gaussian processes or equivalently Gaussian random fields are preferred (Diggle and Ribeiro, 2007). In order to construct smooth surfaces over continuous spatial domains, also known as *kriging* (Krige, 1951), the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{Q}^{-1}$ is usually specified directly by using a certain covariance function. In contrast to the GMRF approach which models the precision matrix, this leads to a dense matrix and therefore violates one of the main requirements for the application to large-scale data, see Section 4.1.1. However, different approaches exist to make continuous spatial data feasible for GMRF priors. For example, spatial domains can be discretized into disjoint regions and treated as discrete spatial data on regular or irregular lattices. More recently, Lindgren et al. (2011) showed that an explicit link between certain Gaussian processes and GMRFs exist. They use stochastic partial differential equations in order to derive an explicit GMRF representation of Gaussian processes with Matérn covariance functions over a triangulated mesh of the spatial domain. This way it is possible to model continuous spatial data directly through the covariance matrix while still having access to computational advantages of GMRFs, see Chapter 3 and 4.

### Interaction between covariates

Varying coefficient models (VCMs, Hastie and Tibshirani, 1993) represent a popular method for modeling interactions of covariates. Here, the function $f(z_{i1})$ in (2.2) is extended to $f(z_{i1})z_{i2}$. In this context, $\boldsymbol{z}_1$ is called the effect-modifier of $\boldsymbol{z}_2$. The corresponding design matrix is obtained by multiplying each column of the design matrix of $f(\boldsymbol{z}_1)$ by $\boldsymbol{z}_2$, thus, the sparsity structure of the design and precision matrix remains identical. Note that, if $\boldsymbol{z}_1$ is categorical and modeled as an i.i.d. random effect and $\boldsymbol{z}_2$ is metric, then the extended term $f(z_{i1})z_{i2}$ coincides with the random slope effect presented in the context of unstructured random effects in Section 2.2.1. If instead, spatial dependence between the nodes in $\boldsymbol{z}_1$ is induced by an approach as given in Section 2.2.1, a spatially varying coefficient is obtained as used in Section 6.2.

If both covariates are metrical, the basis function approach for nonlinear continuous effects as presented in Section 2.2.1 can be extended to obtain two-dimensional interaction surfaces (Chen, 1993): Consider the basis function representation (2.11) for $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ that is defined over the set of knots $\boldsymbol{\tau}_1 = \{\tau_{11}, \dots, \tau_{1m_1}\}$ and $\boldsymbol{\tau}_2 = \{\tau_{21}, \dots, \tau_{2m_2}\}$, respectively. An approximation of the interaction surface $f(\boldsymbol{z}_1, \boldsymbol{z}_2)$ can then be obtained by placing the elements in $\boldsymbol{\gamma}$ on the grid that is spanned by $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ and by computing the design matrix $\boldsymbol{Z}$ by the (Kronecker) tensor product of the B-spline design matrices $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$, thus $f(\boldsymbol{z}_1, \boldsymbol{z}_2) = \boldsymbol{Z}\boldsymbol{\gamma}$ with

$$f(z_{i1}, z_{i2}) = \sum_{j=1}^{m_1} \sum_{l=1}^{m_2} \gamma_{j,l} B_j^D(z_{i1}) B_l^D(z_{i2}).$$

Since the elements of $\boldsymbol{\gamma}$ are spatially aligned over a regular lattice, the same smoothing priors as for discrete spatial effects can be chosen. In order to obtain fairly smooth surfaces, GMRF priors that account for higher neighborhood orders are of particular interest, such as approximations to the biharmonic differential operator mentioned in Section 2.2.1 and discussed in more detail in Fahrmeir and Kneib (2011, Section 5.3). These authors also discuss an alternative definition of interaction surfaces based on radial bases along with appropriate penalties.

The sparsity structure of the precision matrix can be considered as outlined in Section 2.2.1. It turns out that the design matrix of the tensor product P-splines is sparse as well, since the Kronecker product of two sparse matrices remains sparse. This is obvious as the Kronecker product between two sparse matrices $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$ multiplies each element in $\boldsymbol{Z}_1$ with the sparse matrix $\boldsymbol{Z}_2$.

### Spatio-temporal effects

The types of effects introduced so far can be used to model temporal and spatial information independently by main effects, i.e. the linear predictor is $\eta = \dots f_{\text{temporal}} + f_{\text{spatial}} + \dots$, where $f_{\text{temporal}}$ may be modeled by a RW1 or RW2 prior as presented in Section 2.2.1 and $f_{\text{spatial}}$ may be modeled as in Section 2.2.1. In order to obtain non-separable space-time models, Knorr-Held (2000) discuss four types of interactions of temporal and spatial main effects with an increasing amount of complexity. The most simplistic model includes two BYM-like effects for temporal and spatial information, i.e. the linear predictor includes one unstructured and one structured effect for time as well as one unstructured and one structured effect for space. Different interaction types are obtained by all possible

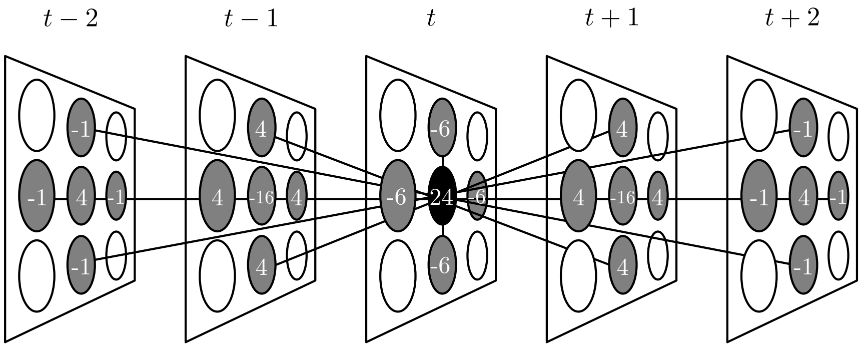


Figure 2.3 Visualization of an undirected graph with spatio-temporal dependency. Neighborhood structure corresponds to a Type IV interaction according to Knorr-Held (2000) between a RW2 in temporal and a RW1 in spatial dimension. Values refer to the row of the black node in the corresponding structure matrix.

combinations of structured and unstructured penalty matrices for the temporal and spatial effects. Structure matrices are combined using the Kronecker product as proposed by Clayton (1996):

$$\boldsymbol{K} = \boldsymbol{K}_{\text{temporal}} \otimes \boldsymbol{K}_{\text{spatial}}. \tag{2.18}$$

The first interaction (Type I interaction) combines the unstructured effects and yields *a priori* independent regression coefficients that are completely unstructured in space and time. In contrast, the most advanced interaction (Type IV interaction) combines the structured main effects and therefore produces regression coefficients that are *a priori* dependent in space and time. As an example for this type of interaction, consider the undirected graph that is shown in Figure 2.3. Here, the neighborhood structure is the result of the combination of a RW2 in temporal and a RW1 in spatial dimension.

Kronecker product penalties are appealing as they lead to precision matrices that depend on a single, global precision parameter only. This simplifies inference on these hyperparameters dramatically, see Section 3.1.2. However, that means that the same parameter is responsible for smoothing in time and space. For data where the spatial dimension dominates the temporal, usage of a global precision parameter may lead to an underestimation of temporal smoothness. Different approaches have been discussed in order to overcome this problem. For example, Gössl et al. (2001) introduced pixel-wise precisions

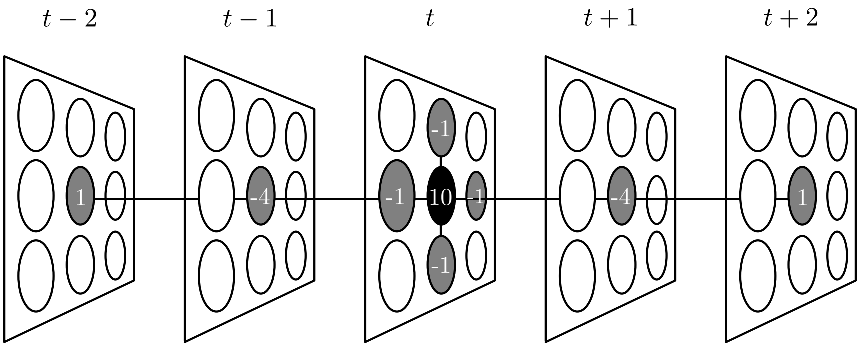


Figure 2.4 Visualization of an undirected graph with spatio-temporal dependency. Neighborhood structure is imposed by a Kronecker sum of a RW2 in temporal and a RW1 in spatial dimension. Values refer to the row of the black node in the corresponding structure matrix.

in the context of modeling fMRI time series. In addition, they expanded the Kronecker product penalty by a temporal main penalty-effect:

$$\begin{aligned}\boldsymbol{Q} &= \boldsymbol{Q}_{\text{spatial}} \otimes \boldsymbol{K}_{\text{temporal}} + \boldsymbol{\Lambda} \otimes \boldsymbol{K}_{\text{temporal}} \\ &= (\boldsymbol{Q}_{\text{spatial}} + \boldsymbol{\Lambda}) \otimes \boldsymbol{K}_{\text{temporal}}.\end{aligned}$$

Here, $\boldsymbol{K}_{\text{temporal}}$ is a RW2 structure matrix, $\boldsymbol{\Lambda}$ is the diagonal matrix of pixel-wise precisions and $\boldsymbol{Q}_{\text{spatial}}$ is given by

$$Q_{\text{spatial},jl} = \begin{cases} \sum_{l \in \mathcal{N}_j} (\kappa_j + \kappa_l) & \text{if } j = l, \\ -(\kappa_j + \kappa_l) & \text{if } l \in \mathcal{N}_j, \\ 0 & \text{otherwise.} \end{cases}$$

A further alternative can be derived by considering the Kronecker sum between the temporal and spatial structure matrices instead of the Kronecker product:

$$\boldsymbol{K} = \boldsymbol{K}_{\text{temporal}} \otimes \boldsymbol{I}_{\text{spatial}} + \boldsymbol{I}_{\text{temporal}} \otimes \boldsymbol{K}_{\text{spatial}}.$$

The neighborhood structure that is induced by applying this formula to a RW2 in temporal and a RW1 in spatial dimension is depicted in Figure 2.4. It can be seen that, in contrast to Figure 2.3, only the direct temporal neighbors are used from neighboring time points. The Kronecker sum approach can easily be extended to establish different smoothing along space and time. For example, by using different precision parameters for these two dimensions one obtains the following precision matrix:

$$\boldsymbol{Q} = \kappa_1 \boldsymbol{K}_{\text{temporal}} \otimes \boldsymbol{I}_{\text{spatial}} + \kappa_2 \boldsymbol{I}_{\text{temporal}} \otimes \boldsymbol{K}_{\text{spatial}}.$$

Rue and Held (2005, page 107) present a similar approach for modeling spatial data on a regular lattice with different smoothing in $x$- and $y$-directions.

In accordance to design matrices of temporal and spatial main effects, the design matrices for spatio-temporal effects are indicator matrices and therefore sparse, although the dimension is increased to $n \times (m_{\text{spatial}} \cdot m_{\text{temporal}})$. Precision matrices can be considered as sparse as well, however, different rank deficiencies arise for the different approaches. The rank of Kronecker product penalties can easily be calculated by basic rules for Kronecker products (Harville, 1997), i.e. $\text{rk}(\boldsymbol{K}) = \text{rk}(\boldsymbol{K}_{\text{temporal}}) \cdot \text{rk}(\boldsymbol{K}_{\text{spatial}})$. For the Kronecker sum with specification as in Figure 2.4 one has $\text{rk}(\boldsymbol{K}) = m_{\text{temporal}} \cdot m_{\text{spatial}} - 2$.

### 2.2.2 Hyperparameters

If the linear predictor contains more than fixed effects, precision parameters $\kappa_k, k = 1, \dots, p$, need to be estimated as well. Hence, in order to complete the model formulation, prior distributions for these hyperparameters as well as for an additional dispersion parameter need to be set up.

#### Precision parameters of GMRF priors

For precision parameters $\kappa_k$, $k = 1, \dots, p$, it is common practice to choose independent Gamma distributions with shape and rate parameters $a_k$ and $b_k$, respectively:

$$p(\kappa_k) = \frac{b_k^{a_k}}{\Gamma(a_k)} \kappa_k^{a_k - 1} \exp(-b_k \kappa_k)$$

Note that this choice is equivalent to an Inverse Gamma prior on the variance parameter $\nu_k = \kappa_k^{-1}$ with shape and rate $a_k$ and $b_k$, respectively,

$$p(\nu_k) = \frac{b_k^{a_k}}{\Gamma(a_k)} \nu_k^{-(a_k+1)} \exp(-b_k/\nu_k),$$

or a log-Gamma prior on $\tau_k = \log \kappa_k$

$$p(\tau_k) = \frac{b_k^{a_k}}{\Gamma(a_k)} \exp(a_k \tau_k - b_k \exp(\tau_k)),$$

which my be preferred by some authors.

The Gamma prior is primary chosen for the following reasons: First, for the case $\boldsymbol{Q}_k = \kappa_k \boldsymbol{K}_k$ it turns out that the Gamma distribution is a conjugate family for the GMRF prior on $\boldsymbol{\gamma}_k$, that is, the full conditional of $\kappa_k$ is again a Gamma distribution, see Chapter 3 for details. Second, by choosing different values for $a_k$ and $b_k$, this distribution is able to cover a wide range of different prior beliefs over a strictly positive domain. For example, a weakly informative prior can be obtained by choosing small values for $a_k$ and $b_k$, e.g. $a_k = b_k = 0.001$, or, alternatively, $a_k = 1$ and $b_k$ small (Brezger and Lang, 2006), whereas smoothing can be increased by choosing larger values for $a_k$ and $b_k$. Gelman (2006) noted that for the choice of $a_k = b_k = \epsilon$ with $\epsilon \to 0$ inference can be quite sensitive with respect to $\epsilon$ if $\kappa_k$ is small. Instead, he suggests to use a non-informative uniform prior on $\kappa_k^{-1}$. However, the potential negative effect of the Gamma prior on the final result can usually be minimized by performing sensitivity analyses with respect to $a_k$ and $b_k$.

### Dispersion parameter

Besides the fact that the prior distribution should support strictly positive real numbers only, no general recommendation for an appropriate prior of an additional dispersion parameter $\phi$ of the likelihood can be given. However, the Gamma distribution with parameters $a_\phi$ and $b_\phi$ is a popular choice, especially for Gaussian response models since the Gamma distribution is the conjugate prior for the precision parameter of a normal likelihood. Further examples that use the Gamma distribution as a prior for $\phi$ can be found for Gamma distributed response variables (Brezger and Lang, 2006, Section 4.2) and for Beta models, see Section 3.1.3.

## 2.3 Chapter summary

In this chapter, all components that are necessary to formulate STAR models have been introduced: the observation model and prior distributions for all unknown parameters. The observation model has been formulated relative vague in order to be as generic as possible.

Furthermore, it has been shown that GMRFs provide a flexible framework with which one is able to build appropriate priors for fixed and unstructured random effects, temporal effects and nonlinear effects for continuous covariates, spatial effects and interactions as well as spatio-temporal effects. Finally, prior specification for hyperparameters has been outlined.

# 3 Inference

Having introduced the basic model formulation for STAR models in Chapter 2, this chapter provides an overview of popular methods that are used to fit these models. Within the Bayesian approach considered here, inference relies solely on the joint posterior. Given the specifications in Chapter 2 this joint posterior can be written as

$$\begin{aligned} p(\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_p, \kappa_1, \dots, \kappa_p, \phi | \boldsymbol{y}) \propto & \prod_{i=1}^{n} p(y_i | \boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_p, \kappa_1, \dots, \kappa_p, \phi) \\ & \times \prod_{k=1}^{p} |\boldsymbol{Q}_k(\kappa_k)|^{1/2} \exp\left(-\frac{1}{2}\boldsymbol{\gamma}_k' \boldsymbol{Q}_k(\kappa_k) \boldsymbol{\gamma}_k\right) \\ & \times \prod_{k=1}^{p} \kappa_k^{a_k - 1} \exp(-b_k \kappa_k) \\ & \times \phi^{a_\phi - 1} \exp(-b_\phi \phi). \end{aligned} \tag{3.1}$$

Here, Gamma priors have been imposed on all precision and dispersion parameters. For the exploration of this posterior three different approaches that can be found in the statistical literature are discussed in this chapter. First, fully Bayes inference based on MCMC is presented in Section 3.1 which is followed by an empirical Bayes approach in Section 3.2. Finally, Section 3.3 presents an approximate Bayes approach.

By all means, this chapter does not claim to give a complete overview over the methods that can be used to analyze joint posterior (3.1). In particular, two approaches that are quite popular in the machine learning literature, namely variational Bayes (Bishop, 2006) and expectation-propagation (Minka, 2001), are not presented. For a discussion on the performance of these methods with respect to latent Gaussian models see the overview in the introduction of Rue et al. (2009).

The MCMC approach given in the following section is also discussed in shortened form in Schmidt et al. (2017, Section 2.3).

## 3.1 MCMC based inference for STAR models

Markov chain Monte Carlo (MCMC) simulation, pioneered by the work of Metropolis et al. (1953) and Hastings (1970), has been used in statistical physics long before it was recognized in mainstream statistics (Robert and Casella, 2011). It was first used in statistics for the analysis of digital images from which the Gibbs sampler (Geman and Geman, 1984) emerged. This procedure helped solving many other complex problems in applied statistics for which no or only unsatisfactory solutions existed. The analysis of spatial data is an example of such situations. Besag et al. (1991) pointed out that these situations can be formulated as image restoration problems and, therefore, be solved by a Gibbs Sampler. This, and the broad availability of fast computational equipment, led to a revolution in Bayesian statistics. Since then, a wide range of different MCMC samplers have been emerged, each with its own advantages and disadvantages.

In Bayesian statistics, MCMC simulation is used to simulate samples from the joint posterior distribution which are then used to summarize this distribution in any possible way. In its easiest form, i.e. the Gibbs sampler, a realization of each unknown parameter is obtained one by one by sampling directly from its full conditional posterior distribution. If direct sampling from the full conditional distribution is not possible, a Metropolis-Hastings step can be performed; also referred to as Metropolis-within-Gibbs. The resulting sampling scheme is quite general and applicable to a wide range of problems. The MCMC algorithm for STAR models presented here is the basic version of the one presented in Fahrmeir et al. (2004). Here, a Metropolis-Hastings step is performed for sampling regression coefficients $\boldsymbol{\gamma}_k, k = 1, \dots, p$, while precision parameters are updated directly from their corresponding full conditionals. The details on these steps are given next.

### 3.1.1 Regression coefficients

For Gaussian response models the full conditionals of the regression coefficients can be derived in closed form and a Gibbs sampler can easily be set up. The framework of auxiliary variable models (Rue and Held, 2005, Chapter 4.3) allows to apply this Gibbs sampler to some non-Gaussian likelihoods, in particular for the Student-$t$ and Laplace distribution (Andrews and Mallows, 1974) as well as the Binomial distribution with probit (Albert and Chib, 1993) and logit link (Holmes and Held, 2006). In the latter work, extensions to multinomial regression models are also presented. For many other non-Gaussian models, however, the full conditionals of the regression coefficients are no longer available in closed form. Therefore, a Metropolis-Hastings step needs to be included inside the Gibbs sampler in order to sample from the correct full conditionals. In this section, it is shown how an appropriate proposal for the Metropolis-Hastings step can be constructed, first for general likelihoods, then the case of exponential families is examined more closely.

### GMRF proposal

For general likelihoods a proposal density for $\boldsymbol{\gamma}_k$ can be obtained by an approximation of the likelihood similar to the GMRF approximation given in Rue (2001). The idea behind this is to match the mode of the likelihood and its corresponding curvature at the mode in order to obtain a simpler and more generic functional form that can easily be combined with the GMRF prior. The basis for the GMRF approximation is a quadratic Taylor expansion of the log-likelihood $l(\boldsymbol{\gamma}_k) = \sum_{i=1}^{n} \log p(y_i|\boldsymbol{\gamma}_k)$ around the current state $\boldsymbol{\gamma}_k^c$ which can be written as

$$l(\boldsymbol{\gamma}_k) \approx a_k^c + (\boldsymbol{b}_k^c)'\boldsymbol{\gamma}_k - \frac{1}{2}\boldsymbol{\gamma}_k'\boldsymbol{C}_k^c\boldsymbol{\gamma}_k \tag{3.2}$$

with coefficients

$$\begin{aligned} a_k^c &= l(\boldsymbol{\gamma}_k^c) - (\boldsymbol{\gamma}_k^c)'\frac{\partial l(\boldsymbol{\gamma}_k^c)}{\partial \boldsymbol{\gamma}_k} + \frac{1}{2}(\boldsymbol{\gamma}_k^c)'\frac{\partial^2 l(\boldsymbol{\gamma}_k^c)}{\partial \boldsymbol{\gamma}_k' \partial \boldsymbol{\gamma}_k}\boldsymbol{\gamma}_k^c \\ \boldsymbol{b}_k^c &= \frac{\partial l(\boldsymbol{\gamma}_k^c)}{\partial \boldsymbol{\gamma}_k} + \boldsymbol{C}_k^c\boldsymbol{\gamma}_k^c \qquad (3.3) \\ \boldsymbol{C}_k^c &= -\frac{\partial^2 l(\boldsymbol{\gamma}_k^c)}{\partial \boldsymbol{\gamma}_k' \partial \boldsymbol{\gamma}_k}. \qquad (3.4) \end{aligned}$$

Since $a_k^c$ in (3.2) does not depend on $\boldsymbol{\gamma}_k$, it can be neglected. For GLMs the derivatives in the above formulas can be computed explicitly, see below. In other cases, these derivatives may be approximated using numerical differentiation techniques (Rue and Held, 2005, page 171).

By using (3.2) as an approximation for the log-likelihood the full conditional $p(\boldsymbol{\gamma}_k|\boldsymbol{y}, \kappa_k)$ can be written as

$$\begin{aligned} p(\boldsymbol{\gamma}_k|\boldsymbol{y}, \kappa_k) &\propto p(\boldsymbol{y}|\boldsymbol{\gamma}_k)p(\boldsymbol{\gamma}_k|\kappa_k) \\ &\propto \exp\left(-\frac{1}{2}\boldsymbol{\gamma}_k'\boldsymbol{Q}_k\boldsymbol{\gamma}_k + \sum_{i=1}^{n}\log p(y_i|\boldsymbol{\gamma}_k)\right) \end{aligned}$$

$$\begin{aligned}
&\approx \exp\left(-\frac{1}{2}\boldsymbol{\gamma}_k'\boldsymbol{Q}_k\boldsymbol{\gamma}_k + a_k^c + (\boldsymbol{b}_k^c)'\boldsymbol{\gamma}_k - \frac{1}{2}\boldsymbol{\gamma}_k'\boldsymbol{C}_k^c\boldsymbol{\gamma}_k\right)\\
&\propto \exp\left(-\frac{1}{2}\boldsymbol{\gamma}_k'(\boldsymbol{Q}_k + \boldsymbol{C}_k^c)\boldsymbol{\gamma}_k + (\boldsymbol{b}_k^c)'\boldsymbol{\gamma}_k\right). \qquad (3.5)
\end{aligned}$$

This corresponds to the core of a multivariate normal distribution, thus, the proposal distribution for $\boldsymbol{\gamma}_k$ based on the GMRF approximation has the form

$$\boldsymbol{\gamma}_k^p|\cdot \sim \mathrm{N}(\tilde{\boldsymbol{\mu}}_k^c, \widetilde{\boldsymbol{Q}}_k^c). \qquad (3.6)$$

Instead of the more familiar notation using the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{Q}^{-1}$, $\mathrm{N}(\cdot, \boldsymbol{Q})$ here refers to a multivariate normal distribution with precision $\boldsymbol{Q}$. The precision matrix of (3.6) is given by

$$\widetilde{\boldsymbol{Q}}_k^c = \boldsymbol{Q}_k + \boldsymbol{C}_k^c \qquad (3.7)$$

and the mean $\tilde{\boldsymbol{\mu}}_k^c$ is the solution of the linear system

$$\widetilde{\boldsymbol{Q}}_k^c \tilde{\boldsymbol{\mu}}_k^c = \boldsymbol{b}_k^c. \qquad (3.8)$$

Sampling a proposal $\boldsymbol{\gamma}_k^p$ from this distribution requires the evaluation of $\tilde{\boldsymbol{\mu}}_k^c$ and $\widetilde{\boldsymbol{Q}}_k^c$ at the current state $\boldsymbol{\gamma}_k^c$. The proposal is accepted with probability

$$\alpha(\boldsymbol{\gamma}_k^c, \boldsymbol{\gamma}_k^p) = \min\left\{1, \frac{p(\boldsymbol{y}|\boldsymbol{\gamma}_k^p)p(\boldsymbol{\gamma}_k^p|\kappa_k)\varphi(\boldsymbol{\gamma}_k^c|\tilde{\boldsymbol{\mu}}_k^p, \widetilde{\boldsymbol{Q}}_k^p)}{p(\boldsymbol{y}|\boldsymbol{\gamma}_k^c)p(\boldsymbol{\gamma}_k^c|\kappa_k)\varphi(\boldsymbol{\gamma}_k^p|\tilde{\boldsymbol{\mu}}_k^c, \widetilde{\boldsymbol{Q}}_k^c)}\right\} \qquad (3.9)$$

where $\varphi(\cdot; \boldsymbol{\mu}, \boldsymbol{Q})$ represents the density of a multivariate normal distributed random variable with mean $\boldsymbol{\mu}$ and precision matrix $\boldsymbol{Q}$. Note that both, $\tilde{\boldsymbol{\mu}}_k^c$ and $\widetilde{\boldsymbol{Q}}_k^c$ depend on the current state of the chain. Therefore, in order to obtain the acceptance probability the normalizing constant of $\varphi$ needs to be calculated which requires the computation of the log-determinant of $\widetilde{\boldsymbol{Q}}_k^c$. In addition, (3.7) and (3.8) need to be re-evaluated given the proposal $\boldsymbol{\gamma}_k^p$.

In case of low acceptance rates the GMRF approximation can be further improved by repeating the Tylor series expansion of the likelihood around the mean of the proposal distribution, $\tilde{\boldsymbol{\mu}}_k^c$. This can be iterated until convergence or just until the desired acceptance rate has been reached (Rue and Held, 2005, p. 172).

### IWLS proposal

More insight into the GMRF approximation can be obtained by considering the special case of GLMs, that is, for likelihoods that are exponential families. In this case, the GMRF approximation coincides with the iterated weighted least squares (IWLS) proposal (Gamerman, 1997). This can be seen by noting that the first derivative of the likelihood with respect to $\boldsymbol{\gamma}_k$, i.e. the score vector, can be written as

$$\boldsymbol{s}(\boldsymbol{\gamma}_k) = \boldsymbol{Z}_k' \boldsymbol{D} \boldsymbol{V}^{-1} (\boldsymbol{y} - \boldsymbol{\mu})$$

and the negative second derivation, i.e. the Fisher information, as

$$\boldsymbol{F}(\boldsymbol{\gamma}_k) = \boldsymbol{Z}_k' \boldsymbol{W} \boldsymbol{Z}_k.$$

Here, $\boldsymbol{D}$ is a diagonal matrix with entries $\partial h(\eta_i)/\partial\eta$, $i = 1, \dots, n$, $\boldsymbol{V}$ is a diagonal matrix with entries $\phi v(\mu_i)/\omega_i$, $i = 1, \dots, n$, where $v(\mu_i)$ and $\omega_i$ are the variance function and weights corresponding to the specific exponential family, respectively, and $\boldsymbol{W} = \boldsymbol{D}\boldsymbol{V}^{-1}\boldsymbol{D}$. From (3.3) and (3.4) it follows directly that $\boldsymbol{C}_k^c = \boldsymbol{F}(\boldsymbol{\gamma}_k)$ and

$$\begin{aligned}
\boldsymbol{b}_k^c &= \boldsymbol{s}(\boldsymbol{\gamma}_k^c) + \boldsymbol{F}(\boldsymbol{\gamma}_k^c)\boldsymbol{\gamma}_k^c \\
&= \boldsymbol{Z}_k' \boldsymbol{D}^c (\boldsymbol{V}^c)^{-1} (\boldsymbol{y} - \boldsymbol{\mu}^c) + \boldsymbol{Z}_k' \boldsymbol{W}^c \boldsymbol{Z}_k \boldsymbol{\gamma}_k^c \\
&= \boldsymbol{Z}_k' \boldsymbol{D}^c (\boldsymbol{V}^c)^{-1} \boldsymbol{D}^c ((\boldsymbol{D}^c)^{-1} (\boldsymbol{y} - \boldsymbol{\mu}^c) + \boldsymbol{Z}_k \boldsymbol{\gamma}_k^c) \\
&= \boldsymbol{Z}_k' \boldsymbol{W}^c (\tilde{\boldsymbol{y}}_k^c - \boldsymbol{\eta}_{-k}^c)
\end{aligned}$$

with working observations $\tilde{\boldsymbol{y}} = \boldsymbol{\eta} + \boldsymbol{D}^{-1}(\boldsymbol{y} - \boldsymbol{\mu})$ and $\boldsymbol{\eta}_{-k}$ describing the linear predictor without the $k$th term. Thus, the parameters of the proposal obtained by the GMRF approximation can be expressed as

$$\widetilde{\boldsymbol{Q}}_k^c = \boldsymbol{Z}_k' \boldsymbol{W}^c \boldsymbol{Z}_k + \boldsymbol{Q}_k \tag{3.10}$$

and

$$\widetilde{\boldsymbol{Q}}_k^c \tilde{\boldsymbol{\mu}}_k^c = \boldsymbol{Z}_k' \boldsymbol{W}^c (\tilde{\boldsymbol{y}}^c - \boldsymbol{\eta}_{-k}^c). \tag{3.11}$$

This formulation coincides directly with the IWLS proposal suggested by Gamerman (1997) for random effects in the context of generalized linear mixed models. Note that $\boldsymbol{D}$, $\boldsymbol{V}$ and therefore $\boldsymbol{W}$ and $\tilde{\boldsymbol{y}}$ depend on the current state of the chain.

### 3.1.2 Precision parameters

If the precision matrix of $\boldsymbol{\gamma}_k$ depends on one precision parameter only through $\boldsymbol{Q}(\kappa_k) = \kappa_k \boldsymbol{K}_k$ and if a Gamma distribution has been chosen as a prior for $\kappa_k$, then its full conditional is given by

$$\begin{aligned} p(\kappa_k|\cdot) &\propto p(\boldsymbol{\gamma}_k|\kappa_k)p(\kappa_k) \\ &\propto \kappa_k^{\mathrm{rk}(\boldsymbol{K}_k)/2} \exp\left(-\frac{\kappa_k}{2}\boldsymbol{\gamma}_k'\boldsymbol{K}_k\boldsymbol{\gamma}_k\right) \kappa_k^{a_k-1} \exp(-b_k\kappa_k) \\ &= \kappa_k^{a_k+\mathrm{rk}(\boldsymbol{K}_k)/2-1} \exp\left(-(b_k + \boldsymbol{\gamma}_k'\boldsymbol{K}_k\boldsymbol{\gamma}_k/2)\kappa_k\right). \end{aligned}$$

Thus, in this case, the full conditionals of these hyperparameters are again Gamma distributions with updated parameters $\tilde{a}_k = a_k + \mathrm{rk}(\boldsymbol{K}_k)/2$ and $\tilde{b}_k = b_k + \boldsymbol{\gamma}_k'\boldsymbol{K}_k\boldsymbol{\gamma}_k/2$.

### 3.1.3 Dispersion parameters

Similar to its prior specification in Chapter 2 no general advice for the full conditional posterior distribution of an additional dispersion parameter $\phi$ can be given. However, in the following two special cases are examined in more detail.

#### Gaussian distributed response

For Gaussian distributed response variables the inverse variance, i.e. the precision is considered as the dispersion parameter $\phi$. Since the Gamma distribution is conjugate to the Gaussian likelihood the full conditional of $\phi$ can be derived in closed form in this case:

$$\begin{aligned} p(\phi|\cdot) &\propto p(\boldsymbol{y}|\boldsymbol{\theta})p(\phi) \\ &\propto \phi^{(n-1)/2} \exp\left(-\phi(\boldsymbol{y}-\boldsymbol{\eta})'(\boldsymbol{y}-\boldsymbol{\eta})/2\right) \phi^{a_\phi-1} \exp(-b_\phi\phi) \\ &\propto \phi^{a_\phi+(n-1)/2-1} \exp\left(-(b_\phi + (\boldsymbol{y}-\boldsymbol{\eta})'(\boldsymbol{y}-\boldsymbol{\eta})/2)\phi\right). \end{aligned}$$

Hence, the full conditional is again a Gamma distribution with updated parameters $\tilde{a}_\phi = a_\phi + (n-1)/2$ and $\tilde{b}_\phi = b_\phi + (\boldsymbol{y}-\boldsymbol{\eta})'(\boldsymbol{y}-\boldsymbol{\eta})/2$.

### Beta distributed response

An appropriate distribution for modeling proportions, i.e. response variables $y_i$ for which $0 < y_i < 1$ holds, is the Beta distribution with parameters $\alpha > 0$ und $\beta > 0$. For the use of regression a common reparametrization can be obtained by defining $\mu = \alpha/(\alpha + \beta)$ and $\phi = \alpha + \beta$, that is, $\alpha = \mu\phi$ and $\beta = (1-\mu)\phi$ (Ferrari and Cribari-Neto, 2004). This yields

$$\mathrm{E}(y_i) = \mu_i \quad \text{and} \quad \mathrm{Var}(y_i) = \frac{\mu_i(1-\mu_i)}{1+\phi}$$

and the log likelihood can be written as

$$\begin{aligned} l_i(\phi) \propto & \log\Gamma(\phi) - \log\Gamma(\mu_i\phi) - \log\Gamma((1-\mu_i)\phi) + (\mu_i\phi - 1)\log y_i \\ & + ((1-\mu_i)\phi - 1)\log(1-y_i). \end{aligned}$$

If, again, a Gamma distribution with fixed parameters $a_\phi$ and $b_\phi$ is chosen as a prior for $\phi$ the full conditional is given by

$$\begin{aligned} p(\phi|\cdot) &\propto p(\phi)p(\boldsymbol{y}|\phi) \\ &\propto \phi^{a_\phi - 1}\exp(-b_\phi\phi)\prod_{i=1}^{n} p(y_i|\phi) \\ &= \phi^{a_\phi - 1}\exp\left(-b_\phi\phi + \sum l_i(y_i|\phi)\right) \end{aligned}$$

Since this is not a known distribution an additional Metropolis-Hastings-step needs to be included in the MCMC algorithm. A proposal can be obtained by approximating the log likelihood by a second order Taylor expansion around the current state of the chain $\phi^c$:

$$l_i(y_i|\phi) \approx \theta_1^c + \theta_2^c\phi - \frac{1}{2}\theta_3^c\phi^2$$

with coefficients $\theta_1^c, \theta_2^c$ and $\theta_3^c$ similar defined as $a_k, \boldsymbol{b}_k$ and $\boldsymbol{C}_k$ in Section 3.1.1. The first derivative with respect to $\phi$ is given by

$$\frac{\partial l_i(\phi)}{\partial\phi} \propto \psi(\phi) - \psi(\mu_i\phi)\mu_i - \psi((1-\mu_i)\phi)(1-\mu_i) + \mu_i\log y_i + (1-\mu_i)\log(1-y_i)$$

and the second by

$$\frac{\partial^2 l_i(\phi)}{\partial^2 \phi^2} \propto \psi'(\phi) - \psi'(\mu_i \phi)\mu_i^2 - \psi'((1-\mu_i)\phi)(1-\mu_i)^2.$$

Here, $\psi$ and $\psi'$ denote the digamma and trigamma function, respectively. With these components the full conditional can be approximated as

$$\begin{aligned} p(\phi|\cdot) &\approx \phi^{a_\phi - 1} \exp\left(-b_\phi \phi + \theta_2^c \phi - \frac{1}{2}\theta_3^c \phi^2\right) \\ &= \phi^{a_\phi - 1} \exp\left(-\frac{1}{2}\theta_3^c \phi^2 + (\theta_2^c - b_\phi)\phi\right). \end{aligned}$$

For the special but quite common case of $a_\phi = 1$ this distribution reduces to a normal distribution with mean $\tilde{\mu}_\phi^c = (\theta_2^c - b_\phi)/\theta_3^c$ and variance $(\sigma_\phi^2)^c = 1/\theta_3^c$, thus, a distribution that can easily be used as a proposal distribution within a Metropolis-Hastings-step. However, care must be taken with respect to the domain of $\phi$: If $\phi$ is small the proposal density may cover parts of the negative real line, although $\phi$ must, by definition, be strictly positive.

In order to obtain the acceptance probability the proposal $\phi^p$ is used to calculate $\theta_2^p$ and $\theta_3^p$ and thus $\tilde{\mu}_\phi^p$ and $(\sigma_\phi^2)^p$. The acceptance probability is then given by:

$$\alpha(\phi^c, \phi^p) = \min\left\{1, \frac{p(\boldsymbol{y}|\phi^p)p(\phi^p)\varphi(\phi^c|\tilde{\mu}_\phi^p, (\sigma_\phi^2)^p)}{p(\boldsymbol{y}|\phi^c)p(\phi^c)\varphi(\phi^p|\tilde{\mu}_\phi^c, (\sigma_\phi^2)^c)}\right\}.$$

### 3.1.4 Additional considerations

#### Initialization

Regression coefficients can be initialized by iteratively computing the mode of the proposal distribution (3.6) for fixed values of the hyperparameters. This is similar to the initialization procedure of Brezger and Lang (2006), who set $\kappa_k = 0.1$ and compute the mode via back-fitting within Fisher scoring. As explained below, for convergence assessment it may be necessary to generate multiple chains with overdispersed starting values. This can be achieved, for example, by sampling $\log \kappa_k$ uniformly in a pre-chosen interval.

### Convergence assessment

Before interpreting MCMC samples as realizations from their joint posterior distribution it is necessary to assess weather the constructed Markov chain has converged to its stationary distribution. Different strategies for convergence assessment have been proposed over the years. In practice, visual inspection of trace plots is usually combined with convergence diagnostics. The latter consist of summary statistics or graphical methods that try to evaluate the quality of Markov chains with respect to convergence and mixing. See Cowles and Carlin (1996) for an extensive review and comparison of different approaches.

One convergence diagnostic which is extensively used in practice is Gelman and Rubin's potential scale reduction factor (Gelman and Rubin, 1992). This method tries to asses both, mixing and convergence by comparing the within- and between-sequence variances of multiple MCMC chains in an ANOVA-like fashion. In a revised version (Gelman et al., 2014, Section 11.4), the potential scale reduction factor for $m$ chains with length $T$ is given by

$$\widehat{R} = \sqrt{\frac{\frac{T-1}{T}W + \frac{1}{T}B}{W}}$$

with between- and within-sequence variances

$$B = \frac{T}{m-1}\sum_{j=1}^{m}(\bar{x}_{(j)} - \bar{x})^2$$
$$W = \frac{1}{m}\sum_{j=1}^{m} s^2_{(j)}.$$

Here, $x$ is the sampled parameter, $\bar{x}_{(j)}$ and $s^2_{(j)}$ the mean and variance of the $j$th chain, respectively, and $\bar{x} = \sum_j \bar{x}_{(j)}$ the overall mean. Values of $\widehat{R}$ far above one indicate that the scale of the sampled distribution of $x$ can be reduced by further sampling. Values close to one, on the other hand, may be interpreted as sufficient convergence to the stationary distribution. In practice, $\widehat{R}$ is calculated for each parameter independently. The main advantage of the potential scale reduction factor is that it only requires the first two moments of each chain. This will be helpfull when considering large-scale data, see Section 4.2.5.

### Sampling under linear constraints

For a given sample $\boldsymbol{\gamma}_k^*$ of (3.6) it may be required to take linear constraints of the form $\boldsymbol{A}\boldsymbol{\gamma}_k^* = \boldsymbol{b}$ into account. Here, $\boldsymbol{A}$ and $\boldsymbol{b}$ are of dimension $r \times m_k$ and $r \times 1$, respectively, where $r$ corresponds to the number of constraints. An example which is often used in practice are sum-to-zero constraints that may be helpful to ensure identifiability over all regression coefficients. In this case, $\boldsymbol{A}$ would be a $1 \times m_k$ row vector of all ones and $\boldsymbol{b} = 0$. Such constraints can easily be considered within MCMC frameworks by conditioning by Kriging (Rue, 2001) in which

$$\widetilde{\boldsymbol{Q}}_k^{-1}\boldsymbol{A}'(\boldsymbol{A}\widetilde{\boldsymbol{Q}}_k^{-1}\boldsymbol{A}')^{-1}(\boldsymbol{A}\boldsymbol{\gamma}_k^* - \boldsymbol{b}) \tag{3.12}$$

is subtracted from the unconditioned sample $\boldsymbol{\gamma}_k^*$. In addition to solving (3.11) and sampling from (3.6) this requires to solve the linear systems $\widetilde{\boldsymbol{Q}}_k\boldsymbol{V} = \boldsymbol{A}'$.

### Block updating of parameters

Due to computational limitations single site updating schemes were the most common sampling strategies for MCMC algorithms in early days (Gilks et al., 1996). Here, all parameters, even the single components in $\boldsymbol{\gamma}_k$, are updated one by one according to their full conditionals giving all other parameters. While this strategy comes with low computational requirements it was recognized early that the resulting Markov chains can suffer from slow mixing and poor convergence, especially when components are highly correlated (Gilks and Roberts, 1996). Block-updating schemes provide an alternative in such situations. Here, multiple parameters are collected in blocks and updated jointly. The method presented in Section 3.1.1 is an example of block updating: all coefficients in $\boldsymbol{\gamma}_k$ are accepted or rejected within one Metropolis-Hastings step. Knorr-Held and Rue (2002) propose a generic method for the joint update of larger blocks, i.e. blocks that consist of regression and precision parameters, for example $(\kappa_k, \boldsymbol{\gamma}_k)$. Here, the joint proposal is obtained by first sampling $\kappa_k$ from a distribution that is independent of $\boldsymbol{\gamma}_k$. The authors suggest to use $z \cdot \kappa_k$ as a proposal for $\kappa_k$ where $z$ is a random variable with density proportional to $1 + 1/z$, with $[1/f, f]$ and $f > 1$. The value $f$ needs to be tuned so that the desired acceptance rate is achieved. Since this distribution does not depend on the current state of the chain it will cancel itself out when computing the acceptance probability. A proposal for the regression coefficient $\boldsymbol{\gamma}_k$ is obtained by subsequently sampling from (3.6). The joint proposal is then accepted with probability

$$\alpha(\boldsymbol{\gamma}_k^c, \boldsymbol{\gamma}_k^p) = \min\left\{1, \frac{p(\boldsymbol{y}|\boldsymbol{\gamma}_k^p)p(\boldsymbol{\gamma}_k^p|\kappa_k^p)p(\kappa_k^p)\varphi(\boldsymbol{\gamma}_k^c|\tilde{\boldsymbol{\mu}}_k^p, \widetilde{\boldsymbol{Q}}_k^p(\kappa^c))}{p(\boldsymbol{y}|\boldsymbol{\gamma}_k^c)p(\boldsymbol{\gamma}_k^c|\kappa_k^c)p(\kappa_k^c)\varphi(\boldsymbol{\gamma}_k^p|\tilde{\boldsymbol{\mu}}_k^c, \widetilde{\boldsymbol{Q}}_k^c(\kappa^p))}\right\}. \tag{3.13}$$

Knorr-Held and Rue analyze different compositions of blocks in the context of disease mapping. Their blocks range from single-site blocks to blocks that contain all unknown parameters. It is concluded that a joint update of GMRF parameters together with corresponding hyperparameters may be necessary to ensure proper mixing and convergence of Markov chains.

In Chapter 4 it is shown that blocking strategies have a major influence on the computational complexity which is induced when considering MCMC inference for large-scale problems. Therefore, a thorough discussion of these strategies is postponed to this chapter.

### 3.1.5 Fully Bayes inference based on MCMC

The approach for constructing Markov chains for all unknown parameters within a fully Bayes setup can be summarized as follows. Given the number of MCMC iterations as well as initialized regression and precision parameters, perform the following steps in each iteration:

(1) For $k = 1, \ldots, p$ do:

a) Use the current state of the chain $\boldsymbol{\gamma}_k^c$ to compute $\tilde{\boldsymbol{Q}}_k^c$ and $\tilde{\boldsymbol{\mu}}_k^c$ according to (3.7) and (3.8), respectively, and sample a proposal $\boldsymbol{\gamma}_k^p$ from (3.6). Use this proposal to recompute $\tilde{\boldsymbol{Q}}_k^p$ and $\tilde{\boldsymbol{\mu}}_k^p$. Accept the proposal as the new state of the chain with probability $\alpha(\boldsymbol{\gamma}_k^c, \boldsymbol{\gamma}_k^p)$ as in (3.9).

b) Generate a new state for the precision parameter $\kappa_k$ by sampling from a Gamma distribution with parameters $\tilde{a}_k = a_k + \text{rk}(\boldsymbol{K}_k)/2$ and $\tilde{b}_k = b_k + \boldsymbol{\gamma}_k'\boldsymbol{K}_k\boldsymbol{\gamma}_k/2$.

(2) For an additional dispersion parameter $\phi$ use a customized sampling strategy. See Section 3.1.3 for possible sampling strategies for Gaussian or Beta distributed response variables.

If precision and regression parameters are updated jointly steps (a) and (b) have to be merged and (3.13) used as the acceptance probability.

Inference on all parameters relies on the samples that remain after discarding the realizations of an initial burn-in phase. From these samples, any quantity of interest with respect to the joint posterior distribution can be approximated.

## 3.2 Empirical Bayes inference for STAR models

This section summarizes the empirical Bayes approach to STAR models which has been presented by Fahrmeir et al. (2004). Based on the work of Green (1987) and Lin and Zhang (1999), Fahrmeir et al. showed how STAR models can be represented as generalized linear mixed models (GLMMs) and, therefore, how well-known estimation techniques from this class of models can be utilized. In this approach it is considered that the precision parameters are unknown constants rather than random quantitates, thus, the term *empirical Bayes*.

The empirical Bayes approach is appealing for the following reasons: First, the use of mixed model methodology allows the application of deterministic algorithms which are usually faster than MCMC simulations. Also, this eliminates concerns about the quality of estimands due to slow mixing or poor convergence of Markov chains. Second, since precision parameters are considered as fixed rather than random, no prior distributions for these hyperparameters need to be specified. This eliminates the necessity for sensitivity analyses with respect to parameters of such hyperpriors.

### 3.2.1 Mixed model representation

Starting point for the mixed model approach is the representation of general STAR models as GLMMs. This can be achieved by a reparametrization of the components in $\boldsymbol{\eta}$ as follows:

$$\begin{aligned} \boldsymbol{Z}_k\boldsymbol{\gamma}_k &= \boldsymbol{Z}_k\left(\tilde{\boldsymbol{U}}_k\boldsymbol{\beta}_k + \tilde{\boldsymbol{X}}_k\boldsymbol{\alpha}_k\right) \\ &= \boldsymbol{U}_k\boldsymbol{\beta}_k + \boldsymbol{X}_k\boldsymbol{\alpha}_k \end{aligned}$$

The decomposition of $\boldsymbol{\gamma}_k$ into $\tilde{\boldsymbol{U}}_k\boldsymbol{\beta}_k$ and $\tilde{\boldsymbol{X}}_k\boldsymbol{\alpha}_k$ is closely connected to the null space of the corresponding structure matrix $\boldsymbol{K}_k$ and can be obtained in different ways (Fahrmeir and Kneib, 2011, Chapter 4.2). One method which is applicable to general structure matrices is based on the spectral decomposition $\boldsymbol{K}_k = \boldsymbol{\Gamma}\boldsymbol{\Omega}\boldsymbol{\Gamma}'$, where $\boldsymbol{\Omega}$ is the diagonal matrix of the eigenvalues of $\boldsymbol{K}_k$ and $\boldsymbol{\Gamma}$ is the concatenation of the corresponding eigenvectors. The number of zero eigenvalues in $\boldsymbol{\Omega}$ equals the rank deficiency of $\boldsymbol{K}_k$. If the decomposition is split along eigenvalues that are zero ($\boldsymbol{\Gamma}_1$ and $\boldsymbol{\Omega}_1$) and those that are non-zero ($\boldsymbol{\Gamma}_2$ and $\boldsymbol{\Omega}_2$), the design matrices for the reparametrization can then be obtained by $\tilde{\boldsymbol{U}}_k = \boldsymbol{\Gamma}_1$ and $\tilde{\boldsymbol{X}}_k = \boldsymbol{\Gamma}_2\boldsymbol{\Omega}_2^{-1/2}$. Thus, $\tilde{\boldsymbol{U}}_k$ corresponds to the part of $\boldsymbol{\gamma}_k$ that is unpenalized by the structure matrix.

Using this reparametrization for all regression coefficients, the complete linear predictor can be represented as

$$\boldsymbol{\eta} = \boldsymbol{U}\boldsymbol{\beta} + \boldsymbol{X}\boldsymbol{\alpha}$$

with the overall design matrices $\boldsymbol{U} = (\boldsymbol{U}_1, \ldots, \boldsymbol{U}_p)$ and $\boldsymbol{X} = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_p)$ as well as the coefficients $\boldsymbol{\beta} = (\boldsymbol{\beta}_1', \ldots, \boldsymbol{\beta}_p')'$ and $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1', \ldots, \boldsymbol{\alpha}_p')'$. If an intercept is present, identity vectors in $\boldsymbol{U}_k$ must be deleted in order to guarantee identifiability for all regression coefficients.

The use of this reparametrization requires a new setup of prior distributions for the regression coefficients $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Following Fahrmeir et al. (2004) a diffuse prior is assumed for the fixed effect part, i.e. $p(\boldsymbol{\beta}) \propto 1$. The second part, $\boldsymbol{\alpha}$, is modeled as an unstructured random effect:

$$\boldsymbol{\alpha}|\boldsymbol{\kappa} \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{\Psi}^{-1})$$

with precision matrix $\boldsymbol{\Psi} = \mathrm{blockdiag}(\kappa_1\boldsymbol{I}_{m_1}, \ldots, \kappa_p\boldsymbol{I}_{m_p})$.

Inference for all unknown parameters is then performed iteratively: First, for given precision parameters one iteration of a Fisher scoring algorithm is performed in order to update all regression coefficients. Second, given updated regression coefficients, one iteration of a Fisher scoring algorithm is performed in order to update the precision parameters. The details of these steps are explained in the next sections.

### 3.2.2 Estimation of regression coefficients

Regression coefficients are estimated by maximizing the joint posterior of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. This can be achieved by a second order Taylor expansion of the log-posterior around $\boldsymbol{\alpha}_0$ and $\boldsymbol{\beta}_0$:

$$\begin{aligned}\log p(\boldsymbol{\beta},\boldsymbol{\alpha}|\boldsymbol{y}) &\propto l(\boldsymbol{\beta},\boldsymbol{\alpha}) + \log p(\boldsymbol{\beta}) + \log(\boldsymbol{\alpha}|\boldsymbol{\Psi}) \\ &\approx a_0 + \boldsymbol{b}_0'(\boldsymbol{\beta}',\boldsymbol{\alpha}')' - \frac{1}{2}(\boldsymbol{\beta}',\boldsymbol{\alpha}')\boldsymbol{C}_0(\boldsymbol{\beta}',\boldsymbol{\alpha}')' \qquad (3.14)\end{aligned}$$

This corresponds to the core of the logarithm of a multivariate Gaussian density with precision $\boldsymbol{C}_0$ and mode $(\tilde{\boldsymbol{\mu}}_\beta', \tilde{\boldsymbol{\mu}}_\alpha')'$ which is the solution of $\boldsymbol{C}_0(\tilde{\boldsymbol{\mu}}_\beta', \tilde{\boldsymbol{\mu}}_\alpha')' = \boldsymbol{b}_0$. Coefficients $\boldsymbol{b}_0$ and $\boldsymbol{C}_0$ can be derived in complete analogy to Section 3.1.1, that is,

$$\begin{aligned}\boldsymbol{b}_0 &= \frac{\partial \log p(\boldsymbol{\beta}_0,\boldsymbol{\alpha}_0|\boldsymbol{y})}{\partial(\boldsymbol{\beta}',\boldsymbol{\alpha}')'} + \boldsymbol{C}_0(\boldsymbol{\beta}_0',\boldsymbol{\alpha}_0')' \\ \boldsymbol{C}_0 &= -\frac{\partial^2 \log p(\boldsymbol{\beta}_0,\boldsymbol{\alpha}_0|\boldsymbol{y})}{\partial(\boldsymbol{\beta}',\boldsymbol{\alpha}')\partial(\boldsymbol{\beta}',\boldsymbol{\alpha}')'}\end{aligned}$$

Using the notation of GLMs the components of these coefficients can be expressed by the score function

$$\boldsymbol{s}(\boldsymbol{\beta}_0,\boldsymbol{\alpha}_0) = \begin{pmatrix} \boldsymbol{U}'\boldsymbol{D}_0\boldsymbol{V}_0^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_0) \\ \boldsymbol{X}'\boldsymbol{D}_0\boldsymbol{V}_0^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_0) - \boldsymbol{\Psi}\boldsymbol{\alpha}_0 \end{pmatrix}$$

and the Fisher information

$$\boldsymbol{F}(\boldsymbol{\beta}_0,\boldsymbol{\alpha}_0) = \begin{pmatrix} \boldsymbol{U}'\boldsymbol{W}_0\boldsymbol{U} & \boldsymbol{U}'\boldsymbol{W}_0\boldsymbol{X} \\ \boldsymbol{X}'\boldsymbol{W}_0\boldsymbol{U} & \boldsymbol{X}'\boldsymbol{W}_0\boldsymbol{X} + \boldsymbol{\Psi} \end{pmatrix}. \qquad (3.15)$$

From its definition it follows immediately that $\boldsymbol{C}_0 = \boldsymbol{F}(\boldsymbol{\beta}_0,\boldsymbol{\alpha}_0)$. By using the definition of working observations as in Section 3.1.1, $\boldsymbol{b}_0$ can be rewritten as

$$\begin{aligned}\boldsymbol{b}_0 &= \boldsymbol{s}(\boldsymbol{\beta}_0,\boldsymbol{\alpha}_0) + \boldsymbol{F}(\boldsymbol{\beta}_0,\boldsymbol{\alpha}_0)(\boldsymbol{\beta}_0',\boldsymbol{\alpha}_0')' \\ &= \begin{pmatrix} \boldsymbol{U}'\boldsymbol{D}_0\boldsymbol{V}_0^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_0) \\ \boldsymbol{X}'\boldsymbol{D}\boldsymbol{V}^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_0) - \boldsymbol{\Psi}\boldsymbol{\alpha}_0 \end{pmatrix} + \begin{pmatrix} \boldsymbol{U}'\boldsymbol{W}_0\boldsymbol{U}\boldsymbol{\beta}_0 + \boldsymbol{U}'\boldsymbol{W}_0\boldsymbol{X}\boldsymbol{\alpha}_0 \\ \boldsymbol{X}'\boldsymbol{W}_0\boldsymbol{U}\boldsymbol{\beta}_0 + \boldsymbol{X}'\boldsymbol{W}_0\boldsymbol{X}\boldsymbol{\alpha}_0 + \boldsymbol{\Psi}\boldsymbol{\alpha}_0 \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{U}'\boldsymbol{D}_0\boldsymbol{V}_0^{-1}\boldsymbol{D}_0(\boldsymbol{D}_0^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_0) + \boldsymbol{U}\boldsymbol{\beta}_0 + \boldsymbol{X}\boldsymbol{\alpha}_0) \\ \boldsymbol{X}'\boldsymbol{D}_0\boldsymbol{V}_0^{-1}\boldsymbol{D}_0(\boldsymbol{D}_0^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_0) + \boldsymbol{U}\boldsymbol{\beta}_0 + \boldsymbol{X}\boldsymbol{\alpha}_0) \end{pmatrix}\end{aligned}$$

$$= \begin{pmatrix} \boldsymbol{U}'\boldsymbol{W}_0\tilde{\boldsymbol{y}}_0 \\ \boldsymbol{X}'\boldsymbol{W}_0\tilde{\boldsymbol{y}}_0 \end{pmatrix}.$$

Therefore, the mode of (3.14) can be written as the solution of

$$\begin{pmatrix} \boldsymbol{U}'\boldsymbol{W}_0\boldsymbol{U} & \boldsymbol{U}'\boldsymbol{W}_0\boldsymbol{X} \\ \boldsymbol{X}'\boldsymbol{W}_0\boldsymbol{U} & \boldsymbol{X}'\boldsymbol{W}_0\boldsymbol{X} + \boldsymbol{\Psi} \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{\mu}}_\beta \\ \tilde{\boldsymbol{\mu}}_\alpha \end{pmatrix} = \begin{pmatrix} \boldsymbol{U}'\boldsymbol{W}_0\tilde{\boldsymbol{y}}_0 \\ \boldsymbol{X}'\boldsymbol{W}_0\tilde{\boldsymbol{y}}_0 \end{pmatrix}. \tag{3.16}$$

Solving this system of linear equations with respect to $\tilde{\boldsymbol{\mu}}_\beta$ and $\tilde{\boldsymbol{\mu}}_\alpha$ corresponds to one iteration of a Fisher scoring algorithm.

### 3.2.3 Estimation of precision and dispersion parameters

By adding $-0.5\tilde{\boldsymbol{y}}\boldsymbol{W}\tilde{\boldsymbol{y}}$ to (3.14) and performing straight forward calculations it follows that

$$\tilde{\boldsymbol{y}}|\boldsymbol{\beta},\boldsymbol{\alpha} \overset{a}{\sim} \mathrm{N}(\boldsymbol{U}\boldsymbol{\beta} + \boldsymbol{X}\boldsymbol{\alpha}, \boldsymbol{W}^{-1}). \tag{3.17}$$

This coincides with a linear mixed model for the working observations. Note that for normal distributed response variables $\tilde{\boldsymbol{y}} = \boldsymbol{y}$ holds. For this special case Harville (1974) shows how the marginal distribution for the error contrasts $\boldsymbol{u} = \boldsymbol{A}'\boldsymbol{y}$ can be obtained. Here, $\boldsymbol{A}$ is a $n \times (n - \dim(\boldsymbol{\beta}))$ matrix given by $\boldsymbol{A}\boldsymbol{A}' = \boldsymbol{U}(\boldsymbol{U}'\boldsymbol{U})^{-1}\boldsymbol{U}'$ with $\boldsymbol{A}'\boldsymbol{A} = \boldsymbol{I}$. The advantage of using the likelihood of error contrasts rather than $\boldsymbol{y}$ is that the resulting marginal likelihood for precision and dispersion parameters does not depend on $\boldsymbol{\beta}$ anymore. This makes it possible to obtain estimates for $\kappa_1, \dots, \kappa_p$ and $\phi$ by accounting for the uncertainty of $\boldsymbol{\beta}$. Estimating these parameters this way is also known as restricted maximum likelihood (REML, Patterson and Thompson, 1971). Applying this method to (3.17) yields the following approximate marginal likelihood for precision and dispersion parameters (Lin and Zhang, 1999):

$$l_M(\boldsymbol{\kappa}, \phi) = -\frac{1}{2}\left(\log|\boldsymbol{\Sigma}| + \log|\boldsymbol{U}'\boldsymbol{\Sigma}^{-1}\boldsymbol{U}| + (\tilde{\boldsymbol{y}} - \boldsymbol{U}\boldsymbol{\beta})'\boldsymbol{\Sigma}^{-1}(\tilde{\boldsymbol{y}} - \boldsymbol{U}\boldsymbol{\beta})\right)$$

with $\boldsymbol{\Sigma} = \boldsymbol{W}^{-1} + \boldsymbol{X}\boldsymbol{\Psi}^{-1}\boldsymbol{X}'$. Maximizing this restricted likelihood with respect to $\kappa_1, \dots, \kappa_p$ and $\phi$ yields REML estimates for these precision and dispersion parameters. Fahrmeir and Kneib (2011, Chapter 3.1.4) point out that the REML estimates for the precision and dispersion parameters coincide with the modes of the corresponding marginal posteriors within a fully Bayesian setup.

Numerical optimization is usually performed by Fisher scoring on the variance parameters $\tau_k^2 = \kappa_k^{-1}, k = 1, \dots, p$, rather than precision parameters (Fahrmeir et al., 2004). Here, in iteration $t+1$ a new value for $(\boldsymbol{\tau}^2, \phi)$ can be found via

$$\begin{pmatrix} \boldsymbol{\tau}^2 \\ \phi \end{pmatrix}^{(t+1)} = \begin{pmatrix} \boldsymbol{\tau}^2 \\ \phi \end{pmatrix}^{(t)} + \boldsymbol{F}(\boldsymbol{\tau}^{2(t)}, \phi^{(t)})^{-1} \boldsymbol{s}(\boldsymbol{\tau}^{2(t)}, \phi^{(t)}) \tag{3.18}$$

with the score vector $\boldsymbol{s}(\boldsymbol{\tau}^2, \phi) = ((\partial l_M(\boldsymbol{\tau}^2, \phi)/\partial \tau_k^2)_{k=1,\dots,,p}, \partial l_M(\boldsymbol{\tau}^2, \phi)/\partial \phi)'$ and the expected Fisher information

$$\boldsymbol{F}(\boldsymbol{\tau}^2, \phi) = -\mathrm{E} \begin{pmatrix} \left( \frac{\partial^2 l(\boldsymbol{\tau}^2, \phi)}{\partial \tau_k^2 \tau_j^2} \right)_{k,j=1,\dots,q} & \frac{\partial^2 l(\boldsymbol{\tau}^2, \phi)}{\partial \tau_k^2 \partial \phi} \\ \frac{\partial^2 l(\boldsymbol{\tau}^2, \phi)}{\partial \phi \partial \tau_k^2} & \frac{\partial^2 l(\boldsymbol{\tau}^2, \phi)}{\partial \phi^2} \end{pmatrix}.$$

When calculating these derivatives care must be taken in order to avoid computation and storing of huge matrices. See Kneib (2006) for details on this and for a more thorough derivation of the following formulas for the derivatives. The first $p$ elements of the score vector are given by

$$\begin{aligned} \frac{\partial l_M(\boldsymbol{\tau}^2, \phi)}{\partial \tau_k^2} = & -\frac{1}{2} \mathrm{tr} \left( \boldsymbol{X}_k' \boldsymbol{W} \boldsymbol{X}_k \right) + \frac{1}{2} \mathrm{tr} \left( \boldsymbol{X}_k' \boldsymbol{W} (\boldsymbol{U}, \boldsymbol{X}) \boldsymbol{F}^{-1} (\boldsymbol{U}, \boldsymbol{X})' \boldsymbol{W} \boldsymbol{X}_k \right) \\ & + \frac{1}{2} \left( \tilde{\boldsymbol{y}} - \boldsymbol{U}\boldsymbol{\beta} - \boldsymbol{X}\boldsymbol{\alpha} \right)' \boldsymbol{W} \boldsymbol{Z}_k \boldsymbol{Z}_k' \boldsymbol{W} \left( \tilde{\boldsymbol{y}} - \boldsymbol{U}\boldsymbol{\beta} - \boldsymbol{X}\boldsymbol{\alpha} \right). \end{aligned}$$

Here, $\boldsymbol{F}^{-1}$ refers to the inverse Fisher information (3.15). If the likelihood contains an additional dispersion parameter the corresponding derivative is given by

$$\begin{aligned} \frac{\partial l_M(\boldsymbol{\tau}^2, \phi)}{\partial \phi} = & -\frac{n}{2\phi} + \frac{1}{2\phi} \mathrm{tr} \left( (\boldsymbol{U}, \boldsymbol{X})' \boldsymbol{W} (\boldsymbol{U}, \boldsymbol{X}) \boldsymbol{F}^{-1} \right) \\ & + \frac{1}{2\phi} \left( \tilde{\boldsymbol{y}} - \boldsymbol{U}\boldsymbol{\beta} - \boldsymbol{X}\boldsymbol{\alpha} \right)' \boldsymbol{W} \left( \tilde{\boldsymbol{y}} - \boldsymbol{U}\boldsymbol{\beta} - \boldsymbol{X}\boldsymbol{\alpha} \right). \end{aligned}$$

The components of the first block of the expected Fisher information, that is, the main and mixed second derivatives for the variance parameters, are given by

$$\begin{aligned} -\mathrm{E} \left( \frac{\partial l_M(\boldsymbol{\tau}^2, \phi)}{\partial \tau_k^2 \partial \tau_l^2} \right) = & \frac{1}{2} \mathrm{tr} \left( \boldsymbol{X}_l' \boldsymbol{W} \boldsymbol{X}_k' \boldsymbol{X}_k' \boldsymbol{W} \boldsymbol{X}_l \right) - \mathrm{tr} \left( \boldsymbol{X}_l' \boldsymbol{W} (\boldsymbol{U}, \boldsymbol{X}) \boldsymbol{F}^{-1} (\boldsymbol{U}, \boldsymbol{X})' \boldsymbol{W} \boldsymbol{X}_k' \boldsymbol{X}_k' \boldsymbol{W} \boldsymbol{X}_l \right) \\ & + \frac{1}{2} \mathrm{tr} \left( \boldsymbol{X}_l' \boldsymbol{W} (\boldsymbol{U}, \boldsymbol{X}) \boldsymbol{F}^{-1} (\boldsymbol{U}, \boldsymbol{X})' \boldsymbol{W} \boldsymbol{X}_k' \boldsymbol{X}_k' \boldsymbol{W} (\boldsymbol{U}, \boldsymbol{X}) \boldsymbol{F}^{-1} (\boldsymbol{U}, \boldsymbol{X})' \boldsymbol{W} \boldsymbol{X}_l \right). \end{aligned}$$

The mixed derivatives with respect to the variance and dispersions parameters are

$$
\begin{aligned}
-\mathrm{E}\left(\frac{\partial^2 l_M(\boldsymbol{\tau}^2,\phi)}{\partial\tau_k^2\partial\phi}\right) =& -\frac{1}{2\phi}\mathrm{tr}\left(\boldsymbol{X}_k'\boldsymbol{W}\boldsymbol{X}_k\right) - \frac{1}{\phi}\mathrm{tr}\left((\boldsymbol{U},\boldsymbol{X})'\boldsymbol{W}\boldsymbol{X}_k'\boldsymbol{X}_k\boldsymbol{W}(\boldsymbol{U},\boldsymbol{X})\boldsymbol{F}^{-1}\right)\\
&+\frac{1}{2\phi}\mathrm{tr}\left((\boldsymbol{U},\boldsymbol{X})'\boldsymbol{W}(\boldsymbol{U},\boldsymbol{X})\boldsymbol{F}^{-1}(\boldsymbol{U},\boldsymbol{X})'\boldsymbol{W}\boldsymbol{X}_k'\boldsymbol{X}_k\boldsymbol{W}(\boldsymbol{U},\boldsymbol{X})\right)
\end{aligned}
$$

and the second derivative with respect to the dispersion parameter is

$$
\begin{aligned}
-\mathrm{E}\left(\frac{\partial^2 l_M(\boldsymbol{\tau}^2,\phi)}{\partial\phi^2}\right) =& \frac{n}{2\phi^2} - \frac{1}{\phi^2}\mathrm{tr}\left((\boldsymbol{U},\boldsymbol{X})'\boldsymbol{W}(\boldsymbol{U},\boldsymbol{X})\boldsymbol{F}^{-1}\right)\\
&+\frac{1}{2\phi^2}\mathrm{tr}\left((\boldsymbol{U},\boldsymbol{X})'\boldsymbol{W}(\boldsymbol{U},\boldsymbol{X})\boldsymbol{F}^{-1}(\boldsymbol{U},\boldsymbol{X})'\boldsymbol{W}(\boldsymbol{U},\boldsymbol{X})\boldsymbol{F}^{-1}\right).
\end{aligned}
$$

### 3.2.4 Empirical Bayes inference using mixed model representation

Given a STAR model in mixed model representation the empirical Bayes approach can be summarized by the following steps. First, initial starting values for regression coefficients need to be chosen. Here, ordinary least squares may provide a good starting point for $\boldsymbol{\beta}$. Next, the following steps are repeated until no more significant changes in the parameters are observed:

(1) To update $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ calculate $\tilde{\boldsymbol{y}}$ and $\boldsymbol{W}$ and compute the mode of the approximate posterior (3.14) by solving (3.16).

(2) To update precision and possible dispersion parameters compute the elements in $\boldsymbol{s}(\boldsymbol{\tau}^2,\phi)$ and $\boldsymbol{F}(\boldsymbol{\tau}^2,\phi)$ and perform one iteration of a Fisher scoring algorithm using (3.18).

After convergence, estimates of the original parameters of the STAR model can be retrieved via

$$\hat{\gamma}_k = \tilde{\boldsymbol{U}}_k\hat{\boldsymbol{\beta}}_k + \tilde{\boldsymbol{X}}_k\hat{\boldsymbol{\alpha}}_k.$$

Inference with regard to the estimated $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\alpha}}$ relies on the Gaussian approximation (3.14) of the posterior. Therefore, standard errors for these coefficients can be obtained from the diagonal elements of the inverse Fisher information (3.15). For function evaluations $\hat{\boldsymbol{f}}_k = \boldsymbol{Z}_k\hat{\gamma}_k$, standard errors are given by the diagonal elements of $\mathrm{cov}(\hat{\boldsymbol{f}}_k) =$

$(\boldsymbol{U}_k\boldsymbol{X}_k)\boldsymbol{F}^{-1}(\boldsymbol{U}_k\boldsymbol{X}_k)'$, see Lin and Zhang (1999). With this, point wise credible intervals for $\hat{\boldsymbol{f}}_k$ can be constructed. A formula for simultaneous credible intervals as well as a discussion of tests on the functional form can be found in Fahrmeir and Kneib (2011, Section 4.2.1).

## 3.3 Approximate inference for STAR models

A method that had a huge impact on the applied Bayesian community is the integrated nested Laplace approximation (INLA) approach presented by Rue et al. (2009). This approach allows to obtain striking precise results for latent Gaussian models in a comparatively short amount of computation time. This is made possible through the effective implementation and the massive use of numerical optimization and integration techniques.

In order to introduce this approach some simplifications with respect to notations need to be made. Let $\boldsymbol{x}$ be the collection of all unknown regression coefficients, i.e. $\boldsymbol{x} = (\boldsymbol{\gamma}_1', \dots, \boldsymbol{\gamma}_p')'$, and let $\boldsymbol{\theta}$ be the set of all unknown hyperparameters, i.e. $\boldsymbol{\theta} = (\kappa_1, \dots, \kappa_q, \phi)$. The main objective of the INLA approach is to estimate the marginal posterior distributions for the regression coefficients

$$p(x_j|\boldsymbol{y}) = \int_{\boldsymbol{\theta}} p(x_j|\boldsymbol{\theta}, \boldsymbol{y})p(\boldsymbol{\theta}|\boldsymbol{y})\mathrm{d}\boldsymbol{\theta}. \tag{3.19}$$

The INLA approach solves this integral by numerical integration, as explained below. This requires appropriate approximations to the first part of the integral, the full conditional posterior distribution of $x_j$ given $\boldsymbol{\theta}$ and $\boldsymbol{y}$, and to the second part of the integral, the marginal distribution of $\boldsymbol{\theta}$.

Before going into more detail the joint distribution of $\boldsymbol{x}$ needs to be derived. In order to guarantee a unique connection of the graph of the GMRF $\boldsymbol{x}$ to the observed data $\boldsymbol{y}$, $\boldsymbol{x}$ is expanded by the linear predictor $\boldsymbol{\eta}$, i.e. $\boldsymbol{x} = (\boldsymbol{\eta}, \boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_p)$, and a small error is added to $\boldsymbol{\eta}$:

$$\boldsymbol{\eta} = \boldsymbol{Z}_1\boldsymbol{\gamma}_1 + \cdots + \boldsymbol{Z}_p\boldsymbol{\gamma}_p + \boldsymbol{\varepsilon}$$

with $\boldsymbol{\varepsilon} \sim N(0, \kappa_\varepsilon^{-1}\boldsymbol{I}_n)$, where $\kappa_\varepsilon$ is set high in order to keep the error small, for example $\kappa_\varepsilon = 1e^6$. Given this parameterization the joint density of $\boldsymbol{x}$ can be derived as follows:

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{\kappa}) &\propto p(\boldsymbol{\varepsilon}|\kappa_\varepsilon)p(\boldsymbol{\gamma}_1|\kappa_1)\cdot\dots\cdot p(\boldsymbol{\gamma}_p|\kappa_p) \\ &= p(\boldsymbol{\eta} - \boldsymbol{Z}_1\boldsymbol{\gamma}_1 - \dots - \boldsymbol{Z}_q\boldsymbol{\gamma}_q|\kappa_\varepsilon)p(\boldsymbol{\gamma}_1|\kappa_1)\cdot\dots\cdot p(\boldsymbol{\gamma}_q|\kappa_q) \\ &\propto \exp\left(-\frac{1}{2}\boldsymbol{x}'\boldsymbol{Q}\boldsymbol{x}\right) \end{aligned} \tag{3.20}$$

where the precision matrix $\boldsymbol{Q}$ has the form

$$\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{Q}_{\eta\eta} & \boldsymbol{Q}_{\eta\gamma_1} & \cdots & \boldsymbol{Q}_{\eta\gamma_p} \\ \boldsymbol{Q}_{\gamma_1\eta} & \boldsymbol{Q}_{\gamma_1\gamma_1} & \cdots & \boldsymbol{Q}_{\gamma_1\gamma_p} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{Q}_{\gamma_p\eta} & \boldsymbol{Q}_{\gamma_p\gamma_1} & \cdots & \boldsymbol{Q}_{\gamma_p\gamma_p} \end{pmatrix}$$

with entries

$$\begin{aligned} \boldsymbol{Q}_{\eta\eta} &= \kappa_\varepsilon \boldsymbol{I}_n \\ \boldsymbol{Q}_{\eta\gamma_k} &= -\kappa_\varepsilon \boldsymbol{Z}_k \\ \boldsymbol{Q}_{\gamma_k\gamma_k} &= \kappa_\varepsilon \boldsymbol{Z}_k'\boldsymbol{Z}_k + \kappa_k\boldsymbol{K}_k \\ \boldsymbol{Q}_{\gamma_k\gamma_l} &= \kappa_\varepsilon \boldsymbol{Z}_k'\boldsymbol{Z}_l,\ k \neq l. \end{aligned} \tag{3.21}$$

Note that this is the joint prior distribution of all unknown regression coefficients and the values of the linear predictor. No observed data has been included into this distribution, yet.

### 3.3.1 Exploring the marginal posterior of $\boldsymbol{\theta}$

The numerical integration of (3.19) is performed over representative points of $\boldsymbol{\theta}$, that is, values of $\boldsymbol{\theta}$ that are most likely under $p(\boldsymbol{\theta}|\boldsymbol{y})$. Starting point for finding these values is the identity

$$\begin{aligned} p(\boldsymbol{\theta}|\boldsymbol{y}) &= \frac{p(\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{y})}{p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})p(\boldsymbol{y})} \\ &\propto \frac{p(\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{y})}{p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})}. \end{aligned} \tag{3.22}$$

Within the INLA approach, $p(\boldsymbol{\theta}|\boldsymbol{y})$ is approximated by a Laplace approximation (Tierney and Kadane, 1986) as explained in the following. First, note that, in accordance to Section 3.1.1, the denominator of (3.22) can be written as

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y}) &\propto p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{x}) \\ &\propto |\boldsymbol{Q}|^{1/2}\exp\left(-\frac{1}{2}\boldsymbol{x}'\boldsymbol{Q}\boldsymbol{x}+\sum_{i=1}^{n}\log p(y_i|\boldsymbol{x})\right). \end{aligned}$$

Now, it becomes obvious why $\boldsymbol{\eta}$ has been added to $\boldsymbol{x}$: The unique connection between $\boldsymbol{\eta}$ and $\boldsymbol{y}$ allows to expand the log-likelihood by a quadratic Taylor expansion around $\boldsymbol{\eta}^0$, i.e.

$$\log p(y_i|\eta_i) \approx a_i + b_i\eta_i - \tfrac{1}{2}c_i\eta_i^2$$

with coefficients

$$\begin{aligned} a_i &= \log p(y_i|\eta_i^0) - \frac{\partial \log p(y_i|\eta_i^0)}{\partial\eta}\eta_i^0 + \frac{1}{2}\frac{\partial^2 \log p(y_i|\eta_i^0)}{\partial\eta^2}(\eta_i^0)^2 \\ b_i &= \frac{\partial \log p(y_i|\eta_i^0)}{\partial\eta} - \frac{\partial^2 \log p(y_i|\eta_i^0)}{\partial\eta^2}\eta_i^0 \\ c_i &= -\frac{\partial^2 \log p(y_i|\eta_i^0)}{\partial\eta^2}. \end{aligned}$$

When defining $\boldsymbol{b} = (b_1,\ldots,b_n,0,\ldots,0)$ and $\boldsymbol{C} = \operatorname{diag}(c_1,\ldots,c_n,0,\ldots,0)$, where the number of zeros in $\boldsymbol{b}$ and in the main diagonal of $\boldsymbol{C}$ equals the dimension of $\boldsymbol{x}_{-\boldsymbol{\eta}}$, $p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})$ can be approximated by

$$\begin{aligned} \tilde{p}_G(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y}) &\approx |\boldsymbol{Q}|^{1/2}\exp\left(-\frac{1}{2}\boldsymbol{x}'\boldsymbol{Q}\boldsymbol{x}+\sum_{i=1}^{n}(a_i+b_i\eta_i-\tfrac{1}{2}c_i\eta_i^2)\right) \\ &\propto |\boldsymbol{Q}|^{1/2}\exp\left(-\frac{1}{2}\boldsymbol{x}'(\boldsymbol{Q}+\boldsymbol{C})\boldsymbol{x}+\boldsymbol{b}'\boldsymbol{x}\right). \end{aligned} \tag{3.23}$$

Plugging this GMRF approximation back into the denominator of (3.22) yields the Laplace approximation of $p(\boldsymbol{\theta}|\boldsymbol{y})$ :

$$\tilde{p}(\boldsymbol{\theta}|\boldsymbol{y}) \approx \left.\frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})\prod_{i=1}^{n}p(y_i|x_i)}{\tilde{p}_G(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})}\right|_{\boldsymbol{x}=\boldsymbol{x}^*(\boldsymbol{\theta})}.$$

Here, $\boldsymbol{x}^*(\boldsymbol{\theta})$ denotes the mode of $\tilde{p}_G(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})$ for $\boldsymbol{\theta}$. Given this approximation for the marginal posterior of $\boldsymbol{\theta}$, the INLA approach starts by finding the mode of this function. This can be achieved by using quasi-Newton methods such as the BFGS algorithm. Note that, during each iteration, the mode of the GMRF approximation in the denominator needs to be calculated for the current value of $\boldsymbol{\theta}$. After the mode of $\tilde{p}(\boldsymbol{\theta}|\boldsymbol{y})$ has been localized, the function is further explored. To this end, the negative Hessian is computed in order to efficiently lay out a grid over the parameter space of $\boldsymbol{\theta}$. The coordinates of this grid serve as the desired representative points for $\boldsymbol{\theta}$. In addition, marginal posteriors $p(\theta_k|\boldsymbol{y})$ can be derived by numerical integration of an interpolation of the function evaluations over the grid structure.

### 3.3.2 Approximation of the full conditional of $x_j$

From equation (3.19) it remains to approximate the first part under the integral, i.e. the full conditional posterior of $x_j$ given $\boldsymbol{\theta}$ and $\boldsymbol{y}$. Rue et al. (2009) present three approximations for this term which differ in their order of complexity and accuracy. The first and simplest is the GMRF approximation (3.23):

$$\tilde{p}_G(x_j|\boldsymbol{\theta}, \boldsymbol{y}) = \varphi(x_j|\mu_j(\boldsymbol{\theta}), \sigma_j^2(\boldsymbol{\theta})).$$

This approximation is appealing since it has already been used during the exploration of $\tilde{p}(\boldsymbol{\theta}|\boldsymbol{y})$. Rue and Martino (2007) explain in detail how the missing marginal variances can be computed recursively from the Cholesky decomposition of the precision matrix. As a more precise alternative to the GMRF approximation, Rue et al. (2009) propose to also apply the Laplace approximation to $p(x_j|\boldsymbol{\theta}, \boldsymbol{y})$. The density of this full Laplace approximation can be written as

$$\tilde{p}_{LA}(x_j|\boldsymbol{\theta}, \boldsymbol{y}) \propto \varphi(x_j|\mu_j(\boldsymbol{\theta}), \sigma_j^2(\boldsymbol{\theta})) \exp(\text{cubic spline}(x_j)).$$

This approximation is extremely precise but computationally more demanding. The third approximation, the simplified Laplace approximation, can be obtained by fitting a skewed normal distribution through a series expansion of the full Laplace approximation. While computationally less demanding than $\tilde{p}_{LA}(x_j|\boldsymbol{\theta}, \boldsymbol{y})$, this approximation is still able to correct for possible mismatches of $\tilde{p}_G(x_j|\boldsymbol{\theta}, \boldsymbol{y})$ with respect to location and skewness.

### 3.3.3 Approximation of the marginal posterior of $x_j$

Given representative points for $\boldsymbol{\theta}$ and corresponding function evaluations as well as $\tilde{p}(x_j|\boldsymbol{\theta}, \boldsymbol{y})$ the marginal posterior of $x_j$ can be approximated by solving (3.19) using numerical integration techniques. To be more precise, the integral is approximated by a weighted finite sum over the support points of $\boldsymbol{\theta}$:

$$\tilde{p}(x_j|\boldsymbol{y}) = \sum_k \tilde{p}(x_j|\boldsymbol{\theta}_k, \boldsymbol{y})\tilde{p}(\boldsymbol{\theta}_k|\boldsymbol{y})\Delta_k. \tag{3.24}$$

### 3.3.4 Approximate inference using the INLA approach

For STAR models, approximate inference using the INLA approach can be summarized by the following steps:

(1) Find the mode of $\tilde{p}(\boldsymbol{\theta}|\boldsymbol{y})$ using quasi-Newton methods.

(2) Create an efficient grid structure using the negative Hessian and explore $\tilde{p}(\boldsymbol{\theta}|\boldsymbol{y})$ along this grid. Keep the coordinates of this grid as support points of $\boldsymbol{\theta}$ as well as the corresponding function evaluations.

(3) Approximate $p(x_j|\boldsymbol{\theta}, \boldsymbol{y})$ using either the GMRF approximation, the Laplace approximation, or the simplified Laplace approximation.

(4) Use the results from steps 1 to 3 to compute the weighted sum (3.24) for the $x_j$'s of interest.

Due to the numerical exploration of $p(\boldsymbol{\theta}|\boldsymbol{y})$ this approach is limited to a moderate number of hyperparameters, that is, numerical complexity increases with the number of hyperparameters and, therefore, with the number of (nonlinear) predictors. This issue will be further discussed with respect to the application to large-scale problems in Chapter 4.

## 3.4 Chapter summary

Three different approaches for fitting STAR models have been presented in this chapter. The first method explores the joint posterior by creating random draws using MCMC techniques. It is applicable to a wide range of situations and easily expandable. The second method utilizes a reparametrization of the linear predictor in order to apply mixed model methodology. The resulting algorithm is deterministic, quite stable and well tested. Finally, the INLA approach has been presented as an approximate method for fitting STAR models. While also deterministic, this approach leads to precise estimates which can be obtained in a comparatively short amount of computation time.

# 4 Adaptations to large-scale problems

In the previous chapter, different strategies for fitting STAR models have been presented. This chapter examines their performance in the presence of high-dimensional regression coefficients. It can be divided into two parts. The first part is concerned with general considerations: An essential requirement is discussed and the three inference strategies are compared. Based on this comparison one inference strategy is chosen to be the main focus in the remainder of this thesis. For this method, the second part presents adjustments that are necessary in order to be applicable to large-scale problems.

The main findings of this Chapter have been published in Schmidt et al. (2017). In particular, the information of the following sections can be found in this publication: Section 4.1.2, 4.2.1, 4.2.2, 4.2.3, 4.2.4, and 4.2.5.

## 4.1 General considerations

### 4.1.1 Sparseness as a prerequisite

As a direct consequence of large-scale data situations, the design and structure matrices of the corresponding GMRF priors will also be of high dimension. Thus, in order to be computational feasible these matrices cannot be stored as regular matrices. In Chapter 2 it has been shown that a variety of effect types can be formulated using sparse matrices, i.e. matrices with only a small amount of non-zero entries. The advantage of dealing with sparse matrices is that they can be represented in special ways which simplify storing and allow for faster computation. In order to give better insight in the advantage with respect to storing, consider the sparse matrix representation implemented in Matlab (Gilbert et al., 1992). Here, only non-zero entries are saved along with their corresponding row and column indices. To be more precise, the following vectors are stored: A vector of row indices for the non-zero values, the non-zero values themselves, and a vector for the cumulative column indices. Therefore, on a 64-bit system where real and integer values are both represented by 8-bytes, a sparse matrix $\boldsymbol{A}$ of dimension $n \times m, m > 1$, requires $16 \cdot \mathrm{nnz}(\boldsymbol{A}) + (m+1) \cdot 8$ bytes of storage, where $\mathrm{nnz}(\cdot)$ describes the number of non-zero elements of a matrix. In contrast, if $\boldsymbol{A}$ would be stored as a dense matrix, it would require $8 \cdot n \cdot m$ bytes of storage. For a concrete example, consider a spatial GMRF prior over a three-dimensional regular lattice of size $n_x \times n_y \times n_z$ where the dependency structure between voxels is induced by a RW1. Thus, according to (2.17) the corresponding structure matrix can be written as

$$\boldsymbol{K}_{xyz} = (\boldsymbol{K}_y \otimes \boldsymbol{I}_{n_x} + \boldsymbol{I}_{n_y} \otimes \boldsymbol{K}_x) \otimes \boldsymbol{I}_z + \boldsymbol{I}_{n_x \cdot n_y} \otimes \boldsymbol{K}_z. \tag{4.1}$$

Here, $\boldsymbol{K}_{(\cdot)}$ is the structure matrix with respect to a RW1 in one dimension as in Section 2.2.1. The left panel of Figure 4.1 shows the required log-storage by a dense (straight line) and sparse (dashed line) representation of this structure matrix with increasing dimensionality for the special case of $n_x = n_y = n_z$ on working station A, see Section A.1. For $n_x = 12$ the dense matrix ($\sim$ 22.8 MB) already requires more than 100 times more storage than the sparse version ($\sim$ 189 KB). The right panel of this figure shows the advantage with respect to computation time. Here, the matrix-vector product $\boldsymbol{K}_{xyz}\boldsymbol{x}$ with $\boldsymbol{x} \sim \mathrm{U}(0,1)$ was computed 1,000 times for $n_x \in \{5, \ldots, 30\}$. In this graph, the lines represent the mean over all 1,000 trials. For $n_x = 13$ this calculation is on average about 100 times faster for the sparse representation. One can therefore conclude that sparse matrices can be seen as a necessary component when working on large-scale problems.

### 4.1.2 Comparison of inference strategies

Three different inference strategies for fitting STAR models have been presented in Chapter 3: A fully Bayes approach based on MCMC, an empirical Bayes method based on mixed model representation, and the INLA procedure as an example for approximate Bayesian inference. This section examines the performance of these methods in the presence of high-dimensional regression coefficients. The main question addressed here is whether it is possible to conduct inference on working stations with limited computational resources. Advantages and disadvantages are discussed and potential bottlenecks identified. Eventually, one inference strategy is chosen for which solutions to these bottlenecks are provided later in this chapter.

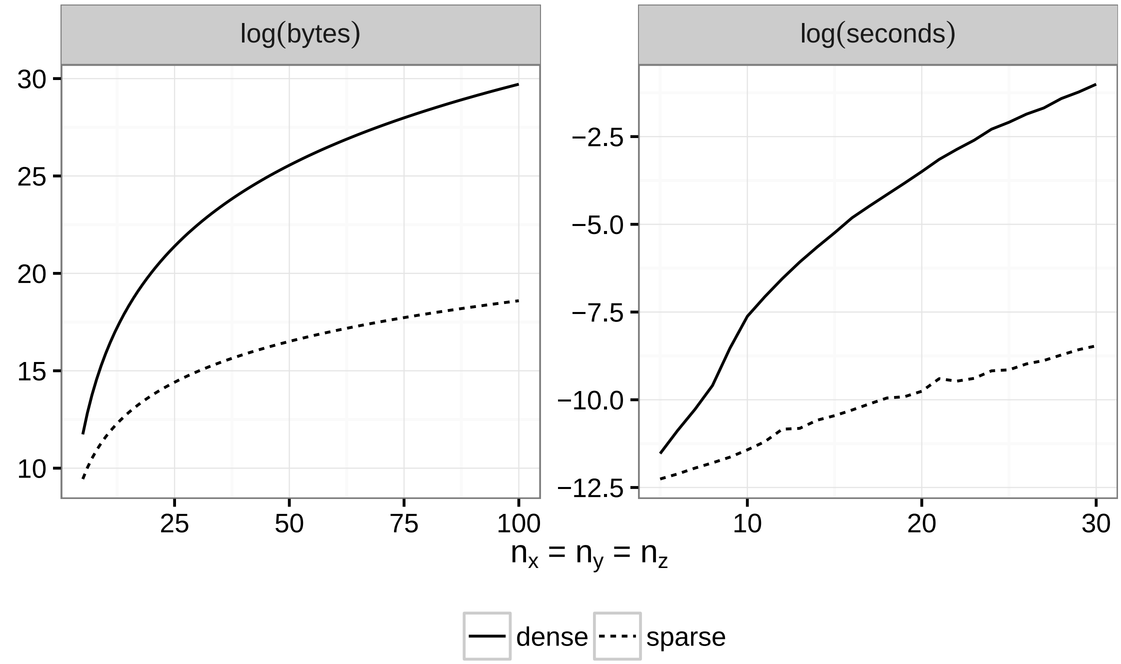


Figure 4.1 Comparison of storage and computation times for dense and sparse matrix representation.

### Interpretation of results

The three inference strategies differ remarkably in the way results are provided. In MCMC based inference, results rely solely on generated samples of the joint posterior distribution. From these samples any quantity of interest can be computed in order to summarize marginal posteriors. For example, moments of any degree, quantiles as well as credible intervals can easily be obtained. In addition, joint posterior distributions can also be summarized. By increasing the number of MCMC samples the accuracy of these summary statistics can be improved to any given precision. Furthermore, since MCMC inference does not rely on asymptotic assumptions it is an appropriate candidate for the analysis of small samples sizes. Of practical relevance is also the fact that transformations of parameters can directly be obtained by transforming the corresponding samples. In a similar manner various contrasts, i.e. linear combinations of regression coefficients, can be computed. Moreover, availability of marginal posteriors for precision parameters allows to assess the variability of these parameters and to make statements about the importance of the corresponding model terms. However, with respect to the existence of high-dimensional regression coefficients the necessity to collect a large number of MCMC samples in order to obtain sufficient precise estimates may lead to computational problems. For example, storing 10,000 samples of a 100,000-dimensional regression coefficient already requires about 7.45 GB of storage.

In the empirical Bayes approach discussed in Section 3.2 regression coefficients are estimated by the posterior mode of the GMRF approximation of the corresponding joint posterior. In addition, approximate standard errors can be obtained from the diagonal elements of the inverse of the Fisher information (3.15). Together, this information is used to approximate marginal posteriors for regression coefficients by univariate Gaussian distributions from which credible intervals can be computed. Accuracy of these estimates is asymptotical, that is, the validity of the Gaussian approximation depends mainly on the sample size. Transformations of parameters can be obtained by change of variables and linear combinations may be computed by using well known results for the sums of Gaussian random variables. However, for the latter an estimate of the covariance between coefficients is necessary. For precision parameters only point estimates can be derived. However, Fahrmeir and Kneib (2011, Section 4.2.1) give an introduction on how to set up null hypothesis significance tests for precision parameters and, thus, for the functional form of nonparametric functions. In general, storing the results of high-dimensional regression coefficients should be possible without further adjustments since the corresponding approximate marginal posteriors can be completely characterized by only two parameters.

The results of the INLA approach are finite sets of function evaluations of all approximate marginal posteriors of interest. From this, standard errors, credible intervals as well as any other measure can be obtained by interpolation and numerical integration techniques. The precision of these results depends on the strategy that has been chosen for the approximation of the full conditionals of regression coefficients, see Section 3.3.2. However, the accuracy is usually comparable with that obtained from long MCMC runs (Rue et al., 2009). Transformations of posterior marginals can be obtained by change of variables using numerical derivatives. Marginal posteriors for linear combinations of nodes can be specified directly by expanding the GMRF $\boldsymbol{x}$ prior to the fitting process. Alternatively, an approximation of these marginal posterior can be obtained by combining the corresponding means and variances and integrating out $\boldsymbol{\theta}$ using numerical integration (Martins et al., 2013). Variability of precision parameters can usually be assessed through marginal posteriors for these parameters. Possible problems with respect to high-dimensional regression coefficients may arise if the number of function evaluations for the marginal posteriors is high. However, these densities can usually be well approximated by a small number of parameters or quantiles.

### Size of Gaussian Markov random fields

The size of the largest GMRF within a model is a critical factor since it has a major influence on the computational requirements. Due to different parametrizations this size varies remarkably along different inference strategies. The Gibbs-structure of the MCMC sampling scheme allows to work on each regression coefficient separately. Thus, the size of the largest GMRF is constituted by the size of the largest regression coefficient which is given by $\max_k m_k$, where $m_k$ is the dimension of $\boldsymbol{\gamma}_k$. For the empirical Bayes approach it is not possible to work on each regression coefficient separately. Instead, updated values for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are obtained by maximizing the GMRF approximation to the joint posterior of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ given in (3.14). The corresponding graph of this GMRF is defined over the elements in $(\boldsymbol{\beta}', \boldsymbol{\alpha}')'$, thus it is of dimension $\sum_k m_k$. In order to evaluate $p(\boldsymbol{\theta}|\boldsymbol{y})$ within the INLA approach it is required to compute the mode of the GMRF approximation $\tilde{p}_G(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})$ given in (3.23). Here, $\boldsymbol{x}$ is the joint GMRF $\boldsymbol{x} = (\boldsymbol{\eta}', \boldsymbol{\gamma}_1', \ldots, \boldsymbol{\gamma}_p')'$ which is of dimension $n + \sum_k m_k$, where $n$ is the number of all observations.

How much the size of these GMRFs differs depends strongly on the problem at hand. For example, the difference between the GMRFs of the MCMC based inference and the empirical Bayes approach for the application in Section 6.1 is limited. Here, the size of the largest GMRF within the MCMC approach is $\max_k m_k = 565{,}475$ and that of $(\boldsymbol{\beta}', \boldsymbol{\alpha}')'$ is $\sum_k m_k = 565{,}477$. However, for the application in Section 6.2 the difference is more pronounced (565,475 for the MCMC approach and $5 \cdot 565{,}475 = 2{,}827{,}375$ for empirical Bayes). For the INLA approach this difference is even more noticeable: For the application in Section 6.1 the corresponding GMRF is of size $n + \sum_k m_k = 71{,}024{,}765 + (565{,}475 + 2) = 71{,}590{,}242$, and for that in Section 6.2 it consists of $247 \cdot 565{,}475 + 5 \cdot 565{,}475 = 142{,}499{,}700$ nodes.

### Sparseness

In Section 4.1.1 it has been shown that sparse matrix algebra is a main condition for handling huge matrices. It is therefore required that the different model parameterizations keep existing sparsity structures and avoid dense representations of large matrices. With respect to the presented MCMC algorithm all sparse matrices are kept during the fitting process as no re-parameterizations of model parameters are performed. However, it may not be obvious that this also holds for $\widetilde{\boldsymbol{Q}}_k$, the precision matrix of the proposal (3.6). From equation (3.4) it can be seen that $\widetilde{\boldsymbol{Q}}_k$ is sparse only if $\boldsymbol{Q}_k$ and $\boldsymbol{C}_k$ are sparse. For $\boldsymbol{Q}_k$ the Markovian property of GMRFs usually ensures that only a small amount of nodes in $\boldsymbol{\gamma}_k$ are directly connected with each other which leads to a high amount of sparseness. The sparsity structure of $\boldsymbol{C}_k$ strongly depends on the structure of $\boldsymbol{Z}_k$. For most effect types presented in Section 2.2.1 the design matrices are indicator matrices. For temporal and spatial effects, for example, each observation is usually assigned to one time point or spatial unit such as an administrative district or pixel. From this it follows that $\partial^2 l(\boldsymbol{\gamma})/(\partial\gamma_k\partial\gamma_l) = 0$ for $k \neq l$, thus $\boldsymbol{C}_k$ is diagonal and therefore computational undemanding. This implies that the amount of sparseness of $\widetilde{\boldsymbol{Q}}_k$ is equal to that of $\boldsymbol{Q}_k$.

In order to enable the use of mixed model methodology the empirical Bayes approach reformulates STAR models as working generalized linear mixed models which requires to decompose $\boldsymbol{\gamma}_k$ into a penalized and an unpenalized part. In general, this decomposition can be obtained by a spectral decomposition of the corresponding structure matrix. Since this decomposition usually produces dense matrices it may not be possible to obtain this factorization for regression coefficients that are of high dimension. However, in some cases, the required factorization can directly be obtained from the construction of the precision matrix. For example, the structure matrix that corresponds to a RW1 or RW2 can be written as $\boldsymbol{K} = \boldsymbol{D}'\boldsymbol{D}$ with $\boldsymbol{D}$ defined as in Section 2.2.1. A similar factorization can be obtained for Kronecker product penalties. Let $\boldsymbol{K} = \boldsymbol{K}_1 \otimes \boldsymbol{K}_2$ with $\boldsymbol{K}_1 = \boldsymbol{D}_1'\boldsymbol{D}_1$ and $\boldsymbol{K}_2 = \boldsymbol{D}_2'\boldsymbol{D}_2$. Basic calculus for Kronecker products then yields $\boldsymbol{K} = (\boldsymbol{D}_1'\boldsymbol{D}_1)\otimes(\boldsymbol{D}_2'\boldsymbol{D}_2) = (\boldsymbol{D}_1' \otimes \boldsymbol{D}_2')(\boldsymbol{D}_1 \otimes \boldsymbol{D}_2)$'. The resulting structure matrices are in both cases sparse. For many other structure matrices, though, such decompositions cannot be given. Consider, for example, the Kronecker sum penalty for the three-dimensional extension of the RW1 given in (2.17). For this structure matrix, which is extensively used in later chapters of this thesis, no such simple factorization is known. Thus, the spectral decomposition needs to be used in this situation which results in dense design matrices. A further violation of sparseness can be observed during the estimation of precision parameters, see below.

With respect to the INLA approach sparseness concerns the precision matrix $\boldsymbol{Q} + \boldsymbol{C}$ of the GMRF approximation $\tilde{p}_G(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})$. For the diagonal blocks of $\boldsymbol{Q}$ the same arguments apply as for the MCMC approach. The off-diagonal blocks are usually sparse as well since they are the product of two sparse design matrices. Sparsity of $\boldsymbol{C}$ follows directly from its definition.

### Estimation of precision parameters

Estimation of precision and dispersion parameters differs greatly between inference strategies. Within the MCMC approach, precision parameters are sampled from Gamma distributions as explained in Section 3.1.2. Usually, this is possible without much computational effort. Sampling of dispersion parameters depends on the problem at hand. However, the two special cases discussed in Section 3.1.3 can easily be applied to large-scale data situations.

The empirical Bayes approach uses REML estimates for variance parameters, see Section 3.2.3. The required derivatives of the approximate marginal log-likelihood includes the computation of traces of huge matrices. For example, consider the first derivative of $l_M(\boldsymbol{\kappa}, \phi)$ with respect to $\tau_k^2 = \kappa_k^{-1}$. Here, the trace of

$$\boldsymbol{X}_k' \boldsymbol{W}(\boldsymbol{U}, \boldsymbol{X}) \boldsymbol{V} \boldsymbol{W} \boldsymbol{X}_k$$

needs to be evaluated with $\boldsymbol{V}$ as the solution of $\boldsymbol{F}\boldsymbol{V} = (\boldsymbol{U}, \boldsymbol{X})'$. Note that solving this system requires solving $\sum_k m_k$ linear systems of the form $\boldsymbol{F}\boldsymbol{x} = \boldsymbol{b}$ with $\dim(\boldsymbol{F}) = (\sum_k m_k) \times (\sum_k m_k)$. If, in addition, $\boldsymbol{U}$ and $\boldsymbol{X}$ are dense matrices the complete product $\boldsymbol{X}_k' \boldsymbol{W}(\boldsymbol{U}, \boldsymbol{X}) \boldsymbol{V} \boldsymbol{W} \boldsymbol{X}_k$ is not explicitly available if one ore more regression coefficients are of high dimension.

The INLA approach estimates precision parameters by exploring $p(\boldsymbol{\theta}|\boldsymbol{y})$ as explained in Section 3.3.1. This procedure works well for a moderate size of $\boldsymbol{\theta}$ which is directly related to the number of nonlinear effects in the linear predictor. Usually, this number rarely exceeds, for example, 10. However, there can be situations where the dimension of $\boldsymbol{\theta}$ exceeds a critical limit. For example, in Section 6.2 a linear regression model is formulated over a three-dimensional grid structure with more than 500,000 nodes. In order to account for heteroscedasticity a single dispersion parameter is provided for each node. It is clear from the way $p(\boldsymbol{\theta}|\boldsymbol{y})$ is explored that in this case the INLA approach is not applicable. Note that in this situation the empirical Bayes approach is also not applicable.

### Standard errors of regression coefficients

In most applications standard errors are required in order to assess significance of regression coefficients. In MCMC inference such standard errors can easily be derived by summarizing the generated MCMC samples. Thus, obtaining precise standard errors is mainly a question of the runtime of the MCMC sampler. However, this means that one must be able to (a) generate samples from the full conditional $\tilde{p}(\boldsymbol{\gamma}_k|\boldsymbol{y}, \kappa_k)$ and (b) evaluate acceptance probability (3.9). If the dimension of $\widetilde{\boldsymbol{Q}}_k$ exceeds a certain size it may not be possible to perform these steps on working stations with limited computational resources. The other two inference strategies suffer from similar problems: Within the empirical Bayes approach standard errors are estimated by the diagonal elements of the inverse Fisher information (3.15) which are difficult to compute if one or more regression coefficients are high-dimensional. For the INLA approach marginal variances for the nodes in $\boldsymbol{x}$ can usually be obtained in a recursive manner from the elements of the Cholesky decomposition of $\boldsymbol{Q}$ (Rue and Martino, 2007) which is of dimension $(n + \sum_k m_k) \times (n + \sum_k m_k)$. As will be seen in the remainder of this chapter it is computational challenging to derive this decomposition for a matrix of this size.

### 4.1.3 Conclusion

Summarizing the above discussion it can be concluded that the empirical Bayes approach is the least appropriate inference strategy for large-scale problems. The main reason for this is the necessary representation as a mixed model which does not guarantee sparsity of high-dimensional model components. Missing sparsity is also a concern in the estimation of variance components. In addition, strong limitations with respect to the number of dispersion parameters make this inference approach not suitable for models that are frequently used for the analysis of medical images (see Section 6.2). This does also apply to the INLA approach, although in most other points this inference strategy outperforms the empirical Bayes method with respect to large-scale problems. By far the best candidate for high-dimensional data situations is MCMC based inference. This approach works on the original regression coefficients, ensures sparsity in all steps, and can easily be extended to more general situations. Therefore, the remaining sections and chapters of this thesis are dedicated to the construction of an efficient MCMC sampling scheme that is able to fit large-scale regression models.

## 4.2 Large-scale inference using MCMC

From the previous sections the following bottlenecks can be identified for the MCMC approach: First, obtaining a sample from a high-dimensional proposal distribution. This task can be split into sampling $\boldsymbol{x}$ from the zero-mean Gaussian $\mathrm{N}(\boldsymbol{0}, \widetilde{\boldsymbol{Q}}_k^{-1})$ and subsequently adding $\tilde{\boldsymbol{\mu}}_k$ which is the solution of (3.8), i.e. $\widetilde{\boldsymbol{Q}}_k\tilde{\boldsymbol{\mu}}_k = \boldsymbol{b}_k$. Second, the evaluation of acceptance probability (3.9) requires to compute the log-determinant of $\widetilde{\boldsymbol{Q}}_k$. The third bottleneck is with regard to summarizing results in the absence of MCMC samples. In this section solutions to all of these points are presented. In order to simplify notation the index $k$ is suppressed and $\boldsymbol{Q}$ is used for $\widetilde{\boldsymbol{Q}}$.

### 4.2.1 Sampling from zero-mean Gaussians

#### Direct approach

Usually, sampling from a GMRF is performed by using the Cholesky decomposition of the precision matrix (Rue, 2001). The Cholesky decomposition of $\boldsymbol{Q}$ is given by $\boldsymbol{Q} = \boldsymbol{L}\boldsymbol{L}'$ where $\boldsymbol{L}$ is a lower triangular matrix. A sample $\boldsymbol{x} \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{Q}^{-1})$ can then be obtained by first sampling $\boldsymbol{z}$ from $\mathrm{N}(\boldsymbol{0}, \boldsymbol{I})$ and subsequently solving the linear system $\boldsymbol{L}'\boldsymbol{x} = \boldsymbol{z}$. It can be shown that the number of non-zero off-diagonal elements in $\boldsymbol{L}$, $n_{\boldsymbol{L}}$, depends on the sparsity structure of $\boldsymbol{Q}$. In particular, $n_{\boldsymbol{L}}$ is always greater or equal than the number of non-zero off-diagonal elements in the lower triangular of $\boldsymbol{Q}$, $n_{\boldsymbol{Q}}$ (Rue and Held, 2005, Corollary 2.2). Thus, computational complexity of the Cholesky decomposition can be measured by the fill-in ratio, $R = n_{\boldsymbol{L}}/n_{\boldsymbol{Q}}$.

Over the last decades strategies for an efficient computation of the Cholesky decomposition have been developed. The majority of these methods try to reduce computational complexity by reordering or permuting the elements in $\boldsymbol{Q}$. Here, the main objective is either to reduce $R$, i.e. increase the sparsity structure of $\boldsymbol{L}$, or to reduce the bandwidth of $\boldsymbol{Q}$. The latter is useful as it can be shown that if $\boldsymbol{Q}$ is a band matrix with bandwidth $b$ then this bandwidth is preserved by the Cholesky decomposition (Rue and Held, 2005, Theorem 2.9). Another popular method is nested dissection (George, 1973). In this divide-and-conquer approach the nodes of $\boldsymbol{Q}$ are recursively split and ordered into conditional independent partitions given different sets of separator nodes. In order to illustrate these methods consider precision matrix (4.1) in Section 4.1.1. The leftmost panel in Figure 4.2 shows the location of all non-zero elements of this matrix for $n_x = n_y = n_z = 10$. There are $7{\cdot}n_x{\cdot}n_y{\cdot}n_z{-}2{\cdot}(n_xn_y{+}n_xn_z{+}n_yn_z) = 6{,}400$ non-zero elements and the maximum bandwidth is $n_x{\cdot}n_y = 100$. The number of non-zero elements in the lower triangular of $\boldsymbol{Q}$ is $n_{\boldsymbol{Q}} = 2{,}700$. The next panel displays the same matrix after permuting the nodes by the symmetric approximate minimum degree permutation (SAMD, George and

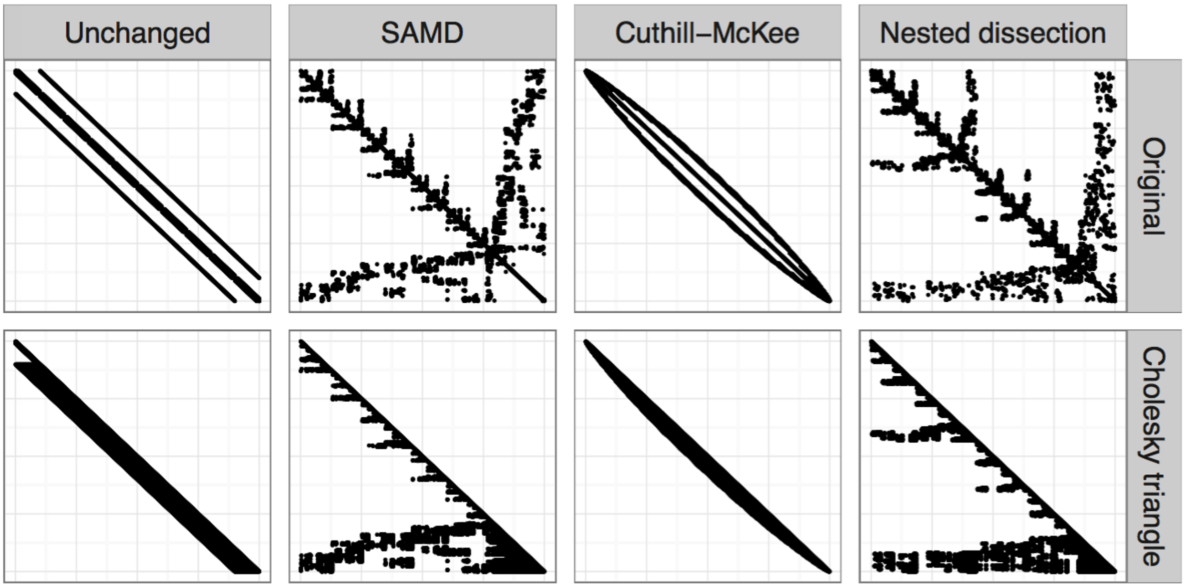


Figure 4.2 Sparsity pattern of precision matrix (4.1). Top row displays the original ordering and the results of different permutation algorithms. Bottom row depicts the sparsity pattern of the corresponding Cholesky factors.

Liu, 1989), as implemented in Matlabs `symamd` function. As an example for a bandwidth reduction algorithm the next panel depicts the sparsity structure of the precision matrix reordered by the Cuthill-McKee algorithm (Cuthill and McKee, 1969), as implemented in Matlabs `symrcm` function. Using this method the maximum bandwidth is reduced to 80. Finally, the rightmost panel shows the sparsity structure after applying the spectral nested dissection ordering (Chan et al., 1995) as implemented in the `meshpart`[1] package. The bottom row of Figure 4.2 depicts the structures of the corresponding Cholesky triangles. For the unmodified matrix one obtains $n_{\boldsymbol{L}} = 90{,}909$, thus $R = 33.67$. The SAMD method yields $R = 11.86$, the Cuthill-McKee algorithm $R = 21.28$, and the nested dissection approach $R = 15.8$. In summary, the best solution that can be obtained by these methods for this specific situation has still about eleven times as many non-zero elements than the original precision matrix. For small graphs this seems manageable. However, note that the fill-in ratio does not remain constant if the dimension of the graph increases. Figure 4.3 displays the relation between the dimension of the graph and the fill-in ratio (left panel) and storage (right panel). It can be seen that both, the fill in ratio as well as storage increase dramatically when the dimension increases. For example, for $n_x = n_y = n_z = 35$ all permutation methods produce Cholesky factors with nearly 100 times as many non-zero

[1] `http://www.cerfacs.fr/algor/Softs/MESHPART/`, visited on January 21st, 2016.

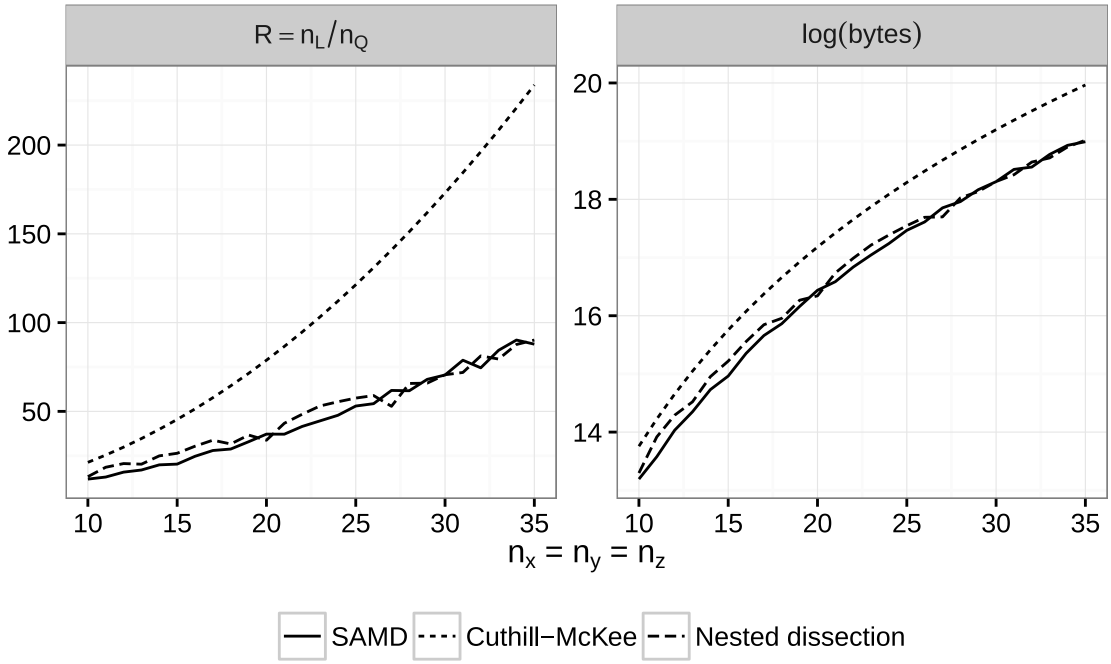


Figure 4.3 Effect of graph dimension on fill-in-ratio and storage of Cholesky factors obtained by different permutation algorithms.

elements as the lower triangular of $\boldsymbol{Q}$. In the application chapter of this thesis the sizes of graphs are considerably larger than in Figure 4.3. For example, in Section 6.3 a graph of size 190 × 190 × 54 with 514,442 active nodes is used for which $n_{\boldsymbol{Q}} = 1{,}517{,}257$. It is clear from Figure 4.3 that, even when using permutation strategies, the factorization of $\boldsymbol{Q}$ would represent a serious problem with respect to computation time and storage. Therefore, the direct approach for sampling from zero-mean Gaussians is not applicable for high-dimensional data situations as considered here.

### Single-site sampler

The Markov property of GMRF priors allows to derive conditional prior distributions for $\gamma_j$ given $\boldsymbol{\gamma}_{-j}$, see for example (2.8) and (2.9) for temporal priors and (2.14) for spatial priors. In general, let $\mathrm{N}(\mu_{j0}, \kappa_{j0}^{-1})$ be the conditional prior for the $j$th element of $\boldsymbol{\gamma}$. Given this formulation it is possible to divide the Metropolis-Hastings updating step given in Section 3.1.1 into $m = \dim(\boldsymbol{\gamma})$ updating steps. This way a one-dimensional proposal density $\mathrm{N}(\tilde{\mu}_j, \tilde{\kappa}_j^{-1})$ is obtained for each element of $\boldsymbol{\gamma}$. Thus, the problem of sampling from high-dimensional Gaussians can be decomposed into the smallest sampling problems possible. Note that $\tilde{\mu}_j$ now depends on the mean of the conditional prior $\mu_{j0}$.

From a chronological point of view single-site MCMC algorithms have greatly contributed to the distribution of Bayesian methods (Besag, et al., 1991). They are easy to implement and do not require a lot of resources. However, these advantages come at a price. Besides the fact that sequential updating of hundreds of thousands of parameters can be quite time consuming, slow mixing due to large dependencies between elements can be a serious issue (Gilks et al., 1996). Therefore, single site updating schemes should only be considered if alternative methods cannot be applied.

### Blocking strategies

As computational power increased in the late 1990's so did the interest in blocking strategies as a way to overcome bad mixing behavior of single site samplers. The main idea of these strategies is to construct a sampler that updates dependent elements in $\boldsymbol{\gamma}$ jointly. Thus, blocks should be created in a way that high dependency can be found within blocks and low dependency between blocks.

A variety of blocking strategies has been published over the last years. While only a few approaches are suited for general MCMC problems most approaches depend on the problem at hand. A blocking algorithm that is of particular interest for GMRFs is the conditional prior proposal approach by Knorr-Held (1999). Here, $\boldsymbol{\gamma}$ is divided into $n_B$ blocks $\boldsymbol{\gamma}_j, j = 1, \ldots, n_B$, for which proposals are generated not by their full conditionals $p(\boldsymbol{\gamma}_j | \boldsymbol{\gamma}_{-j}, \boldsymbol{y}, \kappa)$, but rather by their conditional prior distributions given the other blocks, i.e. $p(\boldsymbol{\gamma}_j | \boldsymbol{\gamma}_{-j}, \kappa)$. One interesting aspect of this approach is that the resulting block proposal does not depend on the current state of the chain for this block. Therefore, when calculating the acceptance probability the proposal density cancels out and the Metropolis-Hastings algorithm reduces to a Metropolis algorithm. Knorr-Held (1999) noted that a deterministic or a random change of block configuration may be necessary in order to guarantee good mixing for parameters near break points. In addition, Brezger and Lang (2006) showed that the IWLS proposal given in Section 3.1.1 outperforms the conditional prior proposals approach with respect to mixing of regression and variance parameters.

Another general blocking strategy has been proposed by Rue (2001). Here, blocks are updated according to their full conditional given all other blocks by using the Cholesky factorization. For the evaluation of acceptance probabilities, the likelihood of the GMRF is approximated by a pseudo likelihood approach. To be more precise, independence is assumed between blocks so that the joint prior of $\boldsymbol{\gamma}$ can be written as $p(\boldsymbol{\gamma}|\kappa) \approx p(\boldsymbol{\gamma}_1|\boldsymbol{\gamma}_{-1}, \kappa) \times \ldots \times p(\boldsymbol{\gamma}_{n_B}|\boldsymbol{\gamma}_{-n_B}, \kappa)$. From this approximation it is obvious that the approach works best for partitions that minimizes dependencies between blocks. Finding such a configuration for general graphs is a non-trivial problem. However, even when such an optimal partition has been found it remains unclear to what extend the approximation error affects final inference. In addition, similar to the approach by Knorr-Held (1999) a change in blocking configuration may be necessary in order to circumvent problems at break points.

Closely related to the above blocking strategies are divide-and-conquer approaches. The one suggested by Rue (2001) is especially suited for GMRFs. Here, the graph is partitioned into blocks that are conditionally independent given a set of separating nodes $\boldsymbol{\gamma}_s$. Again, blocks are updated by their full conditionals. The difference to the blocking strategies above is that the marginal prior distribution for the set of separating nodes is required for which the marginal covariance matrix needs to be computed. This way problems at break points are avoided. However, because of this the divide-and-conquer approach is either limited to situations where this matrix can be handled by direct methods or it requires a complex iterative procedure where the remaining blocks are partitioned in a recursive manner until all marginal covariances are of a certain size.

In summary, blocking strategies can be used to sample from high-dimensional Gaussians if direct methods are not available. In contrast to single-site samplers they usually perform better with respect to mixing and speed. However, finding an appropriate block configuration is a non-trivial task and depends highly on the problem at hand. In addition, implementation can be difficult and a careful bookkeeping of indices is required. Note that the block update of parameters as discussed in Section 3.1.4 requires updating $\boldsymbol{\gamma}$ in one step and, therefore, does not provide a solution for sampling from high-dimensional Gaussians.

### Approximate sampling

Over the last years, much progress has been made with respect to approximate sampling of zero-mean Gaussians. Most promising approaches utilize Krylov subspace methods, a class of iterated methods for sparse linear systems (Liesen and Strakos, 2012). For example, Chow and Saad (2014) discuss approximate sampling from zero-mean Gaussians given their covariance matrices while Aune et al. (2013) and Simpson et al. (2013) present solutions for the case of precision matrices. In general, Krylov subspace methods are able to provide an

**Algorithm 1** Lanczos algorithm.

```
Set $\boldsymbol{v}_0 = 0$ and $\beta_1 = 0$ and initialize $\boldsymbol{v}_1$
for $j = 1, \dots, r$ do
    $\boldsymbol{w} = \boldsymbol{Q}\boldsymbol{v}_j - \beta_j \boldsymbol{v}_{j-1}$
    $\alpha_j = \boldsymbol{w}'\boldsymbol{v}_j$
    $\boldsymbol{w} = \boldsymbol{w} - \alpha_j \boldsymbol{v}_j$
    $\beta_{j+1} = ||\boldsymbol{w}||_2$
    $\boldsymbol{v}_j = \boldsymbol{w}/\beta_{j+1}$
end for
```

approximation to the general problem $f(\boldsymbol{Q})\boldsymbol{b}$ by only using matrix-vector products. Here, $f$ is an arbitrary function, for example $f(\boldsymbol{Q}) = \boldsymbol{Q}^{-1}$ for solving systems of linear equations, or $f(\boldsymbol{Q}) = \boldsymbol{Q}^{-1/2}$ for sampling from $\mathrm{N}(\boldsymbol{0}, \boldsymbol{Q}^{-1})$. This is made possible by projecting the original problem into the much smaller Krylov subspace $\mathcal{K}_r(\boldsymbol{Q}, \boldsymbol{v})$ which is spanned by the Krylov sequence $\boldsymbol{b}, \boldsymbol{Q}\boldsymbol{b}, \boldsymbol{Q}^2\boldsymbol{b}, \dots, \boldsymbol{Q}^{r-1}\boldsymbol{b}$. If $\boldsymbol{V}_r = (\boldsymbol{v}_1, \dots, \boldsymbol{v}_r)$ is an orthogonal basis of $\mathcal{K}_r$ the orthogonal projection of the exact solution on the Krylov subspace is given by

$$\tilde{\boldsymbol{x}} = \boldsymbol{V}_r \boldsymbol{V}_r' f(\boldsymbol{Q})\boldsymbol{b}. \tag{4.2}$$

In case the (modified) Gram-Schmidt orthogonalization is used to build $\boldsymbol{V}_r$ one obtains the Arnoldi algorithm (Saad, 2003, Section 6.3). If in addition $\boldsymbol{Q}$ is Hermitian, the special case of the Lanczos algorithm (Saad, 2003, Section 6.6) is received. This algorithm is given in Algorithm 1. Besides the orthonormal basis $\boldsymbol{V}_r$ of $\mathcal{K}_r$ this algorithm produces coefficients $\alpha_j$ and $\beta_j$, where $j = 1, \dots, r$, which form the tridiagonal matrix $\boldsymbol{T}_r$:

$$\boldsymbol{T}_r = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{r-1} & \alpha_{r-1} & \beta_r \\ & & & \beta_r & \alpha_r \end{pmatrix}. \tag{4.3}$$

This setup satisfies

$$\boldsymbol{Q}\boldsymbol{V}_r = \boldsymbol{V}_r \boldsymbol{T}_r + \beta_{r+1} \boldsymbol{v}_{r+1} \boldsymbol{e}_r'$$

where $\boldsymbol{e}_r$ is the $r$th column of the identity matrix. Note that $\boldsymbol{V}_r'\boldsymbol{V}_r = \boldsymbol{I}$ and $\boldsymbol{V}_r'\boldsymbol{v}_{r+1} = 0$ since $\boldsymbol{V}_r$ is an orthonormal basis. From this it follows immediately that $\boldsymbol{V}_r'\boldsymbol{Q}\boldsymbol{V}_r = \boldsymbol{T}_r$.

In order to solve $\boldsymbol{Q}^{1/2}\boldsymbol{x} = \boldsymbol{z}$ the first vector of $\boldsymbol{V}_r$ is set to $\boldsymbol{v}_1 = \boldsymbol{z}/||\boldsymbol{z}||_2$. Thus, the approximate solution (4.2) can be rewritten as

$$\tilde{\boldsymbol{x}} = \beta \boldsymbol{V}_r \boldsymbol{V}_r' \boldsymbol{Q}^{-1/2} \boldsymbol{V}_r \boldsymbol{e}_1$$

with $\beta = ||\boldsymbol{z}||_2$. The final approximation to $\boldsymbol{Q}^{-1/2}\boldsymbol{z}$ is obtained by further approximating $\boldsymbol{V}_r' f(\boldsymbol{Q})\boldsymbol{V}_r$ by $f(\boldsymbol{V}_r'\boldsymbol{Q}\boldsymbol{V}_r)$, thus

$$\tilde{\boldsymbol{x}}^* = \beta \boldsymbol{V}_r \boldsymbol{T}_r^{-1/2} \boldsymbol{e}_1. \tag{4.4}$$

By using this approximation $f$ only needs to be applied to the much smaller matrix $\boldsymbol{T}_r$ which can be obtained with low computational cost if $r$ is small, which is usually the case. Following Chow and Saad (2014) the algorithm can be stopped if the relative change in $\tilde{\boldsymbol{x}}^*$ falls below a given threshold.

As will be shown throughout the remaining chapters, sampling using the Lanczos algorithm is, compared to other approaches, extremely fast and sufficient precise. In particular, convergence behavior is discussed in Section 4.2.3 and further investigations with respect to approximation errors are given in Chapter 5.

## 4.2.2 Solving systems of linear equations

Once a sample from the zero-mean Gaussian $\mathrm{N}(\boldsymbol{0}, \boldsymbol{Q}^{-1})$ is available all that remains in order to obtain a full sample of the proposal for $\boldsymbol{\gamma}$ is solving $\boldsymbol{Q}\boldsymbol{\mu} = \boldsymbol{b}$. Note that if additional linear constraints as introduced in Section 3.1.4 are present further linear systems need to be solved.

### Direct approach

Computing $\boldsymbol{\mu}$ is usually performed as proposed in Rue (2001), that is, by first solving $\boldsymbol{L}\boldsymbol{v} = \boldsymbol{b}$ and subsequently solving $\boldsymbol{L}'\boldsymbol{\mu} = \boldsymbol{v}$. Here, $\boldsymbol{L}$ is the Cholesky factor obtained by $\boldsymbol{Q} = \boldsymbol{L}\boldsymbol{L}'$. However, with respect to large matrices this method suffers from the same limitations outlined in Section 4.2.1 and is therefore not applicable for situations considered here.

#### Iterative methods for sparse linear systems

For large precision matrices Rue (2001) suggests to avoid the Cholesky decomposition and to use iterative methods for sparse linear systems instead. In particular, he recommends to use the well known conjugate gradient method (Hestenes and Stiefel, 1952), another Krylov subspace methods for solving sparse linear systems (Saad, 2003, Section 6.7). Several alternatives have been proposed in the literature over the years, for example the biconjugate gradient method (Fletcher, 1976), the (generalized) minimum residual method (Paige and Saunders, 1975; Saad and Schultz, 1986), as well as the conjugate gradients squared method (Sonneveld, 1989). In addition, the Lanczos algorithm presented in Section 4.2.1 can also be used for solving linear systems. This can be achieved by using Algorithm 1 and setting $\boldsymbol{v}_1 = \boldsymbol{r}_0/\beta$ with $r_0 = \boldsymbol{b} - \boldsymbol{Q}\boldsymbol{x}_0$ and $\beta = ||\boldsymbol{r}_0||$ (Saad, 2003, Algorithm 6.16). Note that due to different starting configurations of $\boldsymbol{v}_1$ it is not possible to use the same basis $\boldsymbol{V}_r$ and coefficients $\alpha_j$ and $\beta_j$, $j = 1, \ldots, r$ for $f(\boldsymbol{Q}) = \boldsymbol{Q}^{-1/2}$ and $f(\boldsymbol{Q}) = \boldsymbol{Q}^{-1}$. Thus, in order to obtain a sample from (3.6) it is required to run the Lanczos algorithm twice.

### 4.2.3 Preconditioning

The quality and convergence behavior of Krylov subspace methods are strongly connected to the *condition* of $\boldsymbol{Q}$. If $\boldsymbol{Q}$ is not-well conditioned, i.e. if the solution $f(\boldsymbol{Q})\boldsymbol{b}$ is sensitive to small perturbations in $\boldsymbol{Q}$ or $\boldsymbol{b}$, then the iterative procedures presented above may suffer from slow convergence and inexact solutions, even after successful convergence. However, even for problems that are well-conditioned the number of iterations required to obtain a suitable approximation to $f(\boldsymbol{Q})\boldsymbol{b}$ may be inadequately high in order to be of practical use. In both situations the performance of Krylov subspace methods can usually be improved by *preconditioning*. This means to reformulate the original problem as one that has the same solution but is easier to solve (Saad, 2003, Chapter 9).

In this section the general framework of preconditioning is presented, first with respect to linear systems and subsequently for approximate sampling using the Lanczos algorithm. This is followed by a presentation of different preconditioners and a brief discussion on the convergence behavior of Krylov subspace methods.

### Preconditioning for linear systems

With respect to linear systems, a preconditioner for $\boldsymbol{Q}$ is a matrix $\boldsymbol{M}$ for which $\boldsymbol{M}^{-1}\boldsymbol{Q}$ has a better condition than $\boldsymbol{Q}$. Then, instead of the original linear system $\boldsymbol{Qx} = \boldsymbol{b}$ the preconditioned problem $\boldsymbol{M}^{-1}\boldsymbol{Qx} = \boldsymbol{M}^{-1}\boldsymbol{b}$ is solved. If the preconditioner is available in factorized form, i.e. $\boldsymbol{M} = \boldsymbol{M}_L\boldsymbol{M}_R$, the preconditioned system can be split into the following equations

$$\boldsymbol{M}_L^{-1}\boldsymbol{Q}\boldsymbol{M}_R^{-1}\boldsymbol{u} = \boldsymbol{M}_L^{-1}\boldsymbol{b}, \qquad \boldsymbol{x} = \boldsymbol{M}_R^{-1}\boldsymbol{u}.$$

From these equations it is obvious that, in order to solve $\boldsymbol{Qx} = \boldsymbol{b}$ more easily, $\boldsymbol{M}$ must be constructed in a way that the solution to $\boldsymbol{Mx} = \boldsymbol{b}$ can be obtained without much effort. This observation requires $\boldsymbol{M}$ or $\boldsymbol{M}_L$ and $\boldsymbol{M}_R$ to be sparse. Further requirements are that the preconditioner should be in “some sense” (Saad, 2003, p. 275) close to $\boldsymbol{Q}$ and nonsingular. This vague formulation of requirements is the reason that the task of finding an appropriate preconditioner for a given problem is sometimes referred to be “a combination of art, science, and intuition” (Hafez et al., 2010, p. 81). However, with respect to the conjugate gradient method robust preconditioners are available as discussed below.

### Preconditioning for sampling

While preconditioning as discussed so far is extensively used with respect to solving linear systems it may not be obvious that this is also a suitable strategy for sampling from high-dimensional Gaussians, i.e. solving $\boldsymbol{Q}^{1/2}\boldsymbol{x} = \boldsymbol{z}$. This problem is addressed in Chow and Saad (2014) with respect to covariance matrices and in Simpson et al. (2013) for precision matrices. In both articles it is pointed out that if $\boldsymbol{Q}$ and $\boldsymbol{M}$ are symmetric positive matrices with $\boldsymbol{M} = \boldsymbol{M}_L\boldsymbol{M}_R$, $\boldsymbol{M}_R = \boldsymbol{M}_L^T$ and if $\boldsymbol{u} \sim \mathrm{N}(\boldsymbol{0}, (\boldsymbol{M}_L\boldsymbol{Q}\boldsymbol{M}_R)^{-1})$, then the solution of $\boldsymbol{M}_R\boldsymbol{x} = \boldsymbol{u}$ is a sample of a zero-mean Gaussian with precision $\boldsymbol{Q}$. Thus, if $\boldsymbol{M}$ meets the above requirements for preconditioners the sampling process can be simplified by (a) obtaining $\boldsymbol{u}$ from a Gaussian with precision matrix $\boldsymbol{M}_L\boldsymbol{Q}\boldsymbol{M}_R$ using the Lanczos algorithm and (b) computing the final sample via $\boldsymbol{x} = \boldsymbol{M}_R^{-1}\boldsymbol{u}$. Obligatory adjustments in Algorithm 1 are with respect to $\boldsymbol{Q}\boldsymbol{v}_j$ in line 2. This is replaced by first solving $\boldsymbol{M}_L\boldsymbol{a} = \boldsymbol{v}_j$, then calculating $\boldsymbol{b} = \boldsymbol{Q}\boldsymbol{a}$ and finally solving $\boldsymbol{M}_R\boldsymbol{w} = \boldsymbol{b}$. See Section A.2.2 for a complete implementation of the preconditioned Lanczos algorithm.

### Preconditioning techniques

Defining an appropriate preconditioner for a particular problem is a difficult task and an active field of research within the field of numerical linear algebra. Therefore, only an incomplete overview of this topic can be provided here.

From a historical point of view, incomplete factorizations preconditioners of the form $\boldsymbol{M} = \boldsymbol{M}_L \boldsymbol{M}_R$ are the most frequently used type of preconditioners. Within this class incomplete versions of the LU (ILU, Varga, 1960) and Cholesky (IC, Meijerink and van der Vorst, 1977) decompositions are the most popular. Incompleteness of these factorizations can be achieved in different ways. The most generic method is to consider only non-zero elements of $\boldsymbol{Q}$ in $\boldsymbol{M}_L$ and $\boldsymbol{M}_R$. Alternatively, a drop tolerance can be specified that keeps only those values in $\boldsymbol{M}_L$ and $\boldsymbol{M}_R$ that lie above this threshold (Saad, 1994). Since $\boldsymbol{Q}$ is symmetric it makes sense to prefer the IC factorization over the ILU factorization, because then only one matrix needs to be stored. With respect to the conjugate gradient method the IC decomposition serves as a popular and quite robust preconditioner. This choice is also made by Chow and Saad (2014) and Simpson et al. (2013) with respect to approximate sampling using the Lanczos algorithm.

Over the last years approximate inverse preconditioners have been proposed as alternatives to incomplete factorizations, see Benzi and Tuma (1999) for a review on early methods. Here, $\boldsymbol{P} = \boldsymbol{M}^{-1}$ is chosen in a way that it approximates the most important aspects of $\boldsymbol{Q}^{-1}$. A large class of such methods consists of iterative procedures that rely on the minimization of $||\boldsymbol{I} - \boldsymbol{Q}\boldsymbol{P}||_F$, where $|| \cdot ||_F$ denotes the Frobenius norm. The well known Newton-like iterations by Schulz (1933) and the (global) minimal residual algorithm by Chow and Saad (1998) provide popular examples within this class of methods. Since the number of non-zero elements in $\boldsymbol{P}$ increases with each iteration of these methods, some sort of dropping rule needs to be applied in order to preserve sparsity. As an alternative to iterative methods approximate inverses can be constructed by utilizing the above discussed incomplete factorizations preconditioners. Here, the task of finding an approximate inverse for $\boldsymbol{Q}$ is divided in approximating the inverse of $\boldsymbol{M}_L$ and $\boldsymbol{M}_R$ which are derived from an ILU decomposition. As Benzi and Tuma (1999) pointed out, a serious disadvantage of this procedure is the introduction of multiple levels of incompleteness, one for the ILU decomposition and one for the approximate inverse of $\boldsymbol{M}_L$ and $\boldsymbol{M}_R$.

As stated at the beginning, this section does not claim to provide a complete overview of preconditioning techniques. Actually, quite the contrary is the case: Important classes such as polynomial (O'Leary, 1991; Fischer and Freund, 1994) and spectral (Giraud and Gratton, 2006) preconditioners are not discussed. More recently, adaptive preconditioning techniques that are able to update the preconditioner in each iteration based on the Krylov subspace formulated so far have been studied (van den Eshof and Hochbruck, 2006; Ilić et al., 2008). Such approaches have also been used within approximate sampling from zero mean Gaussians (Simpson et al., 2007). However, other authors reported that under realistic conditions, i.e. finite-precision arithmetic, adaptive methods often do not perform better than usual preconditioners (Monteiro et al., 2004) or become unstable (Saad, 2003, Section 9.4). In addition, the more specialized a preconditioner becomes the less applicable it is for general situations. Therefore, a detailed discussion of more advanced preconditioners is not pursued at this point.

### Convergence behavior

Convergence behavior of Krylov subspace methods is usually analyzed by providing upper bounds for the error $||f(\boldsymbol{Q})\boldsymbol{b}-\boldsymbol{x}_m^*||$, where $\boldsymbol{x}_m^*$ is the approximate solution to $f(\boldsymbol{Q})\boldsymbol{b}$ after $m$ iterations (Saad, 2011, Section 6.11). Most error bounds that can be found in the literature are concerned with special choices of $f$, such as $f(x) = x^{-1}$ or $f(x) = \exp(x)$. One of the few generalizations is provided by Ilić et al. (2010) who prove that the following error bound holds for all transforms $f$ which are of particular interest here, i.e. $f(x) = x^{-1}$ and $f(x) = x^{1/2}$:

$$||f(\boldsymbol{Q})\boldsymbol{b} - \boldsymbol{x}_m^*|| \leq f(\lambda_{\min})||\boldsymbol{r}_m||. \tag{4.5}$$

In this notation, $\lambda_{\min}$ is the smallest eigenvalue of $\boldsymbol{Q}$ and $\boldsymbol{r}_m$ is the residual obtained after applying $m$ steps of the conjugate gradient method for solving $\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{b}$. The advantage of this result is that the convergence behavior of the conjugate gradient method can directly be assigned to the Lanczos algorithm for approximate sampling by setting $f(\lambda_{\min}) = \lambda_{\min}^{-1/2}$ (Simpson et al., 2013). According to this error bound, convergence is linear with respect to the number of iterations, although, in practice "super-linear" convergence is usually observed (Chow and Saad, 2014). However, as pointed out by Chow and Saad (2014) this and further existing error bounds do not consider improved convergence which usually follows from preconditioning. It is therefore expected that the convergence performance for preconditioned systems is even better.

A disadvantage of error bounds like (4.5) is that only statements about the absolute error between the true and approximate solution can be made. This may not account for convergence problems of GMRFs with high precision where realizations tend to be be highly smoothed and are often heavily shrinked towards zero. In this case, the absolute error becomes small rather fast while the relative error still shows large discrepancies. Thus, convergence may be indicated by theoretical error bounds although the approximate solution is far away from the true solution. Situations where this problem may occur are further discussed in Chapter 5.

### 4.2.4 Calculation of log-determinants

For Gaussian distributed response variables the proposal density of $\boldsymbol{\gamma}$ corresponds to the full conditional posterior, thus, the MCMC algorithm reduces to a Gibbs sampler. In this case, computation of the log-determinant of $\boldsymbol{Q}$, $\log\det(\boldsymbol{Q})$, is not necessary. For non-Gaussian responses, however, evaluation of acceptance probability (3.9) is required for which the normalizing constant of the proposal density is an important component. Therefore, computation of $\log\det(\boldsymbol{Q})$ is necessary in this case. If $\boldsymbol{\gamma}$ and $\kappa$ are updated jointly as discussed in Section 3.1.4, the log-determinant is required in any case, no matter what response family is considered.

#### Direct approach

Similar to sampling, the Cholesky factor plays a major role within the "direct" approach of computing $\log\det(\boldsymbol{Q})$, as the determinant of a triangular matrix equals the product of its diagonal elements (Harville, 1997, Section 13.1c). Thus, $\log\det(\boldsymbol{Q}) = \log\det(\boldsymbol{L}) + \log\det(\boldsymbol{L}')$ can be written as

$$\log\det(\boldsymbol{Q}) = 2\sum_i \log L_{ii}. \tag{4.6}$$

As shown in the previous sections, if $\boldsymbol{Q}$ exceeds a certain size, it may not be possible to compute the Cholesky factorization. Thus, alternative strategies need to be considered.

### Approximation of log-determinants

Methods that try to approximate (log-)determinants started to emerge in the early 1990s. Key to most of these methods is the identity $\log\det(\boldsymbol{Q}) = \mathrm{tr}(\log \boldsymbol{Q})$, where $\log \boldsymbol{Q}$ denotes the matrix logarithm. Martin (1992) was the first who used Taylor series expansion of the trace of the matrix logarithm. Bai et al. (1996) used this identity to formulate a Monte Carlo approach that yields an estimate as well as lower and upper bounds for the log-determinant. Their method makes extensive use of Gaussian quadrature and related theory. Later, the same group provided deterministic bounds that are not as precise but less computational demanding (Bai and Golub, 1996). Nevertheless, the main advantage of stochastic estimators, i.e. the fact that their accuracy can easily be improved by increasing the sample size, led to the development of further Monte Carlo methods for the approximation of log-determinants. For example, Thron et al. (1996, 1998) provided a stochastic estimator for $\mathrm{tr}(\log \boldsymbol{Q})$ by combining the Padè approximation of the matrix logarithm with the so called complex $Z_2$ noise trace estimator. Another Monte Carlo algorithm was proposed by Barry and Pace (1999) for the special case of spatial weight matrices as they occur within spatial econometric models (LeSage and Pace, 2009). Reusken (2001) constructed a sparse approximate inverse (Cosgrove et al., 1992) of the Cholesky triangle $\boldsymbol{L}$ and used its diagonal elements to formulate a deterministic approximation for $\det(\boldsymbol{Q})^{1/n}$. Another deterministic approach was proposed by Pace and LeSage (2004). They replaced the Taylor series expansion of $\mathrm{tr}(\log \boldsymbol{Q})$ used by Martin (1992) with Chebyshev polynomials (Mason and Handscomb, 2002) which yield more precise approximations. For dense covariance matrices within Gaussian processes, Zhang and Leithead (2007) extended the Taylor series expansion by a set of compensation schemes which includes stochastic trace estimation by random seeds. This method was further improved by Zhang et al. (2008) who used uniformly distributed seeds instead of Gaussian seeds. Another stochastic estimate for the special case of GMRFs has been provided by Aune et al. (2014). Here, $\mathrm{tr}(\log \boldsymbol{Q})$ is estimated by combining Cauchy's integral formula for the computation of the matrix logarithm, Krylov subspace methods for solving linear systems, and stochastic estimators for traces. In order to be applicable this approach requires careful coloring of the adjacency graph of $\boldsymbol{Q}$. More recently, Han et al. (2015) improved the Chebyshev expansion by Pace and LeSage (2004) by estimating traces of huge matrices using the stochastic Hutchinson estimator (Hutchinson, 1990), which allows to use higher order Chebyshev polynomials resulting in a more precise approximation. The combination of this with the general advantages of Monte Carlo estimates makes this approach a suitable candidate for the application within large-scale problems. Therefore, in the following, this method is explained in more detail as it is used for the approximation of log-determinants in the remainder of this thesis.

The approach by Han et al. can be roughly divided into two steps: First, the log-determinant is approximated by a Chebyshev expansion. In the second step, traces that arise in this representation are estimated by the stochastic Hutchinson estimator. For the Chebyshev expansion of the log-determinant first kind Chebyshev polynomials of degree $p$ are used (Mason and Handscomb, 2002, Section 1.2.1). In general, Chebyshev expansions are characterized by an excellent approximation of functions $f(x)$ with $x \in [-1, 1]$. This is accomplished by a combination of a sequence of orthogonal polynomials and non-equidistant nodes that are well distributed over the interval $[-1, 1]$. In order to apply the Chebyshev expansion for the estimation of $\log\det(\boldsymbol{Q})$ Han et al. consider the matrix $\boldsymbol{A} = \boldsymbol{I} - \boldsymbol{Q}$ instead of $\boldsymbol{Q}$ for which

$$\begin{aligned} \log\det(\boldsymbol{Q}) &= \log\det(\boldsymbol{I} - \boldsymbol{A}) \\ &= \sum_i \log(1 - \lambda_i) \end{aligned} \tag{4.7}$$

holds. Thus, $f(x_i) = \log(1 - x_i)$ with $x_i = \lambda_i$. Here, $\lambda_1, \ldots, \lambda_m$ are the eigenvalues of $\boldsymbol{A}$. The necessary condition of $x_i \in [-1, 1]$ can be achieved by dividing all elements of $\boldsymbol{Q}$ by $\delta_\sigma = \sigma_{\min} + \sigma_{\max}$, where $\sigma_{\min}$ and $\sigma_{\max}$ are the extreme eigenvalues, i.e. the minimal and maximal eigenvalues of $\boldsymbol{Q}$. From this it follows that $\lambda_i \in [0, 1]$. A procedure for estimating extreme eigenvalues of positive definite matrices based on the Lanczos algorithm is presented in Section A.3.2. Once the eigenvalues of $\boldsymbol{A}$ have been standardized the components of (4.7) can be approximated by Chebyshev expansions: $\sum_i \log(1 - \lambda_i) \approx \sum_i p_n(\lambda_i)$. Here, $p_n(\lambda_i) = \sum_{j=0}^{p} c_j T_j(\lambda_i)$ where $T_j$ denotes the first kind Chebyshev polynomial of degree $j$, and $c_j$ the corresponding coefficient. By setting $T_0(\lambda) = 1$ and $T_1(\lambda) = \lambda$ the polynomials can be recursively defined as $T_{j+1}(\lambda) = 2\lambda T_j(\lambda) - T_{j-1}(\lambda)$, $j \geq 1$. The coefficients are given by

$$c_j = \frac{1 + I(j > 0)}{p + 1} \sum_{k=0}^{p} \log(1 - x_k) T_j(x_k)$$

with $x_k = \cos\left(\frac{\pi(k+1/2)}{p+1}\right)$for $k = 0, \ldots, p$. Since these coefficients do not depend on $i$ one can write

$$\begin{aligned}\sum_i \log(1 - \lambda_i) &\approx \sum_i \sum_{j=0}^{p} c_j T_j(\lambda_i)\\ &= \sum_{j=0}^{p} c_j \sum_i T_j(\lambda_i).\end{aligned}$$

The next and last step makes use of the fact that the sum of eigenvalues equals the trace of a matrix (Harville, 1997, Section 21.6) and that this result can directly be expanded to matrix polynomials. Thus, the final Chebyshev expansion can be written as

$$\log\det(\boldsymbol{Q}) \approx \sum_{j=0}^{p} c_j \mathrm{tr}(T_j(\boldsymbol{A})). \tag{4.8}$$

The most expensive step within this approximation is the calculation of $\mathrm{tr}(T_j(\boldsymbol{A}))$: Due to its recursive nature $T_j(\boldsymbol{A})$ includes powers of $\boldsymbol{I} - \boldsymbol{Q}$ which may be time consuming or even impossible to obtain. Starting point for an approximation to $\mathrm{tr}(T_j(\boldsymbol{A}))$ is to write the exact solution as

$$\mathrm{tr}(T_j(\boldsymbol{A})) = \sum_{i=1}^{n} \boldsymbol{e}_i' T_j(\boldsymbol{A}) \boldsymbol{e}_i. \tag{4.9}$$

Here, $\boldsymbol{e}_i$ is the $i$th column of the $n$-dimensional identity matrix. Hutchinson (1990) proposed to approximate (4.9) by the following stochastic estimator: Let $\boldsymbol{u}_l$ be a $n$-dimensional random vector with entries $\{-1, 1\}$, where each state has probability 1/2. Then,

$$\widehat{\mathrm{tr}}(T_j(\boldsymbol{A})) = \frac{1}{r} \sum_{l=1}^{r} \boldsymbol{u}_l' T_j(\boldsymbol{A}) \boldsymbol{u}_l \tag{4.10}$$

is an unbiased estimator of (4.9). In addition, Hutchinson showed that (4.10) has the smallest variance of all such stochastic estimators. In order to avoid direct computation of $T_j(\boldsymbol{A})$ its recursive definition can be exploited. That is, the vector $\boldsymbol{w}_{j,l} = T_j(\boldsymbol{A})\boldsymbol{u}_l$ can be recursively defined by $\boldsymbol{w}_{j+1,l} = 2\boldsymbol{A}\boldsymbol{w}_{j,l} - \boldsymbol{w}_{j-1,l}$. By combining this with (4.8) the final approximation of the log-determinant is given by

$$\log\det(\boldsymbol{Q}) \approx \sum_{j=0}^{p} c_j \frac{1}{r} \sum_{l=1}^{r} \boldsymbol{u}_l' \boldsymbol{w}_{j,l}.$$

### A modified sampling scheme

The evaluation of acceptance probability (3.9) requires to compute $\log\det(\boldsymbol{Q})$ because $\boldsymbol{Q}$ depends on the current state of the Markov chain, i.e. in the notation of Section 3.1.1: $\widetilde{\boldsymbol{Q}}^p = \widetilde{\boldsymbol{Q}}^p(\boldsymbol{\gamma}^c)$. This insight may help to provide an alternative solution to the approximation of log-determinants: If it were possible to free $\widetilde{\boldsymbol{Q}}^p$ from the dependency of $\boldsymbol{\gamma}^c$ the log-determinant in the acceptance probability would cancel itself out. The source of this dependency can be found in the construction of the GMRF proposal. Here, the log-likelihood is approximated by a quadratic Taylor expansion around $\boldsymbol{\gamma}^c$. Thus, by choosing a different point around which the series is expanded the desired independence between $\widetilde{\boldsymbol{Q}}^p$ and the current state of the Markov chain can be achieved. A natural candidate for this point is the mean of the last accepted proposal density, which will be denoted by $\tilde{\boldsymbol{\mu}}^*$. Besides the fact that this simple modification eliminates the necessity to compute $\log\det(\boldsymbol{Q})$, it also avoids the expensive re-computation of the mean and precision matrix that are usually required for the evaluation of the acceptance probability.

For the special case of GLMs this modification corresponds to replacing $\boldsymbol{\gamma}^c$ by $\tilde{\boldsymbol{\mu}}^*$ in $\boldsymbol{\eta}^c$:

$$\boldsymbol{\eta}^* = \boldsymbol{\eta}^c + \boldsymbol{Z}(\tilde{\boldsymbol{\mu}}^* - \boldsymbol{\gamma}^c).$$

This approach was proposed by Brezger and Lang (2006) in order to increase acceptance rates. As Brezger and Lang pointed out, high acceptance rates can be a desirable feature for the application to large spatial effects, because the MCMC sampler becomes more sensitive to small changes in the coefficients.

Note that when updating $\boldsymbol{\gamma}$ and $\kappa$ jointly this modification cannot be applied since $\boldsymbol{Q}$ now depends on the current state of the chain through $\kappa$. Thus, approximating $\log\det(\boldsymbol{Q})$ is obligatory in this case.

## 4.2.5 On-line calculation of posterior moments

Saving thousands of samples of high-dimensional regression coefficients can be problematic with regard to computational memory. Algorithms for the on-line calculation of posterior moments offer an attractive alternative in these situations, especially since the marginal posteriors of regression coefficients can usually be quite well approximated by Gaussian distributions which, in turn, are fully parameterized by their mean and variance.

On-line calculation of the mean and variance is rather simple given the nature of their composition. However, to avoid numerical problems the method proposed by Welford (1962) for the calculation of first and second moments can be used. The vector of means, $\bar{\boldsymbol{\gamma}}_t$, and variances, $\boldsymbol{s}_t^2 = \boldsymbol{s}_t/(t-1)$, of $\boldsymbol{\gamma}$ for the first $t$ samples $\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_t$ can be obtained by

$$\bar{\boldsymbol{\gamma}}_t = \bar{\boldsymbol{\gamma}}_{t-1} + \tfrac{1}{t}(\boldsymbol{\gamma}_t - \bar{\boldsymbol{\gamma}}_{t-1}) \tag{4.11}$$

$$\boldsymbol{s}_t = \boldsymbol{s}_{t-1} + (\boldsymbol{\gamma}_t - \bar{\boldsymbol{\gamma}}_{t-1}) \circ (\boldsymbol{\gamma}_t - \bar{\boldsymbol{\gamma}}_t), \tag{4.12}$$

where $\circ$ denotes the Hadamard product. Thus, the marginal posterior for the $j$th element of $\boldsymbol{\gamma}$ after $t$ samples can be approximated by $\mathrm{N}(\bar{\gamma}_{j,t}, s_{j,t}^2)$.

If more than one chain is generated the results obtained by (4.11) and (4.12) can be combined. Suppose $m$ chains with equal length $T$ are available and that $\bar{\boldsymbol{\gamma}}^{(l)}$ and $\boldsymbol{s}^{2(l)}$ denote the vectors of means and variances for the $j$th chain, respectively. These moments can then be aggregated as follows:

$$\bar{\boldsymbol{\gamma}} = \frac{1}{m}\sum_{l=1}^{m} \bar{\boldsymbol{\gamma}}^{(l)}$$

$$\boldsymbol{s}^2 = \frac{T-1}{Tm-1}\left(\sum_{l=1}^{m} \boldsymbol{s}^{2(l)} + \frac{T}{T-1}\sum_{j=1}^{m}(\bar{\boldsymbol{\gamma}}^{(l)} - \bar{\boldsymbol{\gamma}})^2\right).$$

### Combinations of regression coefficients

An important advantage of Bayesian inference based on MCMC over other inference strategies is the possibility to conduct inference for any combination of parameters by simply aggregating the corresponding samples in the desired way. Having access only to marginal posterior moments does not necessarily constitute a serious limitation. For example, suppose $\boldsymbol{\gamma}_1$ and $\boldsymbol{\gamma}_2$ are regression coefficients defined over the same graph and that the marginal posteriors for the $j$th elements of these vectors are given by $\mathrm{N}(\bar{\gamma}_{1,j}, s_{1,j}^2)$ and $\mathrm{N}(\bar{\gamma}_{2,j}, s_{2,j}^2)$, respectively. If one is interested in a linear combination of $\boldsymbol{\gamma}_1$ and $\boldsymbol{\gamma}_2$, for example $\boldsymbol{\gamma}_{1+2} = \boldsymbol{\gamma}_1 + \boldsymbol{\gamma}_2$, the marginal posteriors alone are not sufficient to obtain the marginal posterior for $\gamma_{1+2,j}$. Instead, the covariance between $\boldsymbol{\gamma}_1$ and $\boldsymbol{\gamma}_2$, $\mathrm{cov}(\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2) = \boldsymbol{s}_{1,2}^2$, is required in addition:

$$\gamma_{1+2,j} | \boldsymbol{y}, \kappa_1, \kappa_2 \sim \mathrm{N}(\bar{\gamma}_{1,j} + \bar{\gamma}_{2,j}, s_{1,j}^2 + s_{2,j}^2 + 2s_{1,2,j}^2).$$

Similar to (4.11) and (4.12) the covariance can be computed on-line in a recursive manner given the first $t$ samples of $\boldsymbol{\gamma}_1$ and $\boldsymbol{\gamma}_2$:

$$\boldsymbol{s}_{1,2,t} = \boldsymbol{s}_{1,2,t-1} + (\boldsymbol{\gamma}_{1,t} - \bar{\boldsymbol{\gamma}}_{1,t-1}) \circ (\boldsymbol{\gamma}_{2,t} - \bar{\boldsymbol{\gamma}}_{2,t}).$$

Note that this strategy cannot be applied to nonlinear combinations of regression coefficients $f(\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_p)$. Here, the best possibility to derive an approximation to the desired posterior marginal is to combine the involved coefficient vectors during the fitting process and to compute appropriate summary statistics on-line. However, in general, for nonlinear combinations it cannot be assumed that the posterior marginal can be accurately approximated by its mean and variance. As an alternative one can split the support of $f$ into bins and count how often a sample falls in each bin. This way, the marginal posteriors can be approximated by interpolating the resulting histogram. Note that even for high-dimensional coefficients this method does not require large computational resources. For example, dividing the support of $f(\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_p)$ into 100 bins requires about 76 MB of storage for coefficients with 100,000 elements. However, a vague idea of the domain of the parameter may be helpful in order to set up the bins in an efficient way.

### Convergence assessment

Note that, although the full chains of regression coefficients may not be available, posterior moments are sufficient to perform convergence diagnostics using the potential scale reduction factor as explained in Section 3.1.4. In addition, complete samples of precision and dispersion parameters as well as of randomly selected components of high-dimensional regression coefficients may be saved and monitored visually.

## 4.3 Chapter summary

In this chapter, MCMC inference has been chosen as an appropriate candidate for fitting STAR models with high-dimensional regression coefficients. Solutions to all potential bottlenecks have been provided: Iterative methods for sparse linear systems have been discussed as a suitable way to sample from high-dimensional Gaussians and alternatives to the computation of log-determinants of high-dimensional precision matrices have been shown. This includes an approximation technique as well as a modified sampling scheme. In addition, storage problems can be avoided through the on-line calculation of posterior moments.

# 5 Simulation studies

The goal of this chapter is to analyze the performance of the methods proposed in Chapter 4. It is organized into two sections: The first part examines the error and quality of specific approximations. Its primary task is to identify situations in which the performance of these approximations is limited. Subsequently, complete MCMC sampling schemes are formulated whose performance is then further investigated under realistic conditions in the second part. Here, the error that is induced by these sampling schemes on the final result is of particular interest. For both sections, data sets are simulated which are adapted to the applications in Chapter 6.

A simulation study similar to the dense data setup below has been published in Schmidt et al. (2017). Furthermore, a modified version of the first part of Section 5.1 can be found in the discussion of Schmidt et al.

## 5.1 Performance of approximations

In Section 4.2.3 it has been discussed that the quality of Krylov subspace methods depends on the condition of the system under consideration which is usually imposed by the condition of $\widetilde{\boldsymbol{Q}}$. Subsequently, preconditioning techniques have been presented that are able to improve the condition of the problem at hand. However, in some situations the benefit of preconditioners is limited. In general, the condition of $\widetilde{\boldsymbol{Q}}$ depends on many factors. Those of most relevance can be identified from the definition of $\widetilde{\boldsymbol{Q}}$ given in equation (3.7), i.e.

$$\widetilde{\boldsymbol{Q}} = \boldsymbol{Q} + \boldsymbol{C}.$$

Here, $\boldsymbol{Q} = \kappa \boldsymbol{K}$ is the precision matrix of the GMRF prior, and $\boldsymbol{C}$ is a diagonal matrix with entries $-\partial^2 l(\boldsymbol{\gamma}_j^c)/(\partial^2 \boldsymbol{\gamma}_j)$ where $j = 1, \ldots, m$, with $m = \dim \boldsymbol{\gamma}$. The first factor that has a subsequent influence on the condition of $\widetilde{\boldsymbol{Q}}$ is $\kappa$. This parameter is mainly responsible for the amount of smoothness between the elements in $\boldsymbol{\gamma}$. As already discussed in Section 4.2.3, a high amount of smoothness may shrink the elements in $\boldsymbol{\gamma}$ strongly towards zero or towards an overall population effect. Thus, even small perturbations in the corresponding linear systems may have a relatively large impact on the systems solution. In such situations, the effect of preconditioning may be limited.

The second factor that affects the condition of $\widetilde{\boldsymbol{Q}}$ is given by the diagonal matrix $\boldsymbol{C}$. Its elements are given by the negative second derivative of the log-likelihood around the current state of the chain, $\boldsymbol{\gamma}^c$. The more information in the data the more pronounced the curvature of the log-likelihood, thus the larger the elements in $\boldsymbol{C}$. The larger the elements are in $\boldsymbol{C}$, the more $\widetilde{\boldsymbol{Q}}$ tends to be diagonally dominant which is associated with well-conditioned problems (Saad, 1995). Accordingly, less information in the data increases the risk of creating a matrix which is nearly non-diagonal dominant[1]. In this context, Saad noted, with respect to approximate inverse preconditioners, that "in the non-diagonal dominant case, we do not know in advance whether or not here exists indeed an approximate inverse which is sparse enough to be practically useful." (Saad, 1995, page 13). Thus, worse performance of preconditioners and, therefore, limited applicability of Krylov subspace methods must be expected in cases where diagonal dominance is not well-pronounced.

In this section, the performance of specific approximations is assessed while accounting for the sources which affect the condition of $\widetilde{\boldsymbol{Q}}$: First, different amounts of information are considered by setting up two different simulations, one where the information is rather dense and one with fairly sparse information content. Second, the effect of the precision parameter is considered by fixing $\kappa$ to values on a pre-specified grid. In order to account for the complex interrelation between model parameters the approximation techniques are evaluated within complete MCMC algorithms. The simulation setups are explained next.

### 5.1.1 Simulation setups

#### Dense data situation

This setup is adapted to the data used in the voxel-wise regression application in Section 6.2. For each subject response values are generated that are aligned over a regular lattice (image) of dimension $n_x = n_y = 120$. Images for $n = 100$ subjects are generated, thus the

[1] Note that due to the strictly positive elements in $\boldsymbol{C}$ and the structure of the precision matrices discussed in Section 2.2.1, $\widetilde{\boldsymbol{Q}}$ will always be diagonally dominant.

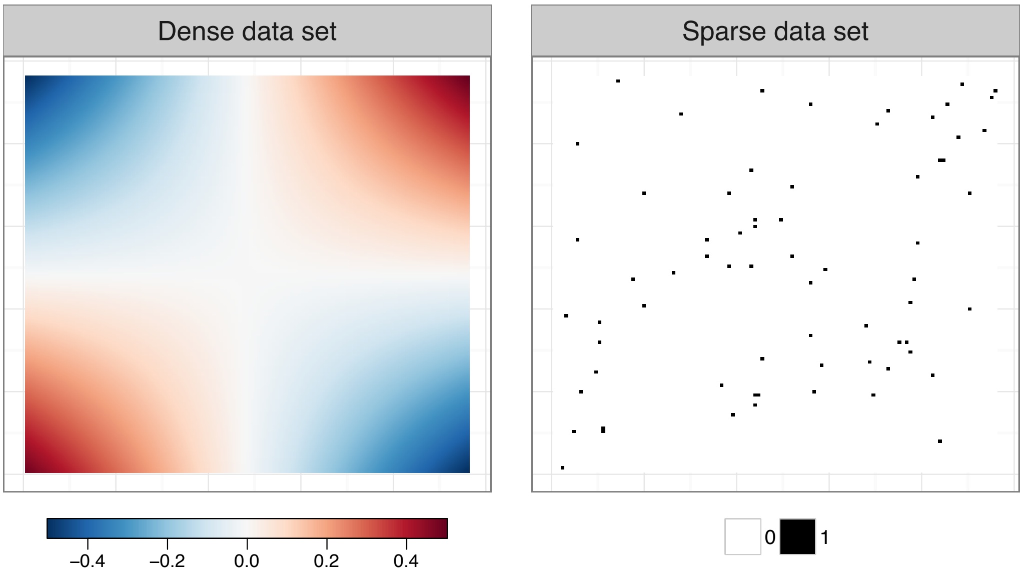


Figure 5.1 True effect image used for the dense data situation (right panel) and generated data for the sparse data situation (left panel).

complete data consist of $n \cdot n_x \cdot n_y$ = 1,440,000 observations in total. Gaussian response values for the $j$th pixel of the $i$th subject are simulated according to the following rule:

$$y_{ij} \sim \mathrm{N}(\eta_{ij}, \kappa_y^{-1}), \text{ with } \kappa_y = 5.0.$$

The linear predictor for the $j$th pixel $\boldsymbol{\eta}_j = (\eta_{1j}, \dots, \eta_{nj})'$ is given by

$$\boldsymbol{\eta}_j = f(j)\boldsymbol{z} + \boldsymbol{X}\boldsymbol{\beta}.$$

Here, $\boldsymbol{z}$ is a metric covariate of dimension $n \times 1$ whose values are randomly assigned over the interval $[-1, 1]$, and $f$ is a smooth function over the pixels of the complete lattice which is created by the following formula:

$$f(j) = (j_x - \tfrac{n_x}{2})(j_y - \tfrac{n_y}{2}).$$

In this notation, $j_x$ and $j_y$ refer to the coordinates of the $j$th pixel of the lattice with respect to the rows and columns, respectively. The values of this effect are linearly scaled to the interval $[-0.5, 0.5]$. The left panel in Figure 5.1 displays the resulting true effect image. Furthermore, the $n \times 2$ matrix $\boldsymbol{X}$ is a fixed design matrix which consists of an intercept and a randomly sampled dummy variable. The corresponding fixed effect $\boldsymbol{\beta}$ is set to $\boldsymbol{\beta} = (3.8, -0.2)'$.

### Sparse data situation

In the microscopy application in Section 6.3, a spatial effect is fitted to a three-dimensional microscopy image of a cell nucleus. The challenge of this data set is that only about 0.63% of the image voxels are non-zero. From the above discussion it can be expected that this large amount of sparseness complicates the successful application of the proposed approximation strategies. Therefore, an additional data set is simulated for which similar difficulties can be expected. To this end, the value 1 is assigned randomly to about 0.7% of the pixels of a regular lattice with dimension $n_x = n_y = 120$. The resulting image is displayed in the right panel of Figure 5.1.

## 5.1.2 Modeling

In order to recover the smooth and fixed effects from the generated dense data set the following regression model is set up for the $j$th pixel:

$$\boldsymbol{y}_j \sim \mathrm{N}(\boldsymbol{\eta}_j, \kappa_y^{-1}\boldsymbol{I}_n)$$

with

$$\boldsymbol{\eta}_j = \gamma_j \boldsymbol{z} + \boldsymbol{X}\boldsymbol{\beta}.$$

As a prior for the smooth effect $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_m)'$, with $m = n_x \cdot n_y$, the two-dimensional extension of the RW1 prior as discussed in Section 2.2.1 is used, that is, the structure matrix of the GMRF prior is given by the following Kronecker-sum penalty:

$$\boldsymbol{K}_{xy} = \boldsymbol{K}_y \otimes \boldsymbol{I}_{n_x} + \boldsymbol{I}_{n_y} \otimes \boldsymbol{K}_x. \tag{5.1}$$

Due to identificational reasons, sum-to-zero constraints are applied to this effect. Considering the fixed effects in $\boldsymbol{\beta}$ a zero-mean gaussian prior with precision matrix $1e^{-6}\boldsymbol{I}_2$ is assigned to this parameter.

For the sparse data situation a spatial Poisson model is fit to the data. Here, it is assumed that the realization of the response value for the $j$th pixel can be described by a Poisson distribution with pixel-specific expectation $\lambda_j$:

$$y_j \sim \mathrm{Po}(\lambda_j)$$

with

$$\lambda_j = \exp(\beta + \gamma_j).$$

Priors for the fixed and smooth effects are the same as for the dense data situation above.

As outlined in the beginning of this section the condition of $\widetilde{\boldsymbol{Q}}$ depends on two sources: the amount of information in the data and the magnitude of the precision parameter. In this simulation study, the first source is controlled by considering a dense and a sparse data set. In order to account for the second source the above models are estimated with different but fixed choices for $\kappa$. In particular, the values $\exp(0), \exp(2), \exp(4)$, and $\exp(8)$ are used.

Approximate sampling is performed as follows: The preconditioned conjugate gradient and the preconditioned Lanczos algorithm are used to obtain the mean and a sample from the zero mean Gaussian, respectively. Sum-to-zero constraints are applied by solving $\widetilde{\boldsymbol{Q}}\boldsymbol{V} = \boldsymbol{A}'$ using preconditioned conjugate gradient and subtracting (3.12) from the proposal. For all Krylov subspace methods the convergence tolerance is set to $1e^{-4}$. As a preconditioner the incomplete Cholesky factorization with a drop tolerance (ICT) is used. In order to analyze the effect of this drop tolerance on the approximated random sample different values are used for this threshold.

With respect to the evaluation of log-determinants the following two strategies are pursued: First, the stochastic Chebyshev estimator by Han et al. (2015) as discussed in Section 4.2.4 is applied. For implementation details see Section A.3. Here, the performance of different degrees of the Chebyshev polynomials as well as different sample sizes of the stochastic trace estimator are analyzed. Second, the Cholesky factor of $\widetilde{\boldsymbol{Q}}$ is approximated by the ICT factorization and the log-determinant is subsequently calculated by formula (4.6).

### 5.1.3 Validation

The moderate size of $\boldsymbol{\gamma}$ in both data sets allows the calculation of the full Cholesky decomposition. Thus, results from approximate sampling and approximate solutions for log-determinants can be compared to their "true" counterparts, i.e. the solutions obtained by the full Cholesky decomposition. To this end the above models are estimated by the MCMC sampling scheme presented in Section 3.1 with $\kappa$ fixed to the values given above. The true and approximate solutions are saved for 1,000 iterations. In each iteration, the same random vector $\boldsymbol{z} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{I})$ is used for direct and approximate sampling. This allows a direct evaluation of the approximation error with respect to sampling using the relative error

$$\delta^2(\boldsymbol{\gamma}^p, \tilde{\boldsymbol{\gamma}}^p) = \frac{||\boldsymbol{\gamma}^p - \tilde{\boldsymbol{\gamma}}^p||_2}{||\boldsymbol{\gamma}^p||_2}. \tag{5.2}$$

In this notation, $\boldsymbol{\gamma}^p$ represents the true proposal and $\tilde{\boldsymbol{\gamma}}^p$ the corresponding approximation. The values of the log-determinants are assessed in a similar way: In each iteration the log-determinant is computed by the use of the full Cholesky factorization, $\log\det(\widetilde{\boldsymbol{Q}})$. This is compared to the approximations obtained by the Chebyshev estimator and the ICT preconditioners, $\widetilde{\log\det}(\widetilde{\boldsymbol{Q}})$, by the relative error

$$\delta^1(\log\det(\widetilde{\boldsymbol{Q}}), \widetilde{\log\det}(\widetilde{\boldsymbol{Q}})) = \frac{\log\det(\widetilde{\boldsymbol{Q}}) - \widetilde{\log\det}(\widetilde{\boldsymbol{Q}})}{\log\det(\widetilde{\boldsymbol{Q}})}. \tag{5.3}$$

The values of (5.2) and (5.3) for all 1,000 iterations are compared visually along all conditions by displaying the corresponding means and ranges.

### 5.1.4 Results

#### Approximate sampling

For the dense data set the relative error between the true and approximate samples remains below 2.5% in all cases, see Figure 5.2. Except for $\log(\kappa) = 0$, the largest errors are observed in situations where the preconditioner has been provided by the IC factorization without additional fill-in. For $\log(\kappa) = 0$ the ICT($1e^{-2}$) preconditioner shows the lowest fill-in ratio which results in the highest error for this choice of $\kappa$. For a given drop tolerance of the ICT preconditioner the error increases noticeably with increasing $\kappa$. By decreasing the

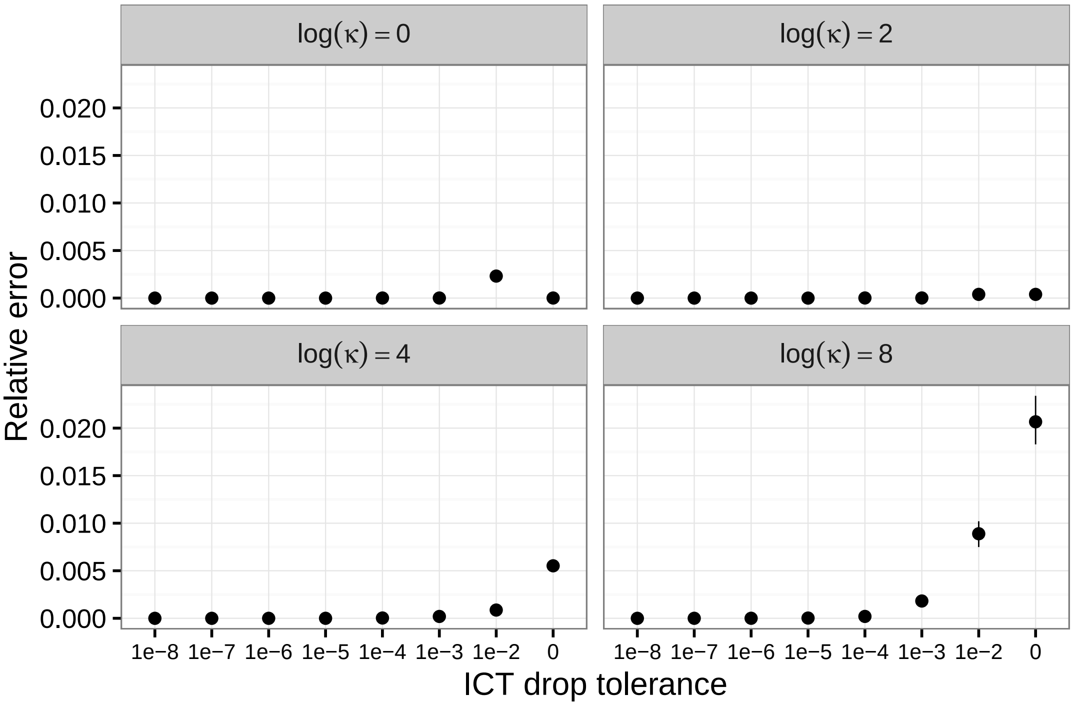


Figure 5.2 Mean and range of (5.2) for the dense data set.

drop tolerance, or, equivalently, by increasing the fill-in ratio, higher errors which result from higher $\kappa$ values can be addressed. However, this is only feasible up to a certain drop tolerance as the fill-in ratio of the preconditioners increases dramatically with $\kappa$. For example, for the $\mathrm{ICT}(1e^{-8})$ factorization a fill-in ratio of $R = 1.24$ is observed for $\log(\kappa) = 0$, whereas $\log(\kappa) = 8$ yields $R = 24.3$.

As expected, more effort is needed in order to obtain comparable error sizes for the sparse data situation, see Figure 5.3. Here, the relative error reaches up to 150%. Similar to the dense data situation the error increases with increasing $\kappa$. Decreasing the drop tolerance leads to an improvement. However, fill-in ratios tend to get high rather fast when decreasing the drop tolerance so that clearance for this parameter is limited. In contrast to the dense data situation the fill-in ratio remains nearly constant if $\kappa$ changes and the drop tolerance is fixed. This indicates that, in this situation, the small amount of information in the data has more influence on the condition of $\widetilde{\boldsymbol{Q}}$ than the precision of the GMRF prior.

Note that in all cases the iterative methods indicates successful convergence, that is, the relative error between the approximate samples of the last two iterations dropped below $1e^{-4}$. For the dense data situation and moderate values of $\kappa$ the mean number of iterations required for convergence lies below five for all three Krylov subspace methods. Only when

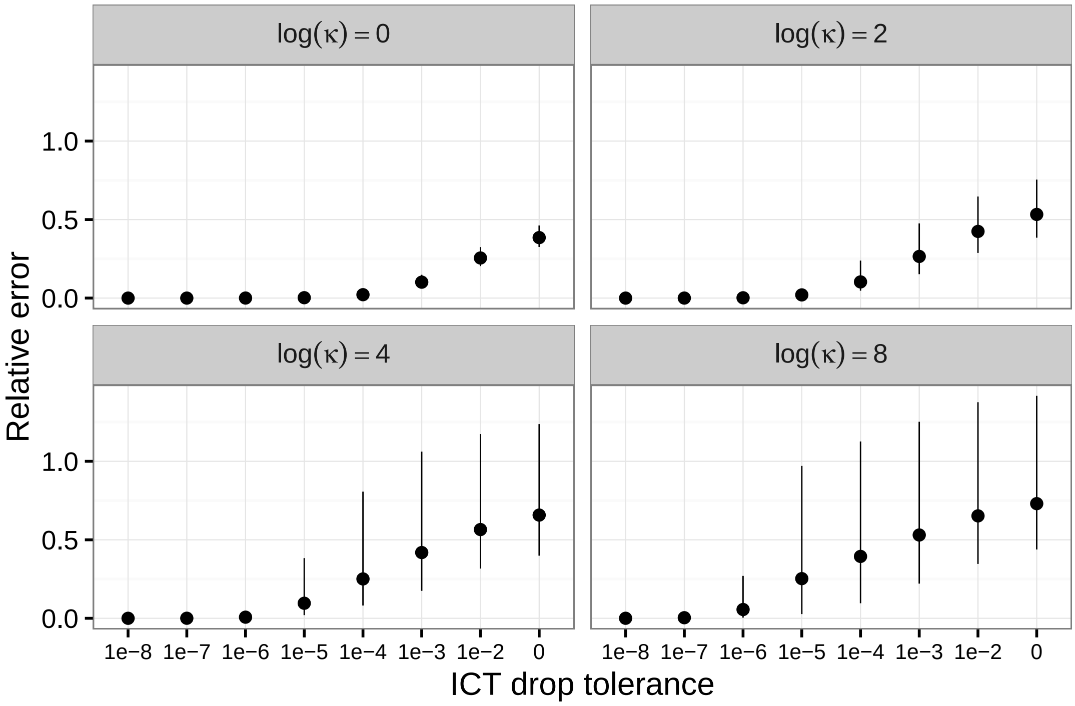


Figure 5.3 Mean and range of (5.2) for the sparse data set.

considering large precision values in combination with relatively high drop tolerances this number increases up to 30 iterations. With respect to the sparse data situation the number of iterations reflects well on the poor condition of the systems to solve. The values are considerably higher than for the dense data situation, especially for high drop tolerances: the number of iterations ranges from 10 to 120. In both data situations, the Lanczos algorithm for approximate sampling needs more iterations than the conjugate gradient method for computing the mean and applying linear constraints.

### Approximation of log-determinants

Compared to the dense data situation the relative errors of the Chebyshev estimator for log-determinants are increased by a factor of 10 for the sparse data set, see Figure 5.4 and 5.5. In addition, the influence of $\kappa$ seems to be reversed between both data situations: For increasing $\kappa$ the approximation worsens for the dense data while it seems to improve for the sparse data set. However, a closer examination reveals that for the latter data set the absolute error remains nearly constant when $\kappa$ varies.

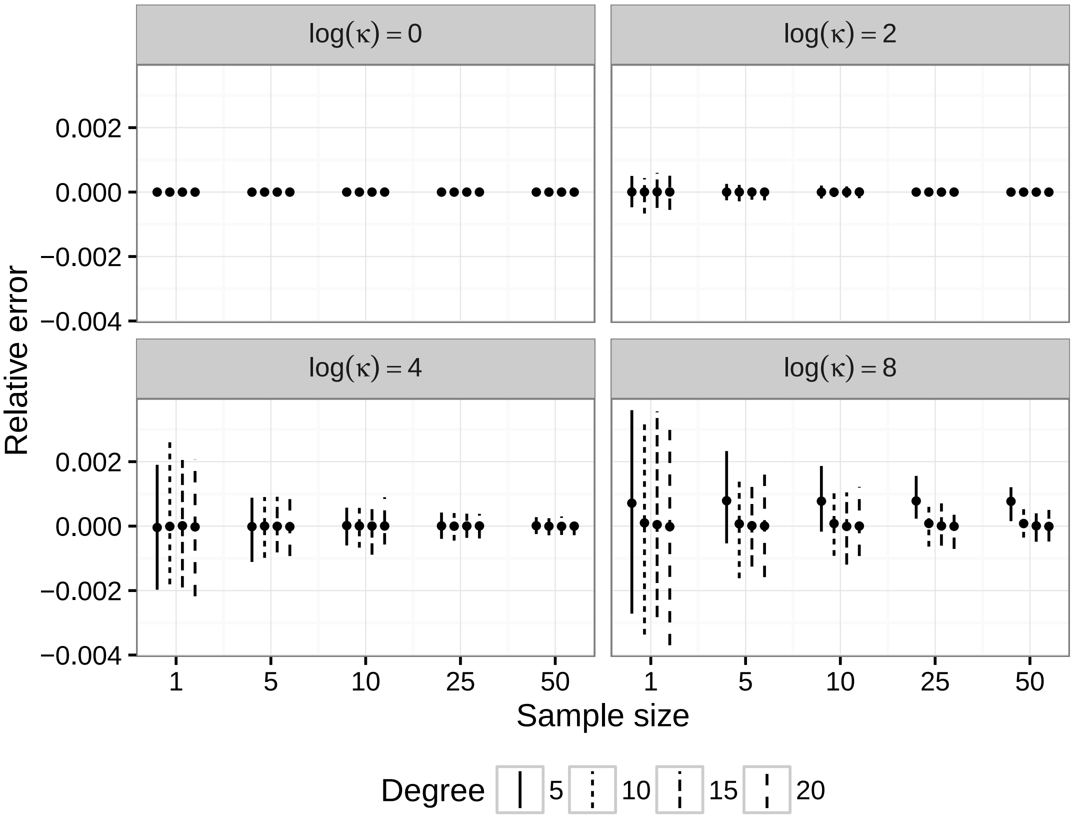


Figure 5.4 Mean and range of (5.3) for the stochastic Chebyshev estimator applied to the dense data set.

This observation is in accordance with what has been observed for approximate sampling: The lack of information in the data seems to outweigh the effect of $\kappa$. The results of the Chebyshev estimator further depend on the number of samples for the stochastic trace estimator and the degree of Chebyshev polynomials. The influence of the former is as expected: With increasing sample size the estimator becomes more precise. It is interesting to note that even a sample size of one leads to relatively small errors. The influence of the degree of the Chebyshev polynomials, on the other hand, is limited: The precision remains nearly constant when increasing this parameter. However, a positive bias for the choice of five degrees can be observed for both data situations.

Compared to the Chebyshev estimator the errors obtained from the approximation using ICT preconditioners are about twice as large in magnitude, see Figure 5.6 and 5.7. Apart from that the general behavior is similar: For the sparse data set the errors are increased by a factor of 10 and the influence of $\kappa$ is again reversed between both data sets. In addition,

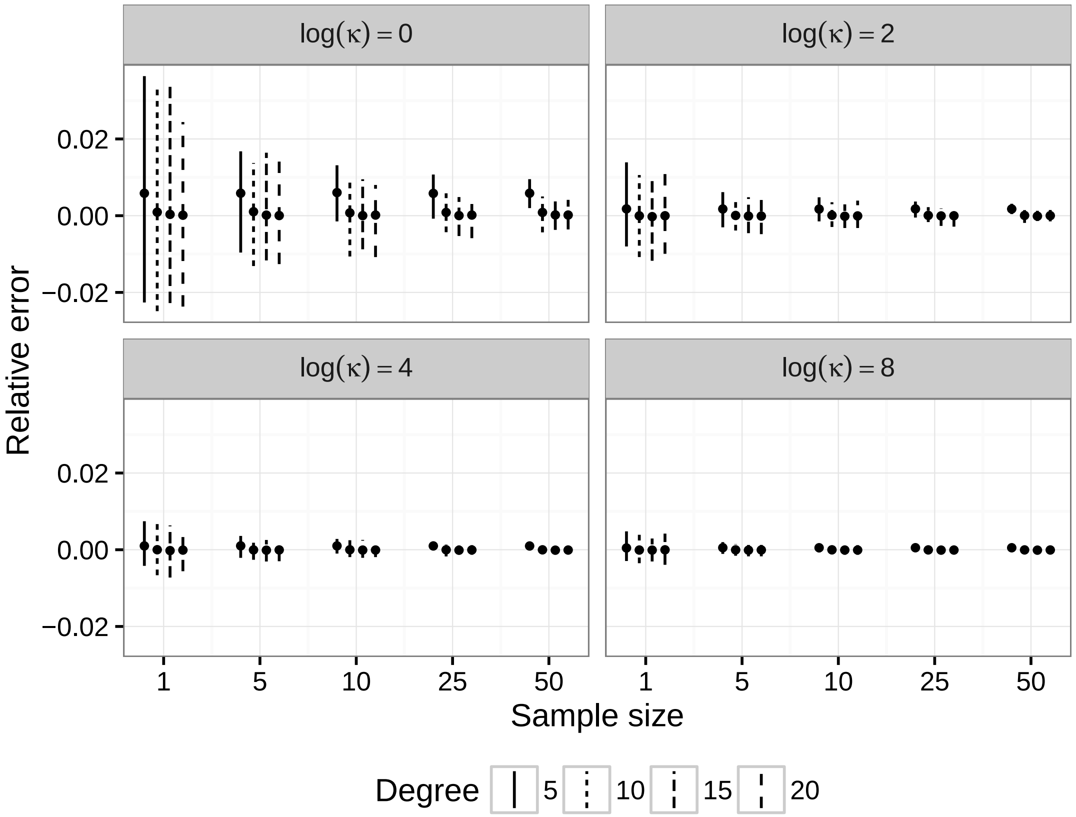


Figure 5.5 Mean and range of (5.3) for the stochastic Chebyshev estimator applied to the sparse data set.

decreasing the drop tolerance, i.e. increasing the fill-in ratio of preconditioners, reduces the approximation error. Furthermore, it is interesting to note that this approximation method strictly underestimates the log-determinant, i.e. the approximation is always smaller than the log-determinant obtained from the full Cholesky factor. This can be seen as an advantage over the Chebyshev approximation because the true difference between the log-determinants in the computation of the acceptance probability is most likely kept in the presence of a systematic bias. Random variation as induced by the Chebyshev estimator, on the other hand, represent an additional source of uncertainty which needs to be accounted by the overall MCMC error.

### 5.1.5 Conclusion

The above simulation results can be summarized as follows: The amount of information in the data seems to have the strongest influence on the condition of $\widetilde{\boldsymbol{Q}}$ and, thus, on

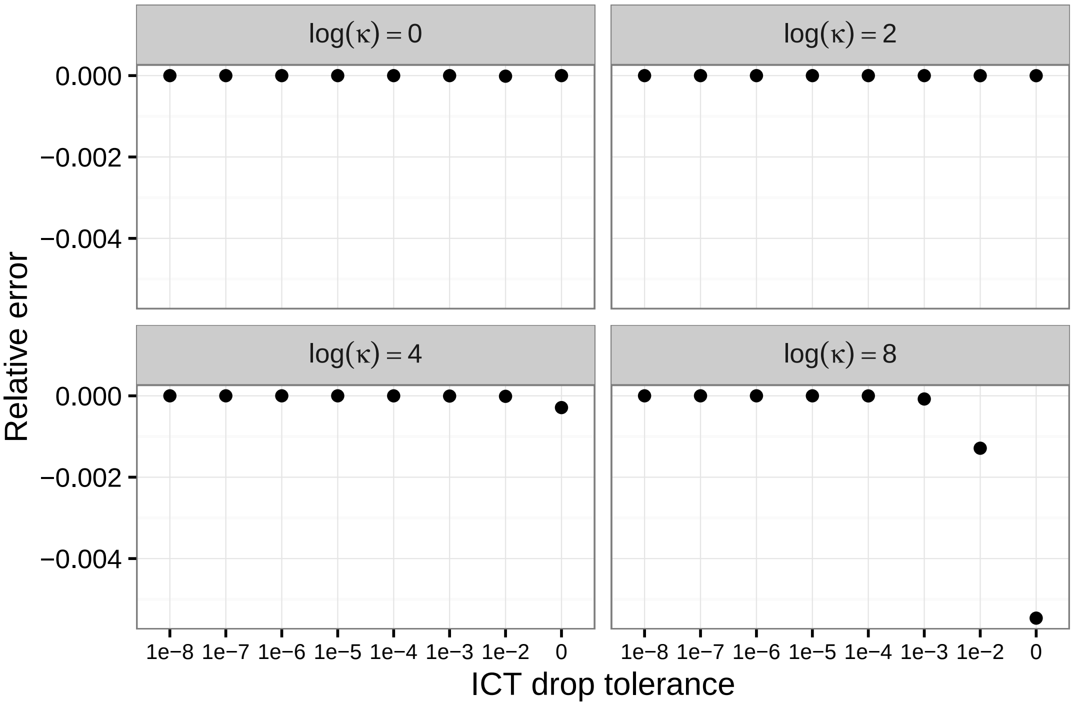


Figure 5.6 Mean and range of (5.3) for the ICT log-determinant approximation applied to the dense data set.

the performance of the Krylov subspace methods for approximate sampling as well as on the approximation of the log-determinant. The magnitude of $\kappa$ also appears to have an effect, especially for the dense data setup. In addition, decreasing the drop tolerance of the incomplete Cholesky factorization leeds to more precise approximations at the cost of an increased fill-in ratio.

## 5.2 Assessing the impact on the final results

In consideration of the above results the goal of this section is to analyze to what extent the MCMC algorithm is able to account for the approximation errors, i.e. if it is possible to obtain “exact” results although critical parts of the algorithm rely on approximations. To this end, the simulation setups are expanded and results are compared to a gold standard.

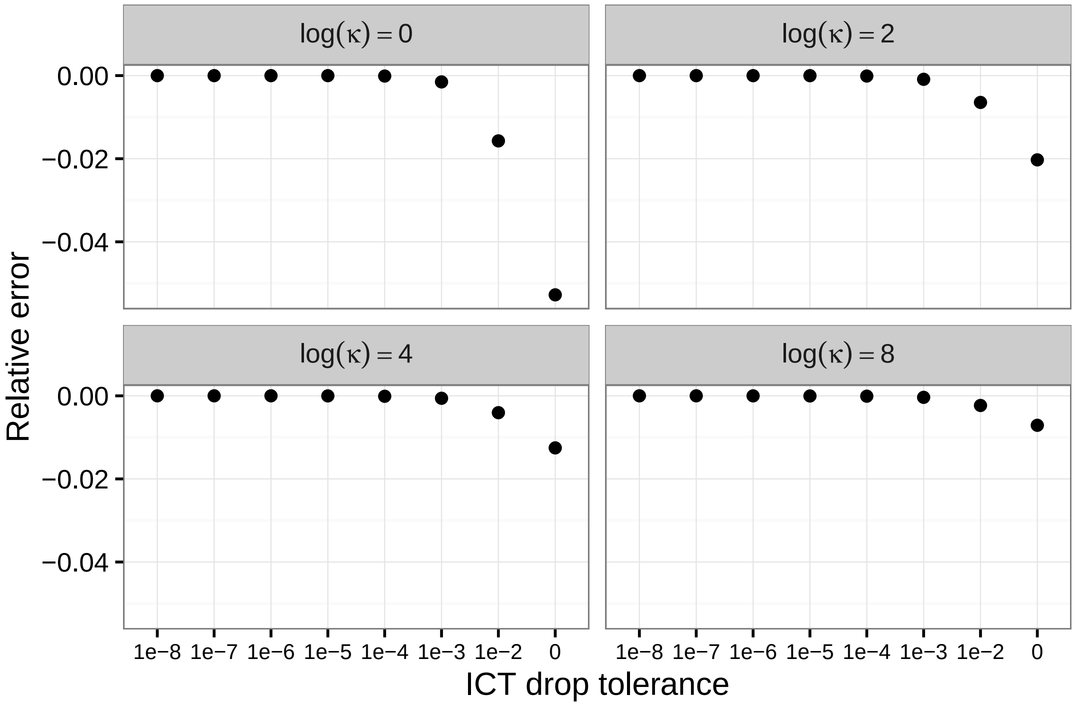


Figure 5.7 Mean and range of (5.3) for the ICT log-determinant approximation applied to the sparse data set.

## 5.2.1 Simulation setup

### Dense data situation

The setup of the pixel-wise data in Section 5.1.1 is expanded by two additional smooth effects. Thus, the linear predictor for the $j$th pixel becomes

$$\boldsymbol{\eta}_j = f_1(j)\boldsymbol{z}_1 + f_2(j)\boldsymbol{z}_2 + f_3(j)\boldsymbol{z}_3 + \boldsymbol{X}\boldsymbol{\beta}.$$

The components $\boldsymbol{z}_1$, $f_1$, $\boldsymbol{X}$, and $\boldsymbol{\beta}$ are constructed in the same way as in Section 5.1.1. The new smooth functions are created by the following formulas:

$$f_2(j) = j_x - \tfrac{n_x}{2} + n_y \sin(\tfrac{j_y}{n_y})$$
$$f_3(j) = \sqrt{(j_x - \tfrac{n_x}{2})^2 + (j_y - \tfrac{n_y}{2})^2}.$$

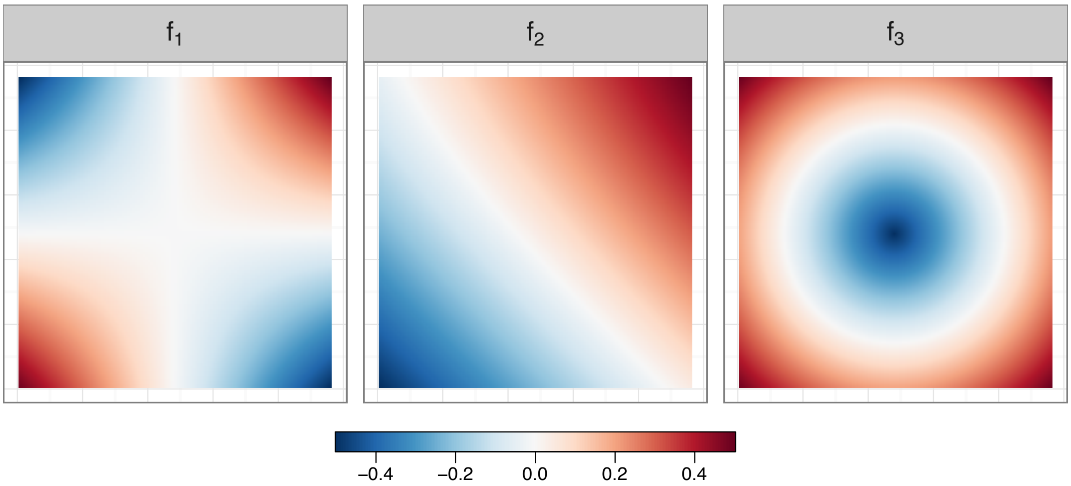


Figure 5.8 Smooth effects used for the simulation.

Again, the values of these effects are linearly scaled to the interval $[-0.5, 0.5]$. Figure 5.8 displays the resulting true effect images. The metric covariates $\boldsymbol{z}_2$ and $\boldsymbol{z}_3$ are randomly assigned over the interval $[-1, 1]$.

### Sparse data situation

The sparse data set is exactly the same as in Section 5.1.1.

## 5.2.2 Modeling

Due to the additional smooth effects the pixel-wise regression model for the dense data set is expanded by

$$\boldsymbol{\eta}_j = \gamma_{j1}\boldsymbol{z}_1 + \gamma_{j2}\boldsymbol{z}_2 + \gamma_{j3}\boldsymbol{z}_3 + \boldsymbol{X}\boldsymbol{\beta}.$$

For all random effects $\boldsymbol{\gamma}_k = (\gamma_{1k}, \ldots, \gamma_{mk})'$, $k = \{1, 2, 3\}$, spatial GMRF priors with Kronecker-sum penalties according to (5.1) are imposed. Priors for fixed effects are unchanged. The model for the sparse data set is unchanged.

In the first section of this chapter it has been shown that the error of the approximate samples expanded with increasing $\kappa$. Since the precision parameter is a random parameter it may seem that no control over this issue can be gained. However, by adjusting the prior distribution of $\kappa$ it may be possible to control the domain of the resulting posterior, especially in situations where the likelihood does not dominate the prior information. Using independent Gamma priors for $\kappa$ this can be achieved by choosing appropriate values for the hyperparameters $a$ and $b$. Therefore, for the sparse data situation two different combinations for the shape and rate parameters of the Gamma prior are chosen: First, the combination of $a = 10$ and $b = 0.001$ is used which yields relative high values of $\kappa$. The second choice, $a = 5$ and $b = 1$, results in a prior for which more probability mass is concentrated around smaller values of $\kappa$. For the dense data situation only one choice is made since the influence of these parameters on the final result is rather limited due to the large amount of information in each pixel. Here, the values $a = 1$ and $b = 1e^{-5}$ are chosen for all precision parameters and the additional dispersion parameter $\kappa_y$.

For approximate sampling IC, ICT($1e^{-4}$), and ICT($1e^{-8}$) factorizations are used as preconditioners. For these choices, three different strategies with respect to the calculation of log-determinants are chosen: The modified sampling scheme as explained in Section 4.2.4 is applied in order to avoid the computation of log-determinants completely. In addition, block updating of $(\boldsymbol{\gamma}, \kappa)$ is performed by using the Chebyshev estimator using $p = 10$ degrees and 500 samples. The latter number has been chosen as a compromise between precision on the one hand and computation time on the other hand. Furthermore, block updating is performed by calculating the log-determinant from an approximate Cholesky decomposition obtained by the ICT factorization. For the block updating strategies the tuning parameter $f$ (see Section 3.1.4) is chosen so that the acceptance rates lie between 0.2 and 0.4.

First and second moments of marginal posteriors for all high-dimensional regression coefficients are computed on-line as explained in Section 4.2.5. For precision parameters and fixed effects all samples are saved. For each model of the dense data set four independent chains of length 16,000 are generated. For the sparse data set, run length is increased to 26,000. On-line calculation of posterior moments for the smooth effects starts after an initial burn-in period of 1,000 samples.

### 5.2.3 Validation

The results of all fitted models are compared to the results obtained by the INLA approach as implemented in the R (R Core Team, 2016) package INLA (Rue et al., 2009). The INLA method is used for the following reasons: First, it gives sufficient precise estimates that are comparable with those from long MCMC runs. Second, convergence problems are avoided that may arise using MCMC. Due to computational restrictions (working station

B, see Section A.1) it was not able to obtain INLA's results for the full Laplace strategy. Instead, the simplified Laplace strategy has been chosen In contrast, all other calculations have been performed on working station A.

The symmetric Kullback–Leibler distance (SKLD) is used to measure the discrepancy between the marginal posteriors obtained by the INLA approach and the corresponding Gaussian approximations resulting from the MCMC algorithm. The SKLD is computed by $(D_{KL}(P_1||P_2) + D_{KL}(P_2||P_1))/2$, where $P_1$ and $P_2$ are two posterior marginals and $D_{KL}(P_1||P_2)$ is the Kullback–Leibler divergence given by

$$D_{KL} = \int\limits_{-\infty}^{\infty} p_1(x) \log \frac{p_1(x)}{p_2(x)} \mathrm{d}x.$$

In order to reveal the locations with the highest discrepancies standardized coefficient maps for all smooth effects are calculated and compared. To be more precise, posterior means are divided by their corresponding standard deviations for both, MCMC and INLA. The differences of these maps are than analyzed graphically.

### 5.2.4 Results

#### Dense data situation

According to the estimated potential scale reduction factors it can be assumed that all chains of the nine models converged to their stationary distributions (potential scale reduction factor $< 1.1$ for all parameters). However, comparing these values along the different approximation strategies indicates better mixing and convergence behavior for the modified sampling scheme. This is confirmed by the sampling paths of $\log(\kappa_1)$ displayed in Figure 5.9. Here, the first 15,000 iterations are displayed along with the upper and lower limits of the corresponding 95% credible interval obtained by INLA (red dotted lines). As can be seen, mixing is best for the modified sampling scheme, although the other strategies still yield acceptable results. The largest differences between the approximation strategies can be observed with respect to the agreement with the marginal posterior estimated by the INLA approach. For example, the 95% credible interval obtained by the Chebyshev strategy using 500 samples and Chebyshev polynomials of degree 10 is about 30 to 40% wider than INLA's result. In contrast, the ICT approximation of the log-determinant yields intervals that are about 4 to 12% wider, and for the modified sampling scheme this

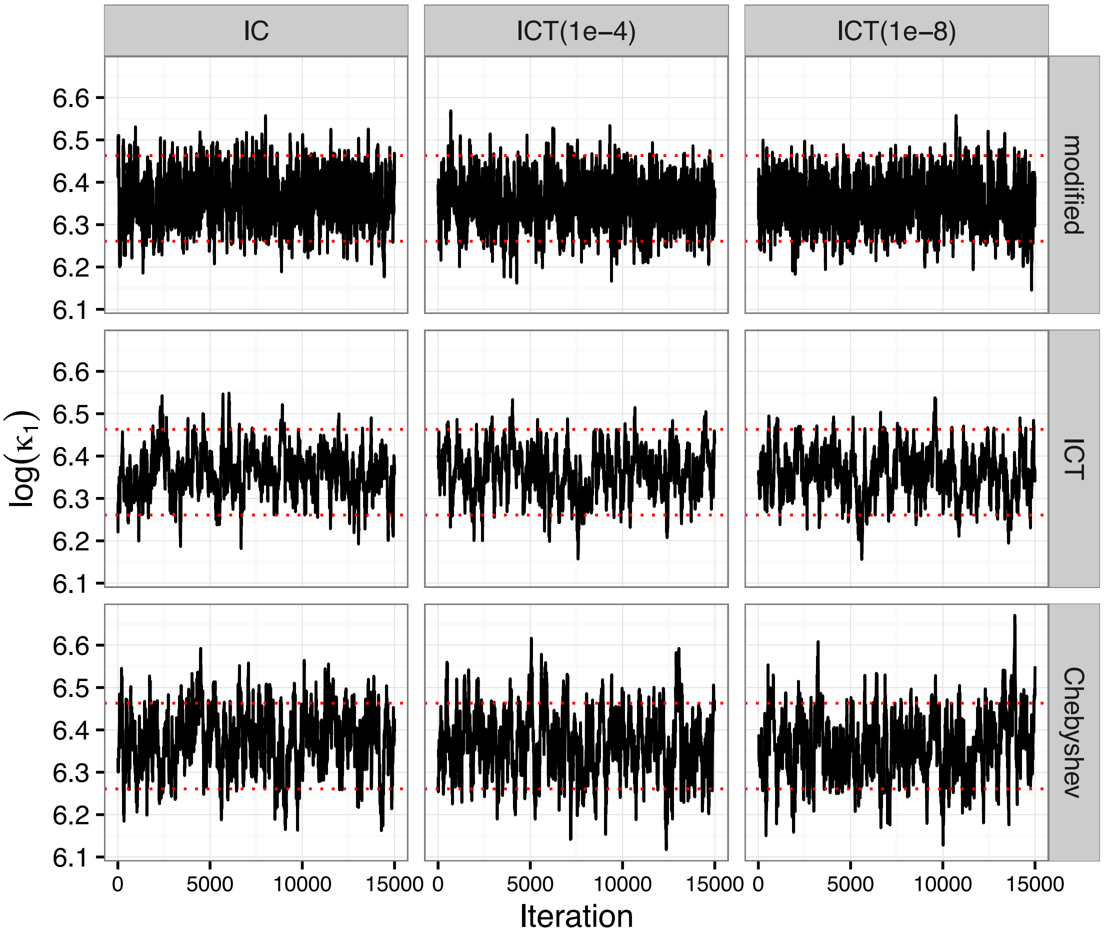


Figure 5.9 Sampling paths for $\log(\kappa_1)$.

discrepancy is less than 1%. This indicates that the MCMC algorithm has difficulties in accounting for the error that is induced by the stochastic trace estimator (4.10) used within the Chebyshev approximation. Of course, this error can be reduced if the sample size for the Hutchinson estimator is increased. However, this would increase computational requirements dramatically which greatly limits its practical use, especially when simpler strategies achieve better results. Also of interest is the fact that a decrease of the drop tolerance, i.e. an increase of the fill-in ratio of the ICT factorization does not yield any noticeable difference in the quality of the sampled paths. This is a surprising result which is in contrast to the results obtained in the first section of this chapter where decreasing the drop tolerance resulted in a noticeable improvement of both the sampling error and the error of the approximations of the log-determinant. Similar results are obtained for the precision parameters of the other smooth effects.

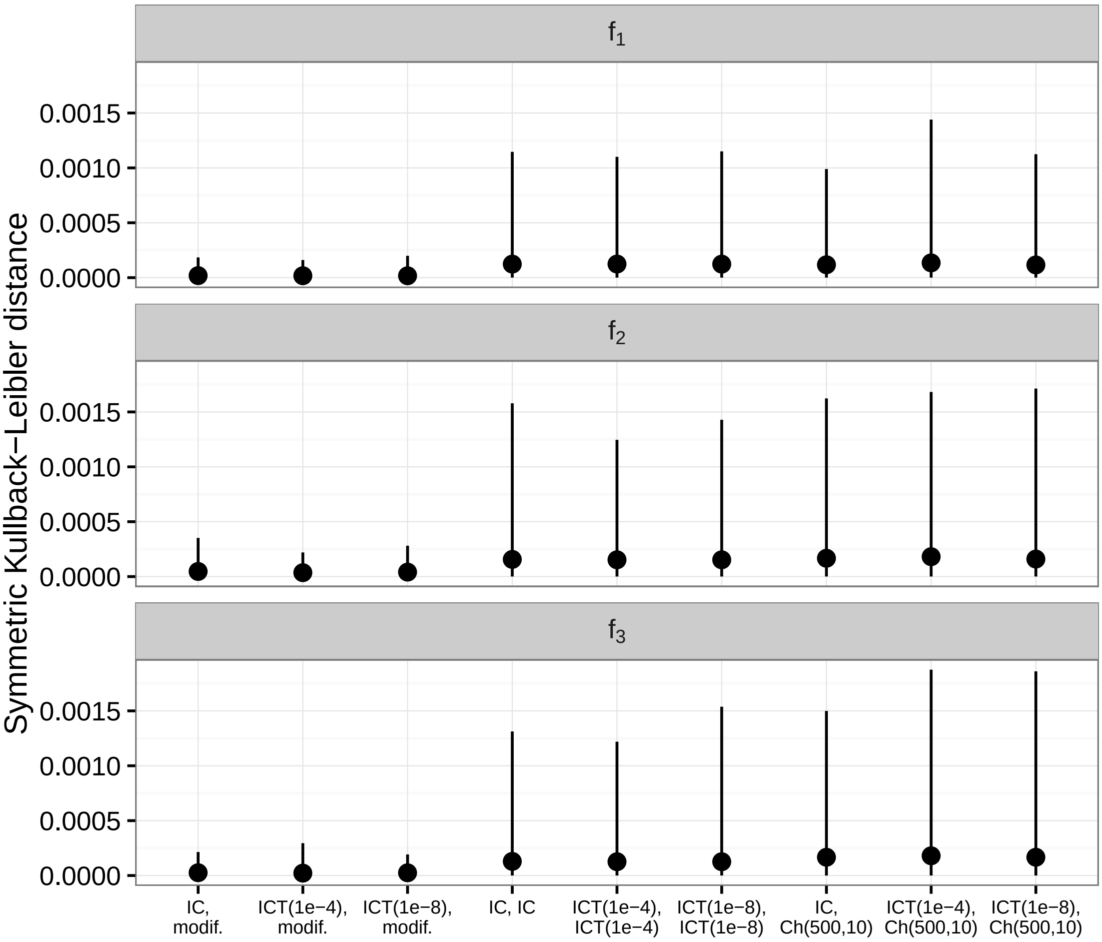


Figure 5.10 SKLD for all three smooth effects.

The values of the SKLD for all three smooth effects are displayed in Figure 5.10. Here, mean values as well as maximum and minimum values are plotted for the different approximation strategies. Overall, agreement between the MCMC results and INLA's marginal posteriors is very good, the maximum SKLD is 0.00188. However, the modified sampling scheme can clearly be seen as the best strategy (maximum SKLD: 0.00035) which is in accordance with the above results. Note that the missing effect of the drop tolerance of the ICT factorization can also be confirmed: An improvement of the agreement between the MCMC and INLA results cannot be observed when increasing the fill-in ratio of the preconditioners.

Figure 5.11 shows the differences of the standardized coefficient maps for all three smooth effects along the approximation strategies. Overall, the modified sampling strategy seems to perform best, followed by the ICT block updating strategy. The largest discrepancies

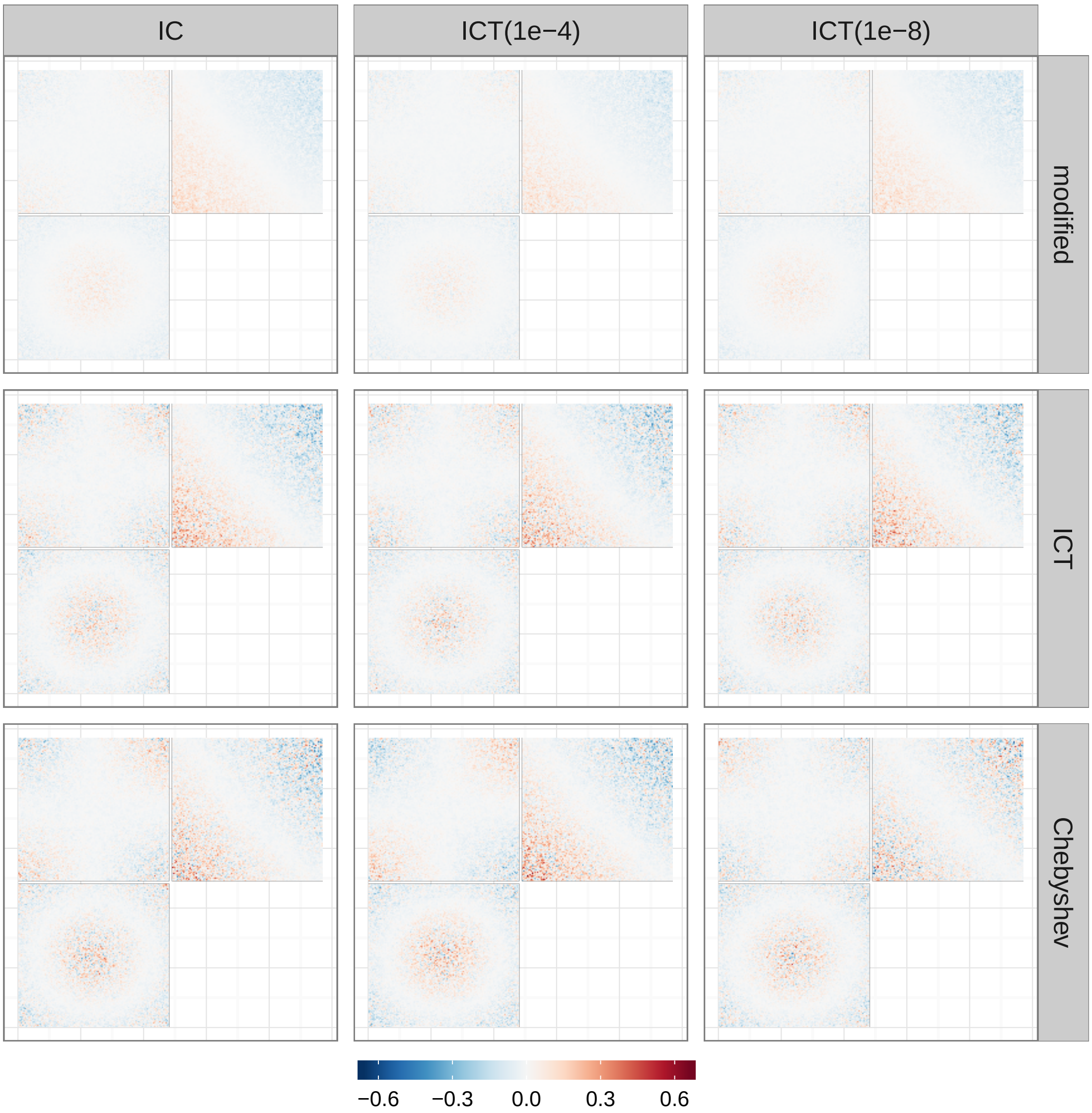


Figure 5.11 Differences of the standardized coefficient maps for all three smooth effects.

can be observed in areas where the true effect maps take their highest values in magnitude. In addition, the effect of decreasing the drop tolerance of the ICT factorization on the differences of the standardized coefficient maps seems limited.

### Sparse data situation

First, the results for the case $a = 5$ and $b = 1$ are considered. Here, all generated MCMC chains converged to their stationary distribution except for the chains regarding $\kappa$. The potential scale reduction factor for this parameter varies between 1.11 and 1.37, whereas

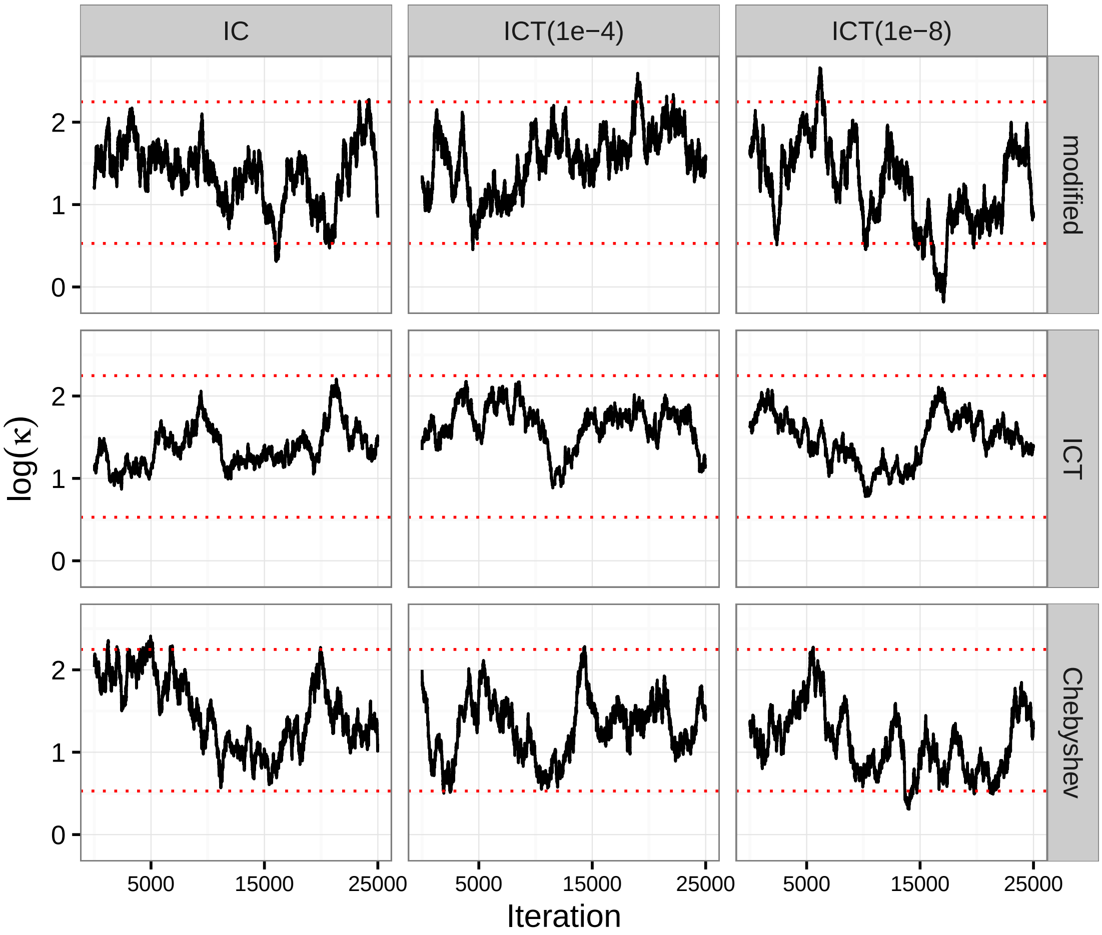


Figure 5.12 Sampling paths for $\log(\kappa)$ for $a = 5$ and $b = 1$.

smaller values have been obtained for the modified sampling scheme and larger values for the block updating strategy using the stochastic Chebyshev approximation. Sampling paths of these chains are displayed in Figure 5.12. Again, the 95% credible intervals for this parameter obtained by the INLA approach are indicated by the red dotted lines. As can be seen the posterior distribution is explored rather slowly: Dependency between successive samples is quite high which results in insufficient mixing. For the block updating strategy using the ICT approximation it can also be seen that not the entire part of the domain of the marginal posterior is reached. Instead, only the inner part is explored, i.e. the variance of the marginal posterior is underestimated. For the other two strategies it seems that sufficient exploration of the marginal posterior could be achieved by increasing the sampling size of the MCMC algorithm.

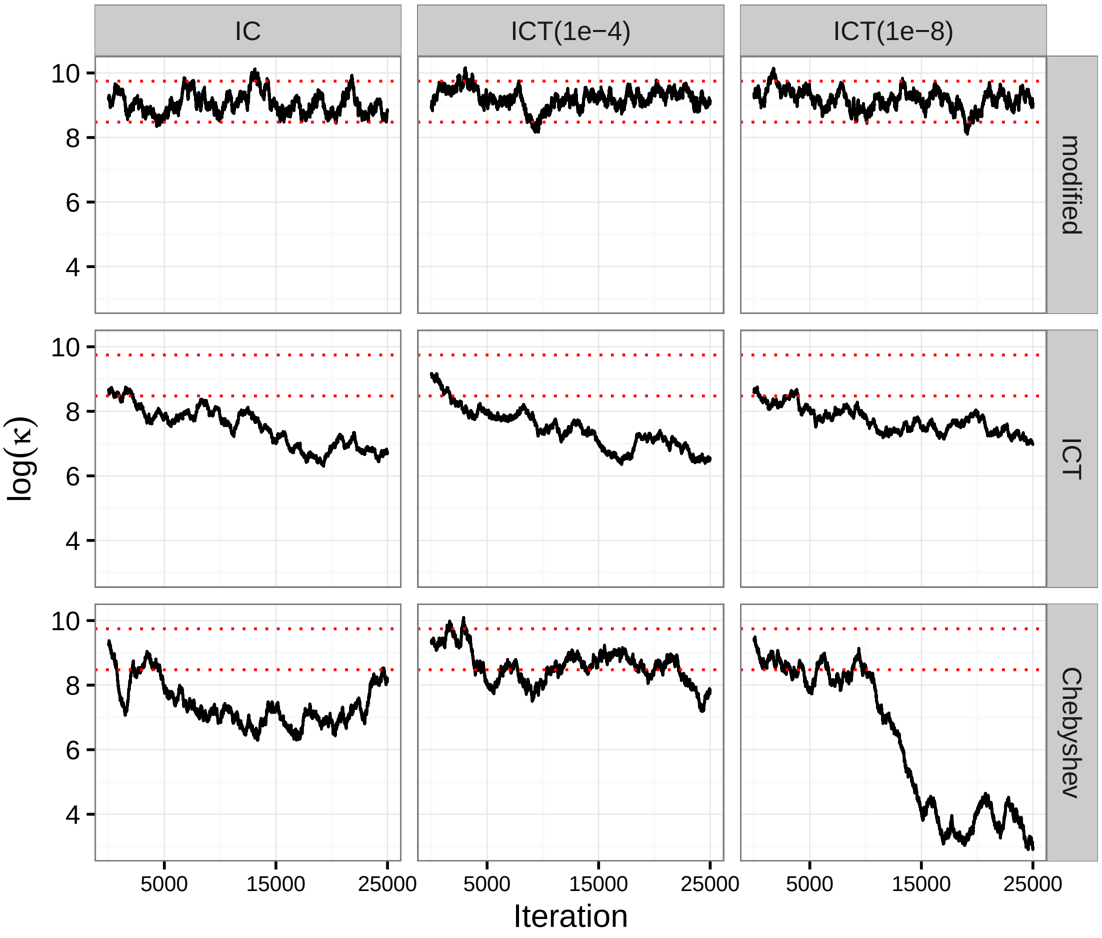


Figure 5.13 Sampling paths for $\log(\kappa)$ for $a = 10$ and $b = 0.001$.

Convergence diagnostics for the second case, i.e. $a = 10$ and $b = 0.001$, are different: For the modified sampling scheme all parameters converged to their stationary distribution (all potential scale reduction factors $< 1.1$). In contrast, the chains of $\kappa$ for the other two strategies did not converge. Here, the potential scale reduction factor for $\kappa$ varies between 1.81 and 2.43. In fact, inspecting the sampling paths for this parameter visually (Figure 5.13) reveals that joint updating $(\gamma, \kappa)$ using the ICT and Chebyshev strategies causes the algorithm to diverge. Interestingly though, in the subspace of the remaining parameters the Markov chain converged independently of $\kappa$. This demonstrates that the effect of the small amount of information in the data on the performance of the MCMC algorithm is more pronounced than the actual value of $\kappa$ – a results which is in accordance with the findings in Section 5.1. In contrast, the modified sampling scheme shows an acceptable behavior, although exploration of the stationary distribution is rather slow. Note that, just as in the

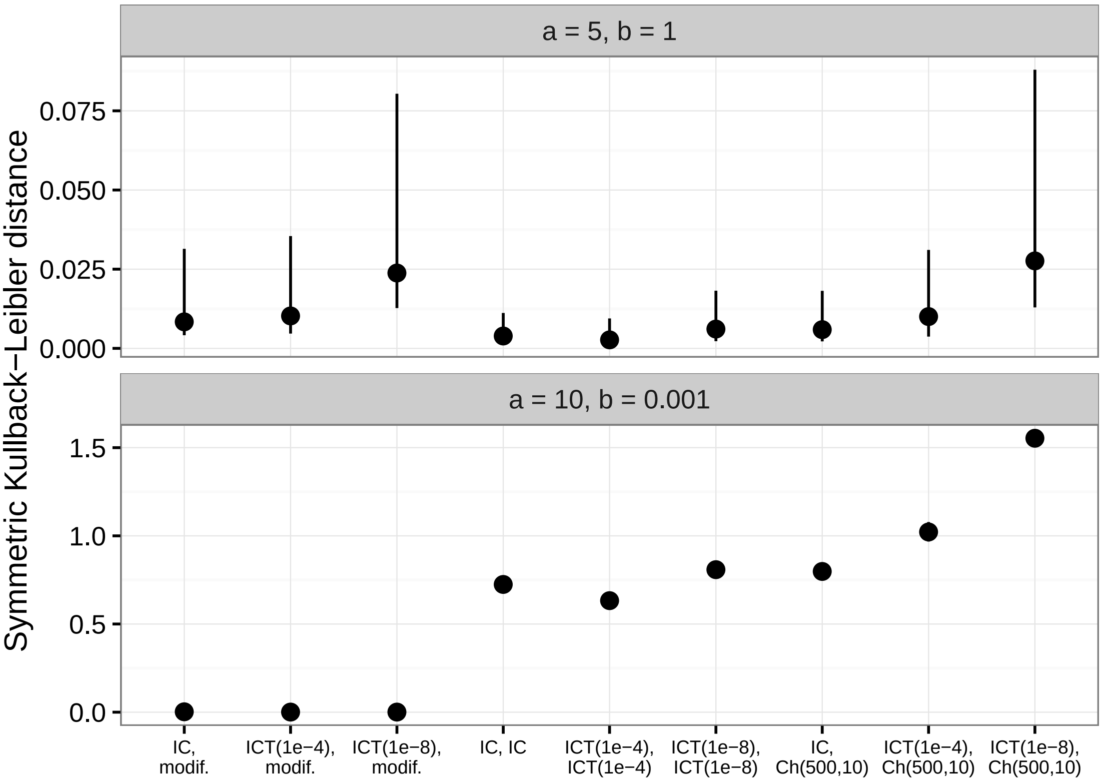


Figure 5.14 SKLD for the smooth effect.

other cases above, an increase of the fill-in ratio of the ICT preconditioner does not seem to yield different results.

Figure 5.14 displays the values of the SKLD for both cases. For $a = 5$ and $b = 1$ the SKLDs are in an acceptable range. Block updating using the ICT factorization leads to the smallest errors, followed by the Chebyshev approximation strategy and the modified sampling scheme. Noticeable is the fact that decreasing the drop tolerance of the ICT factorization leads to an increase of the SKLDs. This is counterintuitive to the results obtained in Section 5.1. For the second scenario ($a = 10$, $b = 0.001$) much higher values for the SKLD can be observed. Here, the modified sampling scheme is as good as in the first case (maximum SKLD: 0.08). However, for the other approximation strategies unacceptable high values for the SKLD are observed. This is not surprising given the fact that the corresponding $\kappa$-chains did diverge.

Figure 5.15 reveals that, for $a = 5$ and $b = 1$, the modified sampling scheme performs best with respect to the magnitude of the differences of standardized coefficient maps. In particular, the simplest strategy (IC and modified sampling scheme) yields the least

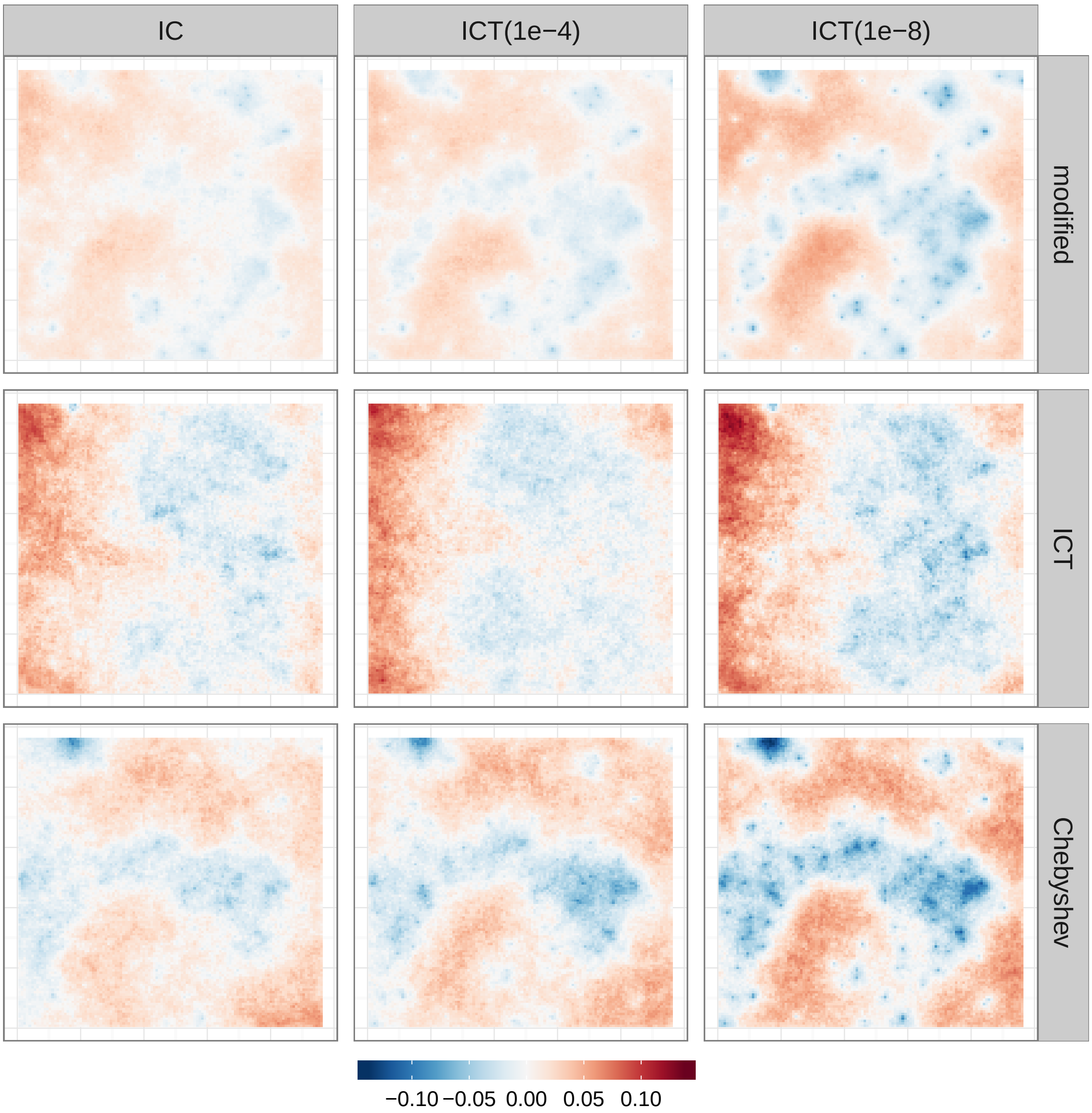


Figure 5.15 Differences of the standardized coefficient maps for all three smooth effects for $a = 5$ and $b = 1$.

pronounced error map. In contrast, standardized coefficients for both blocking strategies show serious deviations from the standardized coefficients obtained by the INLA approach. In addition, it can also be seen that the error increases with decreasing drop tolerance of the ICT factorization.

The results for $a = 10$ and $b = 0.001$ are similar (see Figure 5.16): Here, the best match with INLA's standardized regression coefficients is obtained by the modified sampling scheme. The worst result is observed for the Chebyshev strategy and the ICT$(1e^{-8})$

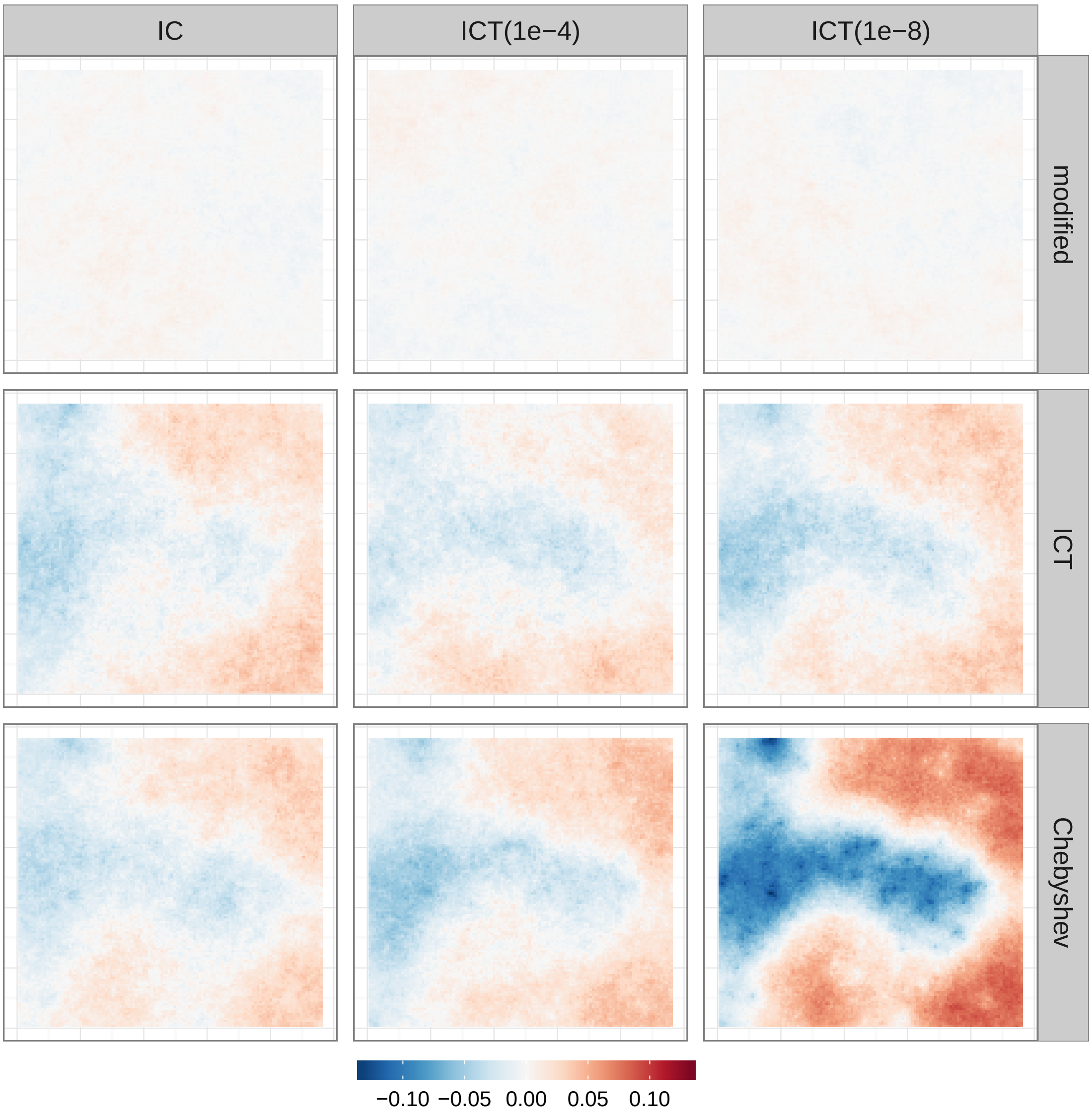


Figure 5.16 Differences of the standardized coefficient maps for all three smooth effects for $a = 10$ and $b = 0.001$.

preconditioner. Again, increasing the fill-in ratio of the preconditioners leads to larger errors.

## 5.3 Chapter summary

The simulations conducted in this chapter allow the following conclusions: First, the impact of the approximation strategies on the final results depends on the data situation. If the information content in the data is dense, good to excellent performance can be achieved. However, if the data set contains only little information, i.e. is sparse, it becomes more difficult for the MCMC sampler to account for the approximation errors.

Second, with respect to the different approximation strategies the following statements can be made: The MCMC error may not be able to account for the inaccuracy which is induced by the stochastic Chebyshev approximation of the log-determinant using 500 samples. For the sparse data set it has been observed that the MCMC algorithm diverged and even for the dense data set the marginal posteriors were significantly wider. Increasing the sample size of the Hutchington estimator may provide a remedy, however, this would lead to a dramatic increase of computation time. Results for the approximation strategy based on the ICT factorization are similar. For the dense data set, however, this method yields acceptable results. By far the best results were obtained from the modified sampling scheme. This strategy yielded stable Markov chains as well as a good agreement with INLA's results in both data situations, although longer MCMC runs are required for the sparse data set. Another point in favor for this relatively simple strategy is less computation time compared to the block update strategies. As a more surprising result an increase of the fill-in ratio of the incomplete Cholesky preconditioner does not result in better performance.

Third, it could be observed that assessing the performance of approximations and the quality of final results is crucial for the application to real world data sets. However, this could be a difficult task as there is no ground truth available in this case. Therefore, visual inspection of sample paths and calculation of convergence diagnostics must be seen as absolutely necessary before results can be interpreted. The number of iterations that are required by the Krylov subspace methods to converge may be of additional help in order to indicate potential difficulties due to ill-conditioned linear systems.

Finally and most importantly, it has been shown that, under certain conditions, approximation strategies exist that are able to provide excellent results to large-scale problems. In the following chapter these strategies are applied to three real world applications.

# 6 Applications

The results obtained in the simulation studies of Chapter 5 can be used to formulate a MCMC algorithm which is able to fit regression models with high-dimensional components as they appear in real-world problems. In particular, the modified sampling scheme is selected and for approximate sampling preconditioning using the IC factorization without additional fill-in is used. In this chapter, this algorithm is used to analyze three different applications in the field of bio-medical imaging. First, a supervised algorithm for the segmentation of MS lesions is trained. Second, a voxel-wise regression model is fitted to segmented GM images in order to reveal differences in age-related atrophy of GM in MS patients. Third, object-based co-localization is performed by fitting a spatial point process to the data of fluorescence microscopy images.

The first application in Schmidt et al. (2017) uses data from the same cohort as in Section 6.1. Thus, the data description given in Section 6.1.1 is partly identical to the description given in Schmidt et al. Furthermore, the second application of this chapter and Section 5.2 in Schmidt et al. (2017) both perform voxel-based morphometry. Thus, the introduction given in Section 6.2.1 is in large parts identical to the corresponding application in Schmidt et al.

## 6.1 Supervised segmentation of MS lesions

In this chapter a procedure for the development of a supervised MS lesion segmentation algorithm is presented. An introduction to the subject at hand will be given as well as a description of the data set. Subsequently, details about the modeling process as well as results are presented.

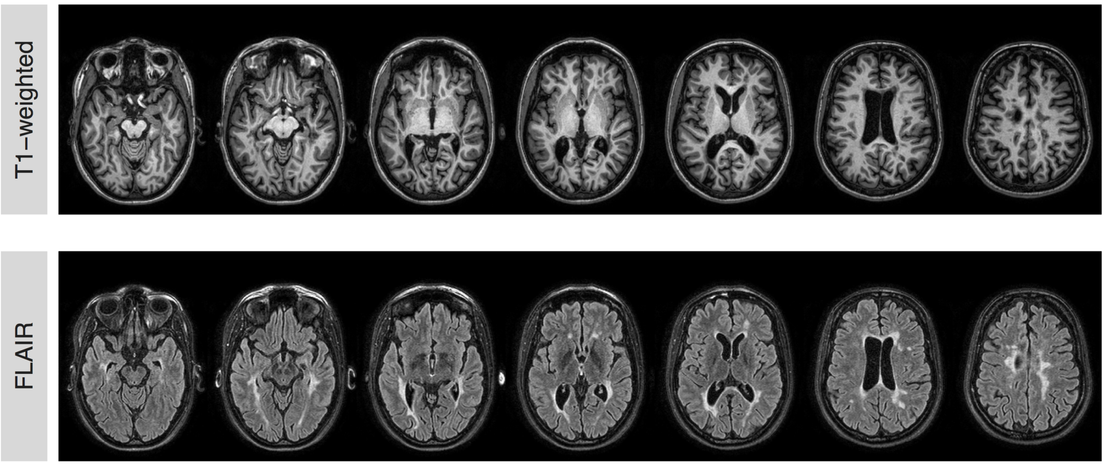


Figure 6.1 Selected axial slices of a T1-weighted and a FLAIR image for a randomly chosen MS patient.

### 6.1.1 Introduction

MS is a chronic inflammatory disease of the central nervous system. Although it is the main reason for early disabilities among young adults (Koch-Henriksen and Sørensen, 2010), its etiology is still unknown. According to the WHO's 2008 MS Atlas (World Health Organization, 2008) the average age of onset is 29.2 years (interquartile range, 25.3 – 31.8). Furthermore, females are about twice as much at risk as men, and strong regional differences can be observed. The clinical picture of MS is rather heterogeneous. While most patients experience their disease in relapses (relapsing-remitting MS), a minority suffer from continuous deterioration of symptoms (primary progressive MS). However, all types of MS have in common that inflammational plaques or lesions may appear in any location of the central nervous system. As a consequence, clinical symptoms can include a large variety of physical and mental problems that can also be associated with other diseases (Compston and Coles, 2008). Thus, a diagnosis only on clinical symptoms is difficult (Crayton et al., 2004). Instead, it is recommended to include radiological findings in order to increase diagnostics sensitivity and specificity (McDonald et al., 2001; Polman et al., 2011). This includes the identification and location of lesions in three-dimensional T2-weighted MR sequences (see the bright regions in the FLAIR image in Figure 6.1). Therefore, accurate segmentation of lesions is an elementary component of clinical research with respect to MS.

### Lesion segmentation algorithms

Manual segmentation of MS lesions in high-dimensional MR images is a difficult task. It strongly depends on the experience of the operator and is, in general, not a reliable procedure for lesion segmentation. Therefore, a variety of tools for the automatic segmentation exists, which includes fully and semi-automatic as well as supervised and unsupervised methods (Mortazavi et al., 2012). The majority of unsupervised methods rely on finite mixture models in combination with some kind of expectation maximization algorithm (van Leemput et al., 2001; Aït-Ali et al., 2005; Khayati et al., 2008; Freifeld et al., 2009). These approaches are popular because they allow including spatial information by the use of Markov random fields. Among supervised methods $k$-nearest-neighbors classification is a popular choice (Anbeek et al., 2004; Wu et al., 2006). Furthermore, classification algorithms based on mixture models have been presented (Herskovits et al., 2008) and, more recently, classifiers based on decision forests have been applied successfully (Akselrod-Ballin et al., 2009; Geremia et al., 2011). However, one of the most obvious supervised procedures has not been presented yet: A classifier based on voxel-wise binary regression models. There is one main difficulty connected with this approach: A binary model needs to be estimated for each brain voxel. This also includes voxels for which no lesions have been observed due to the limited sample size of the training set. For these voxels, no valid inference based on binary regression models can be performed. A solution to this problem can be accomplished by considering the spatial information among voxels and fitting the regression models jointly, for example using the framework of STAR models with spatially structured regression coefficients as presented in Section 2.2.1. However, this leads to a new problem: Considering all voxels jointly would produce a model that will probably be too large in order to be accessible with limited computational resources. Here, the methods presented in the previous chapters are used to overcome this problem.

### Data

The data set consists of MR images of 53 MS patients that are part of a larger cohort (312 MS patients) which has been collected at the Department of Neurology, Technische Universität München, Munich, Germany. The selected images correspond to patients with high total lesion volume (TLV). To be more specific, only patients with TLV $>$ 10 ml are used within the training process in order to keep the number of total data points within an acceptable range. Note that this reduction in sample size does not affect the information

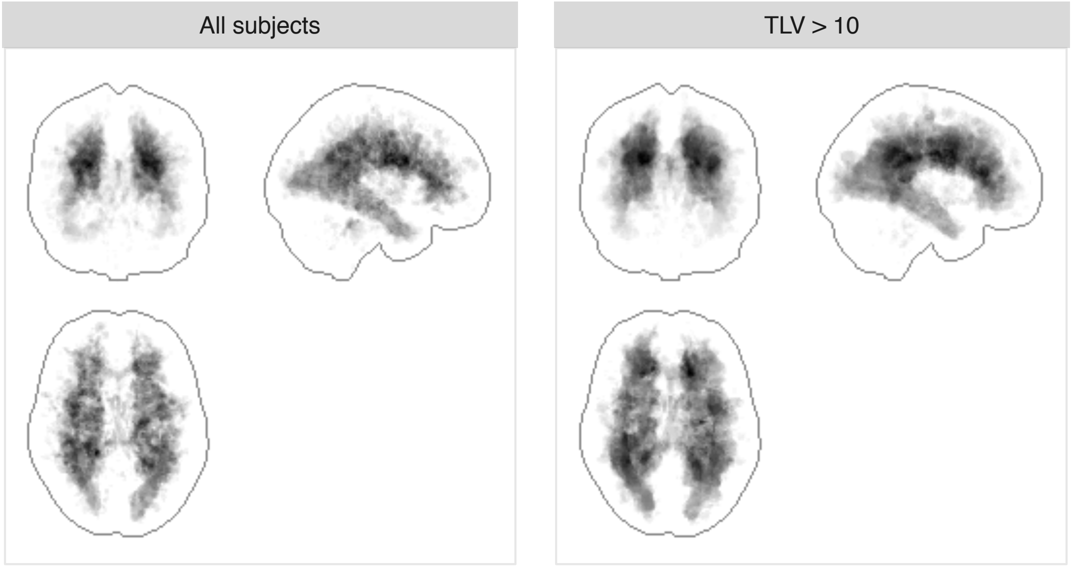


Figure 6.2 Maximum intensity projections for the position of MS lesions for all 312 subjects (left) and the selected 53 subjects (right).

of lesion distribution over the brain: Figure 6.2 shows that the position of lesion voxels of the full cohort (left panel) can be well approximated by the lesion structure of the 53 selected MS patients (right panel). This figure shows the maximum intensity projections (Wallis et al., 1989) of MS lesions in MNI space, that is, lesion voxels are averaged over saggital, cranial, and axial dimension of normalized binary lesion maps estimated by the lesion growth algorithm implemented in the LST toolbox (Schmidt et al., 2012) for the SPM[1] package.

The actual training data consists of three-dimensional gradient echo T1-weighted and T2-weighted fluid-attenuated inverse recovery (FLAIR) images. Both types of images were acquired on the same 3 Tesla scanner (Achieva, Philips, Netherlands). For the T1-weighted image, 170 contiguous sagittal 1 mm slices with a field of view of 240 × 240 mm were recorded. Voxel size is $1.0 \times 1.0 \times 1.0$ mm, repetition time (TR) was set to 9 ms and echo time (TE) to 4 ms. For the FLAIR image, 144 contiguous axial 1.5 mm slices with a field of view of 230 × 185 mm have been obtained. The voxel size for these images is $1.0 \times 1.0 \times 1.5$ mm; TR was set to 104 ms, TE to 140 ms, and inversion time to 2,750 ms.

In addition to the training data a test data set from the same cohort is chosen to illustrate the performance of the final segmentation algorithm. MR parameters of the training and test data sets are identical.

[1] http://www.fil.ion.ucl.ac.uk/spm/

### 6.1.2 Modeling

In this section a binary regression model with spatially varying intercept is constructed to segment lesions which appear hyperintense in FLAIR images.

#### Reference lesion maps

As the lesion segmentation procedure presented here is a supervised classification algorithm, reference lesion segmentations are required for the training data set. To this end, lesion location was estimated by the lesion growth algorithm (LGA), an unsupervised MS lesion segmentation algorithm implemented in the LST toolbox for SPM. For all subjects the default settings have been used. Since the LGA requires to coregister the FLAIR image to the T1-weighted image prior to lesion segmentation, the resulting probability lesion maps are in the space of the T1-weighted image. The final binary reference lesion maps have been obtained by thresholding the probability lesion maps at 0.5.

#### Preprocessing and feature extraction

Two features are extracted from the available MR images, beginning with the position of each brain voxel in a standard (MNI, Evans et al., 1993) space. To this end, FLAIR images which have been coregistered to the T1-weighted images are normalized to MNI space using the "Normalize" function implemented in SPM. This routine creates an inverse deformation field which can be used to map MNI coordinates into the subject specific native space. Note that by using this procedure multiple voxels of the original MR image can have the same MNI coordinate. This is due to the fact that the MNI space is usually of lower dimension than the individual native space of the actual MR images. This way, a total number of 565,475 MNI coordinates are distributed over the individual MR images.

The second feature is the so-called lesion belief map. This image is produced by the following steps: First, the FLAIR image is roughly segmented into the three main tissue classes GM, WM, and CSF. Subsequently, FLAIR intensities are standardized by dividing each voxel by the mean of segmented GM. In addition, the mean of standardized GM voxels is subtracted from all FLAIR intensities. Only positive differences are kept, negative values are set to zero. Furthermore, the remaining differences are multiplied by a tissue probability map for WM which is obtained by applying the inverse deformation field from above to the tissue probability maps included in SPM. The resulting lesion belief map shows voxels that appear hyperintense in the FLAIR image and which are likely to be part of WM in healthy subjects, thus, possible lesion candidates.

### Training

In the training phase, the above features are efficiently combined using a voxel-wise logistic regression model which includes a spatially varying intercept. To be more precise, the following model is used:

$$y_{ij} \sim \mathrm{B}(\pi_{ij}) \;\; \text{with} \;\; \pi_{ij} = \frac{\exp(\eta_{ij})}{1+\exp(\eta_{ij})}.$$

Here, $y_{ij}$ is the value of the reference lesion map for the $j$th voxel and $i$th subject. The linear predictor is chosen to be

$$\eta_{ij} = \beta_0 + \beta_1 x_{ij} + \gamma_j$$

where $\beta_0$ is the overall intercept, $x_{ij}$ the value of the lesion belief map for the $j$th voxel and $i$th subject, and $\beta_1$ the corresponding effect of the lesion belief map. In addition, the vector $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_m)'$ is a spatially varying intercept. For the vector of fixed effects, $\boldsymbol{\beta} = (\beta_0, \beta_1)'$, a zero-mean Gaussian prior with precision matrix $1e^{-6}\boldsymbol{I}_2$, is used. The spatial effect, $\boldsymbol{\gamma}$, is modeled by the three-dimensional extension of the RW1 prior based on Kronecker-sum-penalties:

$$\boldsymbol{K} = \boldsymbol{K}_{n_z} \oplus (\boldsymbol{K}_{n_y} \oplus \boldsymbol{K}_{n_x}).$$

Here, $n_x, n_y$ and $n_z$ are the dimensions of the MNI template, i.e. $n_x = 121$, $n_y = 145$, and $n_z = 121$. Note that only 565,475 voxels (about 27%) of this template are active brain voxels and therefore included in $\boldsymbol{\gamma}$. Hence, it is necessary to select only relevant columns in $\boldsymbol{K}$ and to adjust its diagonal elements by the number of neighboring voxels accordingly. Furthermore, a sum-to-zero constraint is imposed on $\boldsymbol{\gamma}$ in order to guarantee identifiability of all regression coefficients. For the corresponding precision parameter, a Gamma distribution with parameters $a$ and $b$ is chosen. Setting these hyperparameters to small values results in a very rough spatial effect where small changes in lesion structure are overestimated. Thus, to guarantee an acceptable amount of smoothness a more informative prior for this precision parameter is chosen by setting $a = 100$ and $b = 1$. This choice is robust, i.e. smaller changes in $a$ and $b$ do not affect the final results.

The present data situation is comparable to the dense data situation in Chapter 5: Multiple observations are available for each node of the GMRF which is induced by $\boldsymbol{\gamma}$. Thus, it can be expected that all investigated approximation strategies perform well. However, as already mentioned in the introduction of this chapter the above model is fitted using the simplest strategy, that is, the modified sampling scheme and the IC factorization without fill-in are used. Except for the fixed effect, the precision parameter as well as five selected components of $\boldsymbol{\gamma}$, posterior moments are computed on-line after an initial burn-in period of 500 samples. In total, four chains of length 10,500 are generated.

### 6.1.3 Results

Mean runtime for sampling 1,000 iterations is about two hours and requires 9 GB RAM in total. Acceptance rates are about 0.91 for the fixed and 0.86 for the spatial effect.

#### Assessment of MCMC results

The highest potential scale reduction factor among all 565,478 parameters is 1.0023. This indicates that all chains converged to their corresponding stationary distributions. Sampling paths of $\kappa$, $\beta_0$, and one selected element of the vector $\boldsymbol{\gamma}$, i.e. $\gamma_{12865}$, are displayed in the first column of Figure 6.3. By visual inspection of these time series it can be seen that mixing is excellent without any convergence problems occurring, which implies that the chosen approximation strategy is sufficient for this data situation. In addition to the displayed trajectories, the second column of Figure 6.3 shows the corresponding estimated marginal posteriors using histograms derived from the samples of all chains. Furthermore, individual marginal posteriors obtained from each chain are indicated by kernel density estimators (black lines). As can be seen, the sampling scheme is able to estimate well shaped marginal posteriors which indicates good exploration of stationary distributions. Furthermore, in all cases results from the single chains agree well with the final distributions. Furthermore, for $\gamma_{12865}$ the marginal posterior is approximated by a Gaussian density using on-line computed posterior moments (red line). Agreement of this approximation with the histogram is satisfying which indicates that keeping only these values during the sampling process is sufficient for elements of the high-dimensional spatial effect.

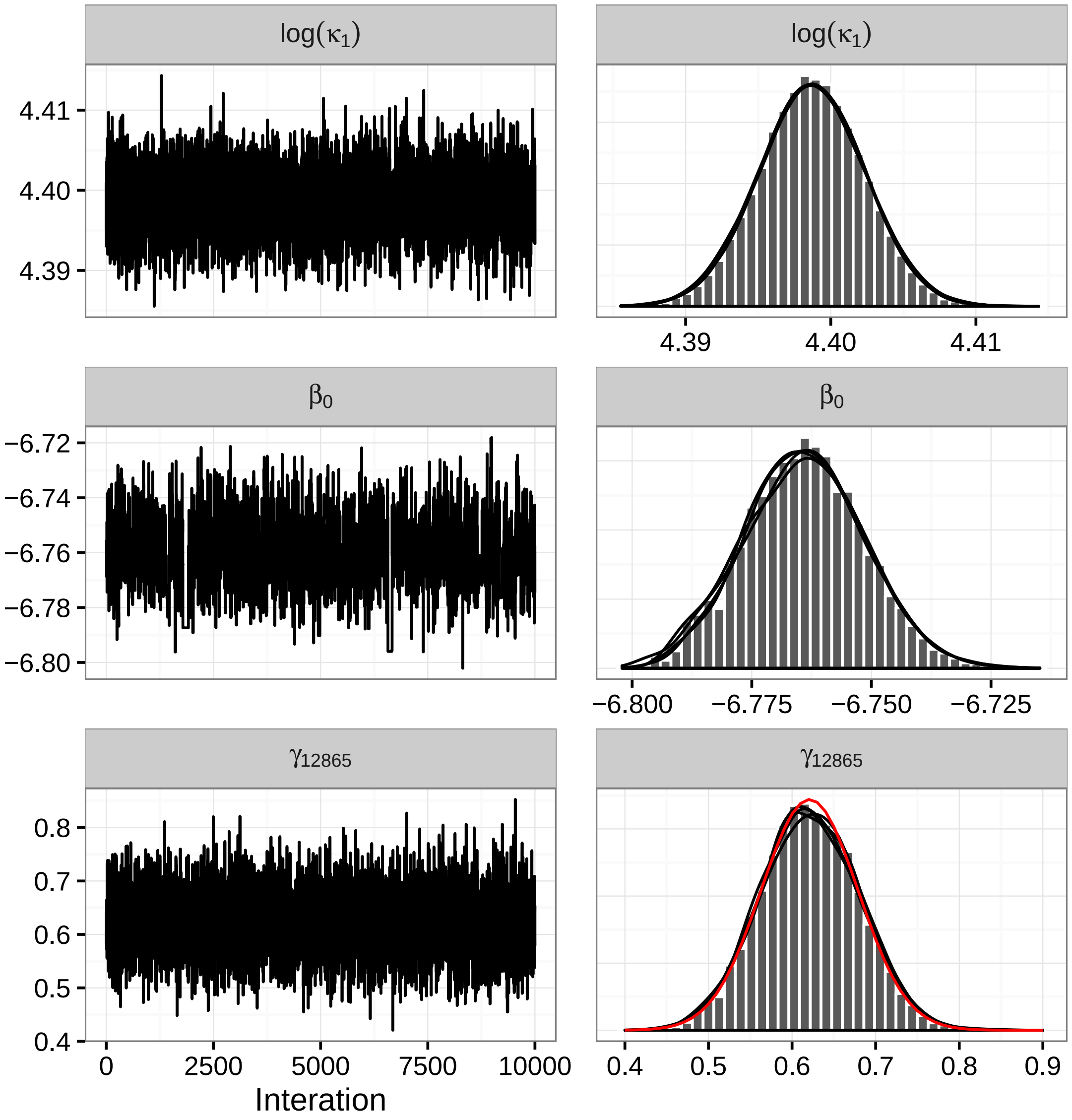


Figure 6.3 MCMC sampling paths (left column) and estimated marginal posteriors (right column) for selected parameters.

Posterior means for the spatially varying intercept are displayed in Figure 6.4. Overall, the estimated effect map has similarities to WM probability maps as they are used within other tissue segmentation algorithms. In particular, an increased risk for MS lesions can be observed in regions of WM around lateral ventricles including the corpus callosum. It can also be seen that the probability of observing a lesion decreases in the complete cerebellum. This may increase the risk of missing out infratentorial lesions in the final segmentation algorithm. This issue is revisited in Section 6.1.3 below. From Figure 6.4 it can also be seen that the estimated effect map is slightly non-symmetric, which is not surprising given

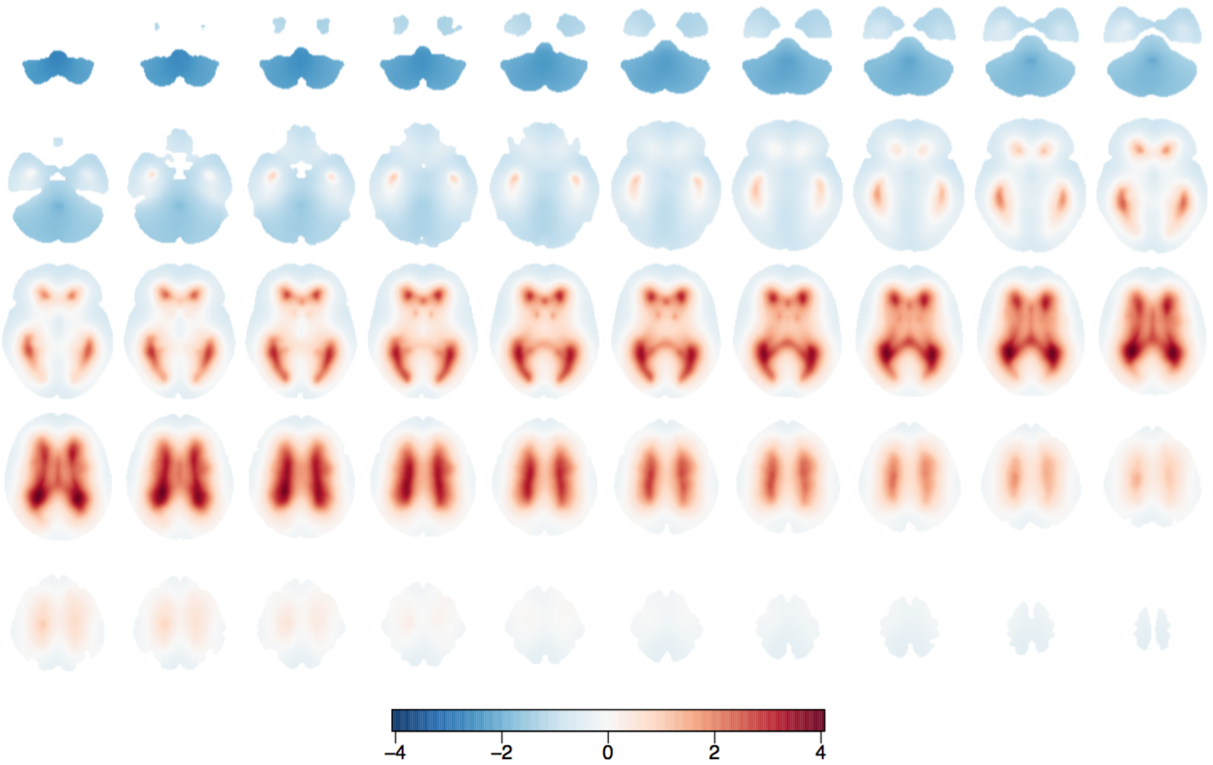


Figure 6.4 Estimated posterior means of the spatially varying intercept.

the relatively small number of patients that are included in the training data set. For the application to new data it is advantageous to symmetrize the effect map. To this end, the values of each slice are mirrored along their central lateral line and all voxel values are replaced by their corresponding mean.

### MS lesion segmentation

In order to apply the estimated model for lesion segmentation relevant features as introduced in Section 6.1.2 need to be extracted. Note that in contrast to the training phase, a T1-weighted image is not necessary as this image was only required in order to obtain reference lesion maps using the LGA. Thus, the following steps are performed in order to extract all relevant features. First, inverse deformation fields are estimated in order to map the posterior mean image of the spatially varying intercept from MNI space into subject specific native space. This yields $\hat{\gamma}$. Next, intensities of FLAIR images are corrected for bias field inhomogeneity using SPM's "Segment" function. From these images, the required

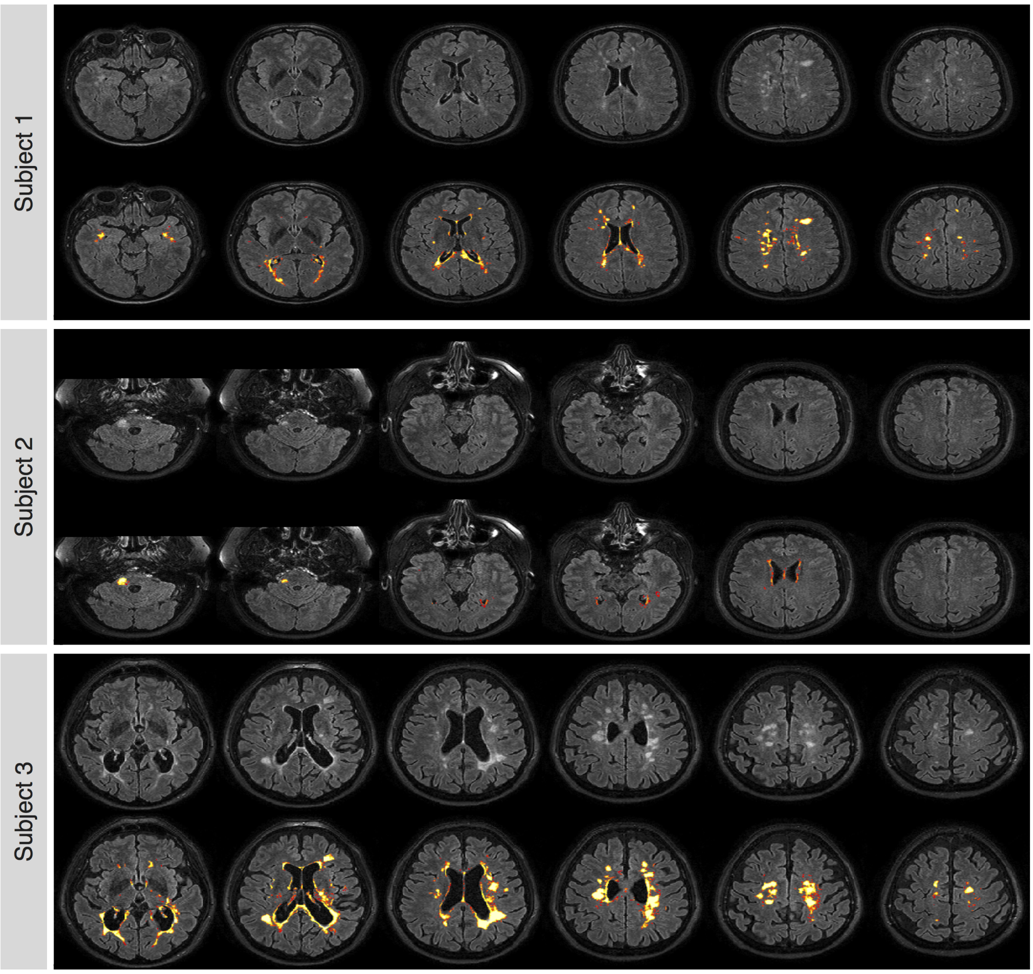


Figure 6.5 FLAIR images and estimated lesion probability maps for three subjects from the test data set.

lesion belief map is computed as explained in Section 6.1.2 which yields $x_j$, $j = 1, \ldots, m$. Combining this with the posterior means of $\beta_0$ and $\beta_1$ yields the following linear predictor

$$\hat{\eta}_j = \hat{\beta}_0 + \hat{\beta}_1 x_j + \hat{\gamma}_j.$$

Lesion segmentation is then performed by computing the lesion probability for each voxel using $\hat{\pi}_j = \exp(\hat{\eta}_j)/(1 + \exp(\hat{\eta}_j))$.

Usually, some post processing steps are required in order to obtain a clean lesion map. For the present segmentation method it is sufficient to omit all lesions that are smaller than 0.015 ml in volume.

The results obtained by this lesion segmentation algorithm for three subjects from the test data set are displayed in Figure 6.5. Here, FLAIR images as well as FLAIR images overlaid with the resulting probability lesion maps are shown. The first patient has been selected due to its complicated lesion pattern: Many small lesions are scattered all over the brain which complicates the segmentation of all affected tissue. As can be seen, the algorithm is able to detect most lesions very well, even tissue which appears only slightly hyperintense. With respect to lesion filling in T1-weighted images this can be seen as an advantage. However, it can also be seen that some smaller lesions on the transition from WM to cortical GM are missing. The results for the second subject demonstrate nicely that the approach is able to segment infratentorial lesions although this area is under-represented in the spatially varying intercept, see Figure 6.4. In addition, some light voxels mostly around the ventricles are detected as well, however, the estimated lesion probabilities for these voxels are rather low. Finally, the third patient has been selected because of its large lesion load. Here, the algorithm is able to segment all lesions while producing some false positives on the border of the cortex, a behavior which seems acceptable given the size of these false positives.

### 6.1.4 Summary

A binary regression model for the classification of voxels in lesion and non-lesion voxels has been presented in this section. By including a spatially varying intercept the model is able to (a) account for differences in lesion appearance across the brain and (b) provide valid results even for voxels where no lesions have been observed due to limited sample size of the training set. This leads to a high-dimensional regression coefficient which necessitates the approximation of certain steps of the fitting process. The selected approximation strategies perform well and produce results which can be used to formulate a useful lesion segmentation algorithm.

It is worth mentioning that the presented approach can easily be extended in order to include further MR modalities, such as T2- and PD-weighted images. In this context it may also be of interest to consider interaction effects of the intensities of different MR images. For example, one can use the fact that lesions appear hypointense in T1-weighted images but hyperintense in FLAIR, T2- and PD-weighted images. In contrast, CSF, for example, appears hypointense in T1-weighted and FLAIR images and hyperintense in T2- and PD-weighted images. This information can be accounted for by interactions of metric covariates as explained in Section 2.2.1. However, including additional MR modalities would also require to carry out further preprocessing steps, namely the coregistration of all MR sequences. Especially for patients with high lesion loads this step may be vulnerable to errors with respect to misalignment of identical brain structures.

Current experience with the above lesion segmentation algorithm shows that it also works well for images that are not part of the data set used here. However, if the tissue contrasts of these images are not comparable to the images used for training, the results obtained by this segmentation procedure may not produce reliable probability lesion maps. This should not come as a surprise since the presented approach is a supervised method. However, as long as training data is available the model parameters can easily be adjusted in order to render the algorithm useful to other data sets.

## 6.2 Differences in age-related atrophy of gray matter in MS patients

In this section the analysis of MR images is revisited. In particular, the hypothesis of a more pronounced age-related atrophy of GM in MS patients is analyzed using a voxel-wise regression model.

### 6.2.1 Introduction

Voxel-wise regression models are widely used for the analysis of MR images. For example, in first-level analyses of functional MRI (fMRI) time series analyses are performed for each voxel and subject. In a second-level analysis results from the first-level analysis are compared in a voxel-wise fashion along different subjects, groups or experimental conditions by a general linear model (Friston et al., 1994). As a further example consider the analysis of GM atrophy in structural MR images. Here the method of voxel-wise morphometry (VBM, Ashburner and Friston, 2000) has been widely accepted. This is subject of the following sections.

#### Voxel-based morphometry

VBM is a widely used approach for the analysis of GM atrophy along the brain. Here, individual images of local GM volume are compared in a voxel-wise manner either along groups by performing statistical hypothesis tests or correlated with explanatory covariates by fitting linear models. The usual procedure consists of smoothing the data prior to analysis, fitting the desired model for each brain voxel independently, and correcting the results for multiple comparisons in order to prevent an increase of the Type I error. Although Bayesian versions exist which eliminate the necessity for *post hoc* correction (Friston and Penny, 2003), spatial information is only considered by smoothing the images with a predetermined smoothing parameter prior to the analysis which represents a non-trivial modification of the original data. Instead, it would be favorable to include spatial information in the modeling step especially as it has been shown that this leads to an increase of the signal to noise ratio (Penny et al., 2005) and statistical power (Schmidt et al., 2013). In the following sections it is shown that the MCMC approach presented in the previous chapters is able to provide such a solution by fitting a model with millions of parameters with only moderate requirements on computational equipment.

#### Data and preprocessing

With respect to MS, a frequently asked and not yet sufficiently answered question is that if and where the correlation between age and GM atrophy differs between MS patients and healthy controls (HCs). This chapter tries to answer this question by analyzing the images of an age and gender matched cohort which contains the data of $n = 247$ subjects (168 MS patients and 79 HCs). The data set consists of normalized images of local GM volume which were estimated from T1-weighted sequences using the tissue segmentation pipeline implemented in VBM8[2]. For MS patients, lesions were filled in T1-weighted images prior to segmentation using the "Lesion filling" routine implemented in the LST toolbox. To this end, lesions were identified using the method presented in the previous chapter. All images were obtained at the Department of Neurology, Technische Universität München, Munich, Germany, using the same imaging protocol as outlined in Section 6.1.1.

### 6.2.2 Modeling

Given the $247 \times 1$ vector of voxel specific GM volume, $\boldsymbol{y}_j$, the following model is formulated:

$$\boldsymbol{y}_j \sim \mathrm{N}(\boldsymbol{\eta}_j, \kappa_j^{-1}\boldsymbol{I}_n)$$

[2] http://www.neuro.uni-jena.de/vbm/download/

with

$$\boldsymbol{\eta}_j = \gamma_{j1} + \gamma_{j2} \cdot \mathbf{age} + \gamma_{j3} \cdot \mathbf{sex} + \gamma_{j4} \cdot \mathbf{ms} + \gamma_{j5} \cdot \mathbf{age} \cdot \mathbf{ms}.$$

Here, **age**, **sex** and **ms** are 247 × 1 vectors that contain the centered age, dummy coded information about gender (male = 0, female = 1), and state of disease (HC = 0, MS = 1), respectively. For all coefficient vectors $\boldsymbol{\gamma}_k$, $k = 1, \ldots, 5$, the same three-dimensional spatial prior and the same brain mask as in the previous application are used. That is, only $m = 565{,}475$ voxels are considered and the entries of the structure matrices are adjusted accordingly. As no overall intercept is included in the model no linear constraints are necessary. For the corresponding precision parameters Gamma priors with shape and rate parameter $a_k$ and $b_k$ are used. Due to the relatively large sample size the effect of the hyperparameters on the final results is limited. However, in order to ensure smooth effect maps which are less sensitive to noise related artifacts, informative priors for the precision parameters are chosen by setting $a_k = 10{,}000$ and $b_k = 10$.

Note that additional complexity is induced by the fact that each voxel is assigned its own precision parameter for the Gaussian likelihood. However, updating these parameters is straightforward if independent Gamma priors with shape and rate parameter $a_y = 1$ and $b_y = 5e^{-5}$ are assumed. Then, the full conditional for the $j$th voxel is again a Gamma distribution with updated parameters $\tilde{a}_y = a_y + n/2$ and $\tilde{b}_y = b_y + 0.5(\boldsymbol{y}_j - \boldsymbol{\eta}_j)'(\boldsymbol{y}_j - \boldsymbol{\eta}_j)$, respectively.

As an approximation strategy the modified sampling scheme using the IC factorization without fill-in is used. Four independent chains of length 11,000 are generated and on-line calculation of posterior moments starts after 1,000 iterations. The above model represents a three-dimensional extension of the dense data situation which has been analyzed in the simulations chapter, thus it can be expected that the chosen approximation strategies work well.

### 6.2.3 Results

Computing 1,000 iterations requires about six hours and 7 GB RAM on working station A.

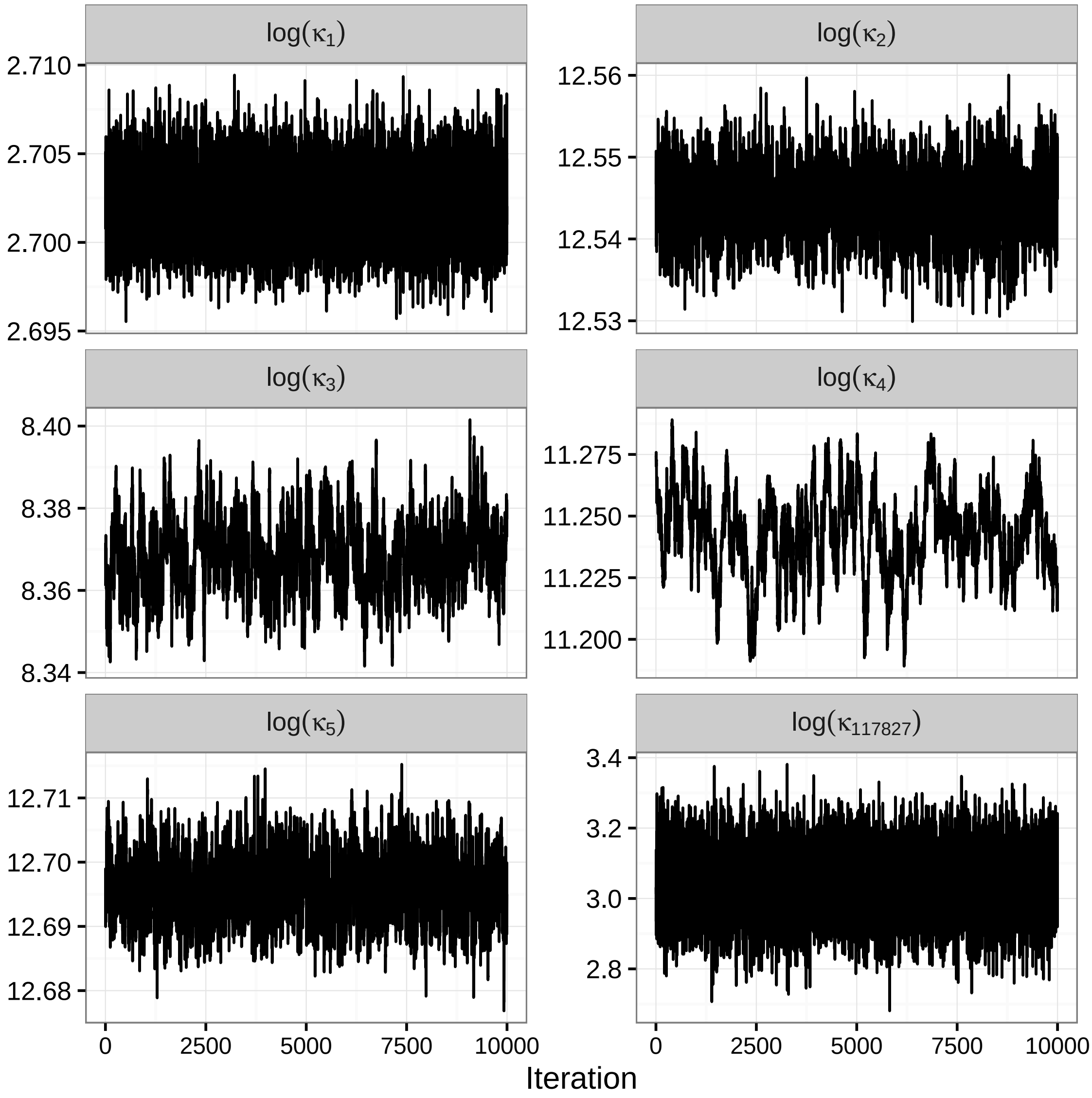


Figure 6.6 Generated chains of the precision parameters for one run.

### Assessing convergence and mixing of Markov chains

According to the potential scale reduction factor all chains converged to their stationary distributions (maximum $\hat{R} = 1.0072$). Sampled paths for the chains of $\log(\kappa_k)$, $k = 1, \ldots, 5$ and for one selected dispersion parameter, $\log(\kappa_{117827})$, of one run are displayed in Figure 6.6. Visual inspection of these chains indicate good mixing. In most cases, posterior distributions are explored rather fast. However, mixing is slower for $\kappa_3$ and $\kappa_4$ but the corresponding posteriors are still sufficiently explored, especially when considering the other chains (not shown).

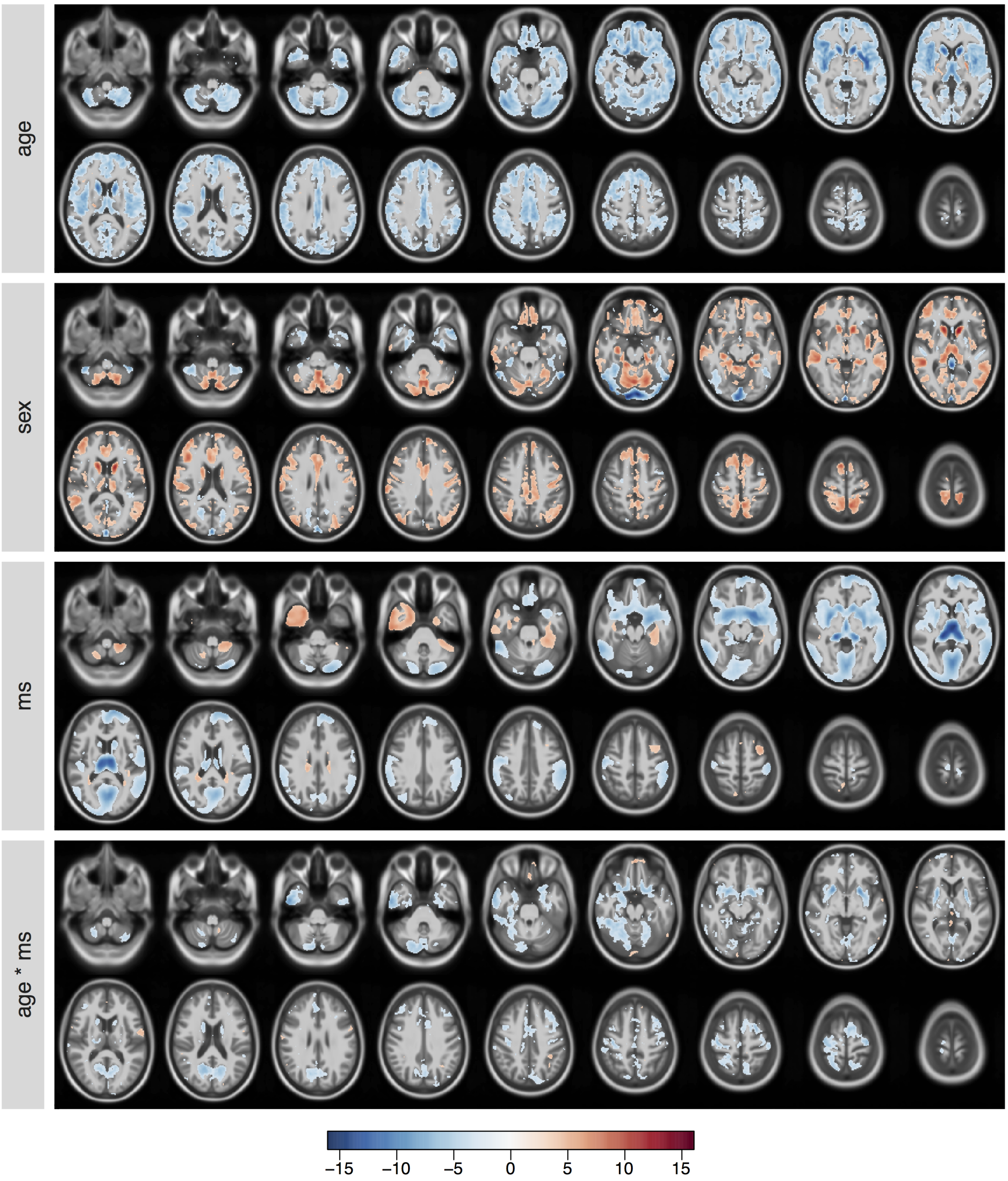


Figure 6.7 Estimated effects for age, sex, state of disease, and the interaction of age and state of disease.

### Assessing the estimated effects

For the final assessment of results the posterior moments obtained from all MCMC runs are combined. Figure 6.7 shows the estimated standardized coefficients (posterior mean divided by posterior standard deviation) for age, sex, state of disease, and the interaction of age and state of disease. For better orientation the estimated coefficient maps are overlaid on a mean T1-weighted image and only those voxels are colorized for which at least 99.9% of their posterior probability mass lies below or above zero. The first row depicts the effect of age on local GM volume for HCs. It can be seen that this effect is mostly negative and aligned all along the cerebral cortex. Regions where this effect is more emphasized are the left and right insula, the left and right putamen, as well as the left and right caudate nucleus. These results are in accordance with previous recorded effects, see for example Hutton et al. (2009). Females seem to have more GM volume at the left and right thalamus, left and right caudate nucleus, parts of the cerebellar cortex, and some minor regions of the cerebral cortex (second row). In addition, a smaller negative cluster, i.e. a region where females tend to have less GM concentration, can be observed near the visual cortex. The third row shows the results for $\gamma_4$, i.e. the effect of MS disease on local GM volume. It seems that MS patients have less GM volume in the region of the left and right caudate nucleus, the thalamus, putamen, and minor pronounced regions within the remaining cerebral cortex. In addition, a positive cluster within the right temporal lobe is noticeable. Finally, for the interaction effect of age and state of disease mostly clusters with negative signs can be observed, i.e. regions where GM atrophy is more pronounced for MS patients than for HCs. In particular, this includes the visual cortex as well as the putamen.

### 6.2.4 Summary

In this application a voxel-wise regression model with 3,392,855 parameters in total has been estimated successfully. Despite this large number of parameters it was able to perform the estimation with only moderate computational requirements. The fully Bayesian approach to voxel-wise regression has some serious advantages compared to classical or frequentist inference. First, due to the data driven regularization of regression coefficients the occurrence of artifacts is limited. Furthermore, it is no longer necessary to smooth the data prior to the analysis. This is an important feature as it eliminates one researchers degree of freedom (Simmons et al., 2011). Another advantage of the fully Bayesian approach is that no *post hoc* corrections for multiple testing need to be applied. This is of particular practical importance as it has been shown that during the last years severe errors occurred when applying correction methods in the context of voxel-wise regression models (Eklund et al., 2016). Finally, previous studies (Penny et al., 2005; Schmidt et al., 2013) indicate that the Bayesian approach is simply more powerful than the frequentist approach to voxel-wise regression with respect to uncover potential clusters of atrophy or activation.

Although one can find biological validations for some of the above results they need to be reproduced using different data sets before they can be further interpreted. After all, results from voxel-based regression of brain images obtained from different subjects must be interpreted with care: The number of preprocessing steps and the complex nature of MRI makes this type of analysis susceptible to artifacts induced by data preparation. In addition, it must be noted that the formulated model most probably does not include all relevant data which is needed to provide reproducible results. For example, other authors account for total intracranial volume as a possible confounder for different head sizes (Hutton et al., 2009). Further variables of interest are handedness (Good et al., 2001), level of education (Rzezak et al., 2015), and more detailed information of disease state, such as the total lesion volume. However, it should not be difficult to include this information once it is available and to apply the framework to an extended model.

## 6.3 Object-based co-localization by a spatial point process

In this final application a simple co-localization analysis is extended by additional spatial information.

### 6.3.1 Introduction

Understanding the spatial arrangement and interaction of cell components is a crucial task in molecular biology. Of particular interest is the question if specific cell functions can only be executed if certain components interact with each other. In this context interaction refers to spatial proximity and spatial correlation of some kind. The investigation of these interactions is the subject of co-localization analyses.

#### Object-based co-localization

In object-based co-localization the data obtained from fluorescence microscopy images are reduced to objects. Precise identification of these objects is feasible due to different coloration using fluorescence markers and image segmentation or object detection methods which are not of interest in this application. For each recorded object the distances to other objects and cellular structures of concern are recorded. Classical procedures of object-based

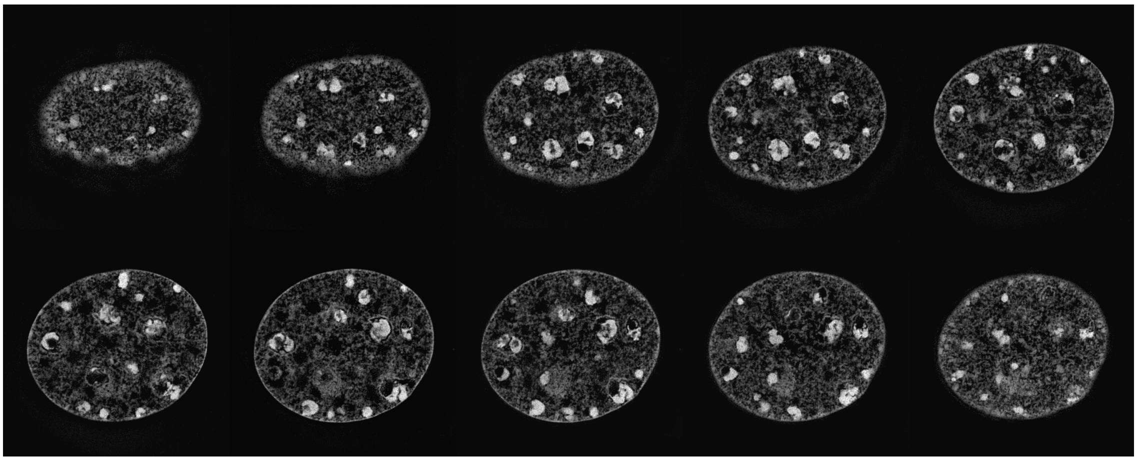


Figure 6.8 Selected slices of the blue layer (DNA intensity), in grayscale.

co-localization try to reveal possible interactions between objects by analyzing the so-called nearest-neighbor-distance co-localization measure, i.e. the number of objects that fall within a pre-defined distance (Lachmanovich et al., 2003). While simple to compute it has been found that this measure oversimplifies the complex nature of spatial proximity between subcellular structures (Helmuth et al., 2010). Therefore, generalizations of this analysis have been presented. For example, Helmuth et al. (2010) use spatial point processes in order to estimate interaction potentials as a function of object distances in a non-parametric fashion. In addition to the utilization of distances it has been noted that the location of the object in the cell itself may provide valuable information for the understanding of cell processes. For example, it has been shown that in human cells each chromosome can be assigned to a specific location in the nucleus (Bolzer et al., 2005). Thus, it seems natural to consider the spatial position of objects within a cell in the analysis. In this application, a spatial Poisson process which is able to account for this type of information is fitted to the data of a three-dimensional cell nucleus.

## Data

The data used in this application comes from the combination of 3D structured illumination microscopy (3D-SIM) and 3D fluorescence in situ hybridization (3D-FISH), see Markaki et al. (2012) for details on these methods. Object of interest is the nucleus of a human cell in the process of DNA replication shortly before cell division. The nucleus has been colorized using different fluorescent stains in order to identify specific components. In particular, three layers have been used to identify chromatin (DNA), and the two genes Ser2 and Pol3.

These structures were colorized in blue, red, and green, respectively. As an example Figure 6.8 displays selected slices of the blue layer (DNA intensity) in grayscale. The preprocessed image is of dimension 190 × 190 × 51 with a voxel size of 0.125 × 0.125 × 0.125 micrometers and 514,442 relevant voxels. For each voxel, dummy variables describing an affiliation to Ser2 (red layer), the distance to the next gene of the same kind as well as the next Pol3 gene (green layer) and DNA intensity, that is the intensity value of the blue layer, have been recorded.

### 6.3.2 Modeling

In this application, the spatial distribution and co-location of Ser2 genes that have been unveiled in the red layer are of interest. To investigate this the following log-linear Poisson model is set up:

$$y_i \sim \text{Po}(\exp(\eta_i)),\ i = 1, \dots, 514{,}442$$

with

$$\eta_i = \beta_0 + f_1(z_{1i}) + f_2(z_{2i}) + f_3(z_{3i}) + f_{\text{spatial}}(i).$$

In this formulation $y_i$ indicates if voxel $i$ belongs to a Ser2 gene. Note that this holds only for about 0.63% of all voxels, thus this data set is similar to the sparse data situation considered in Chapter 5. The metric covariates $\boldsymbol{z}_1$, $\boldsymbol{z}_2$, and $\boldsymbol{z}_3$ are the recorded distances to other nearest Ser2 genes, nearest Pol3 genes, and the voxels DNA intensity, respectively. The effects of these covariates are modeled using Bayesian P-splines as discussed in Section 2.2.1. To be more precise, design matrices $\boldsymbol{Z}_1$, $\boldsymbol{Z}_2$, and $\boldsymbol{Z}_3$ are constructed using B-spline basis functions of degree $D = 3$ and 30 equidistant knots over the corresponding covariate domain. The main effect of interest, that is, differences in the spatial distribution of Ser2 genes, is revealed by including $f_{\text{spatial}}$ into the linear predictor, i.e. by a spatially varying intercept. Overall, the linear predictor can be written as

$$\boldsymbol{\eta} = \beta_0 + \boldsymbol{Z}_1\boldsymbol{\gamma}_1 + \boldsymbol{Z}_2\boldsymbol{\gamma}_2 + \boldsymbol{Z}_3\boldsymbol{\gamma}_3 + \boldsymbol{\gamma}_4.$$

Prior distributions are set up as follows. For the overall intercept a non-informative GMRF prior $\text{N}(0, 1e^{-6})$ is chosen. The coefficients of the nonlinear smooth effects are modeled by RW2 priors. The dimension of the spatially varying intercept is 514,442 × 1.

Similar to the previous applications, a three-dimensional extension of the RW1 prior is specified for this regression coefficient. In addition, sum-to-zero constraints are imposed on all non-parametric effects in order to guarantee identifiability. For precision parameters independent weakly-informative Gamma distributions are chosen with $a_k = 1$ and $b_k = 5e^{-5}$ for $k = 1, \dots, 3$. Due to the low information content in the data the choice of hyperparameters for the precision of $\boldsymbol{\gamma}_4$ is rather sensitive with respect to the final results. Therefore, in order to support the estimation process and to promote a more smooth result for the spatial effect a more informative prior is chosen, i.e. a Gamma distribution with parameters $a_4 = 1000$ and $b_4 = 10$.

Note that the above model specification coincides with the definition of a log-Gaussian Cox process model: given the realizations of the GMRF components the process is a Poisson process (Diggle et al., 2013). Similar models have been discussed previously, for example by including multiple spatially structured effects (Illian et al., 2012).

Samples for the overall intercept and the coefficients of the smooth effects can be saved without any modification. However, due to the dimension of $\boldsymbol{\gamma}_4$, the approximation strategies introduced in Chapter 4 need to be applied. As only one data point is connected to each voxel and each voxel is assigned an element of $\boldsymbol{\gamma}_4$ it is obvious that similar problems must be expected as for the sparse data set considered in the simulation studies in Section 5.1.1, which includes problems with block updating strategies. Indeed, attempts with these strategies, especially using the stochastic Chebyshev approximation of log-determinants, did reveal convergence problems of the Markov chain for $\kappa_4$. Thus, again the modified sampling scheme is applied and for preconditioning the IC factorization without additional fill-in is used. With this strategy four independent chains of length 20,000 and an initial burn-in period of length 1,000 are produced.

### 6.3.3 Results

Generating 1,000 iterations for the above model requires about two hours and less than 2 GB RAM on working station A. Acceptance rates are as follows: For the overall intercept 88% of all proposals are accepted. For $\boldsymbol{\gamma}_1$, $\boldsymbol{\gamma}_2$, and $\boldsymbol{\gamma}_3$ acceptance rates are about 45, 29, and 66%, respectively. Finally, for the spatially varying intercept 97% of the proposed samples are accepted.

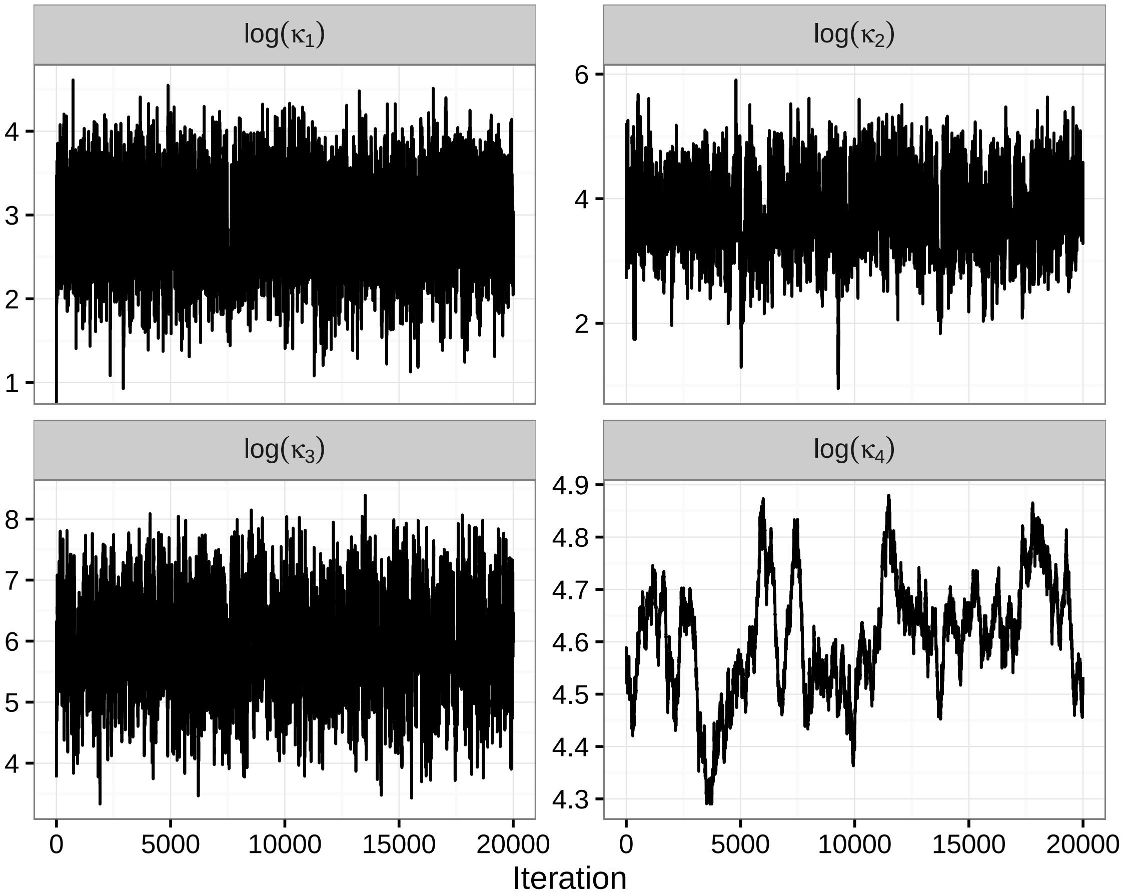


Figure 6.9 Generated chains of the precision parameters for one run.

### Assessing convergence and mixing of Markov chains

The estimated potential scale reduction factors indicate convergence of all chains (maximum $\hat{R} = 1.01$), except for $\kappa_4$ ($\hat{R} = 1.12$). Figure 6.9 displays the generated chains of the precision parameters for one run. While mixing for $\kappa_1, \ldots, \kappa_3$ is excellent, there is definitely room for improvement for $\kappa_4$. Here, realizations are characterized by strong dependencies which result in very slow mixing. However, the chains of the remaining runs yield similar results so that there is no reason to believe that the algorithm diverges, it just requires more time to fully explore the posterior. Thus, the results for this parameter can still be deemed useful.

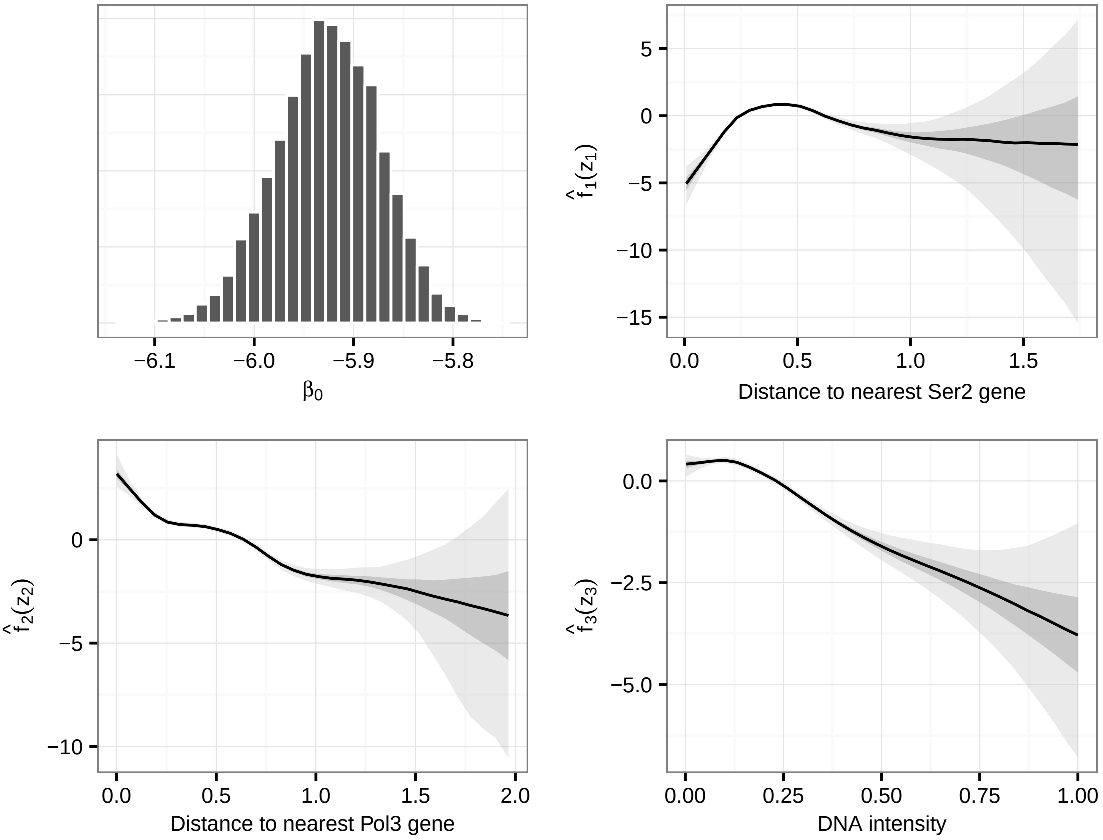


Figure 6.10 Estimated marginal posterior for the overall intercept (upper left panel) and estimated smooth effects for all continuous covariates. The black lines display the posterior median and the shaded regions correspond to the equal-tailed 50% and 95% credible intervals.

### Assessing the estimated effects

The following results are constructed from the results of all runs. The marginal posterior of the overall intercept is displayed in the upper left panel of Figure 6.10. The 95% credible interval for this parameter is $[-6.03, -5.83]$. This indicates that the overall probability of observing a Ser2 gene within the cell nucleus is rather small which is in accordance with the sparse nature of the problem. From the upper right panel of Figure 6.10 it can be seen that the probability of observing a Ser2 gene next to a gene of the same kind is increased if their distance is between 0.25 and 0.6 nanometers, that is, within these distances Ser2 genes tend to cluster. Moreover, up to a distance of 0.65 nanometers Ser2 genes are more likely located near Pol3 genes (lower left panel of Figure 6.10): Up to about 0.65 nanometers distance the estimated smooth effect is positive. Finally, the effect of the voxels DNA intensity is positive up to an intensity of about 0.2 and negative afterwards which indicates that Ser2 is more likely in regions with low DNA intensity (lower right panel of Figure 6.10).

Figure 6.11 depicts the standardized coefficients (posterior mean divided by posterior standard deviation) for the spatially varying intercept. Here, 35 successive slices of the cell nucleus have been selected. It can be seen that it is more likely to observe Ser2 genes at the edge of the nucleus; for most voxels in the centre of the cell negative coefficients are estimated. Given the magnitude of these standardized coefficients it becomes obvious that the information about the location inside the nucleus may not be of practical relevance to the question at hand, at least compared to the effect sizes of the other covariates. Furthermore, the variation along slices ($z$-direction) is less pronounced than along the other dimensions ($x$- and $y$-direction). This information may be used to ignore the $z$-direction of the cell, that is, the three-dimensional spatial effect may be reduced to a two-dimensional spatial effect. This would increase the information available for each pixel and, most probably, stabilize the estimation process.

### 6.3.4 Summary

Due to the sparse nature of the problem the proposed MCMC algorithm suffers from similar problems as the sparse data situation analyzed in Chapter 5. In particular, mixing of Markov chains for the precision parameter of the spatially varying intercept is insufficient and longer runs are necessary in order to fully explore the corresponding marginal posterior. Despite this problem it has been shown that it is possible to derive useful results with moderate computational equipment.

Modeling the distances between objects of interest by nonlinear effects presents a simple but yet efficient way to perform object-based co-localization under the consideration of multiple layers. It has been shown that it is possible to include additional spatial information, although for the data set considered here it may not be necessary to do so. If further investigations do not confirm the necessity of accounting for the three-dimensional information it may be reasonable to collapse the data structure and to consider a simpler two-dimensional effect. Furthermore, the biological findings must be replicated as the analysis performed is based on single-cell data.

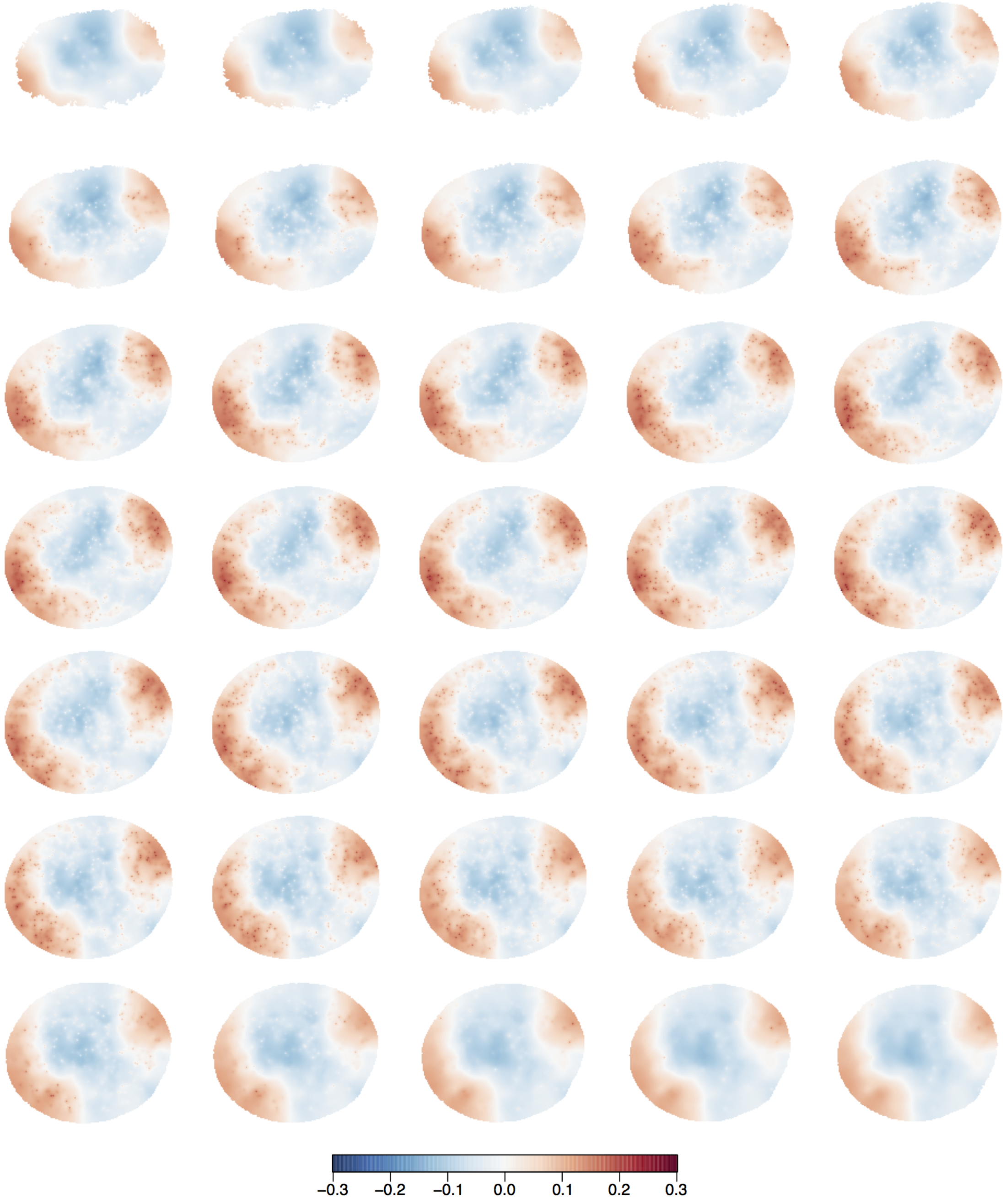


Figure 6.11 Standardized coefficients for the spatially varying intercept.

# 7 Summary and outlook

## 7.1 Summary

Main objective of this thesis was to find a way to estimate regression models with high-dimensional coefficients with low to moderate computational equipment. To this end, the framework of STAR models has been chosen as it allows to include a large variety of different effect types including spatial effects. In order to estimate models with high-dimensional regression coefficients an MCMC algorithm has been constructed. The computational requirements can be described as low to moderate. Thus, the algorithm allows the applied statistician to perform appropriate inference for large-scale time-insensitive problems without the need of well equipped working stations.

Without a question, deterministic methods for fitting STAR models have clear advantages compared to MCMC based inference. For example, they are usually faster and do not depend on mixing and convergence of Markov chains. This can be beneficial with respect to the complete modeling process which can be seen as an iterative procedure in which model fitting and model checking complement each other. However, the discussion in Chapter 4 revealed that deterministic methods either violate important requirements (sparsity) or are not flexible enough to account for application-specific features (large number of precision or dispersion parameters). In contrast, MCMC based inference meets all requirements and successfully addresses special data situations. The computational bottlenecks which are associated with this method, namely sampling from large-scale Gaussians and computing the log-determinant of huge matrices, were successfully eliminated. With regard to sampling this could be achieved by Krylov subspace methods: First, the conjugate gradient method is applied in order to compute the mean of the proposal. Next, the Lanczos algorithm is used to obtain an approximate sample from the zero mean Gaussian. Finally, the sample may be corrected in order to adjust for linear constraints. In order to improve convergence of these models, preconditioning is necessary.

In most cases the incomplete Cholesky factorization with zero fill-in has been found to be sufficient.

A modified sampling scheme has been used in order to circumvent the necessity of the calculation of log-determinants of huge matrices. The resulting algorithm does not only convince with its performance in the simulation studies conducted in Chapter 5, it is also preferable to the alternative methods in terms of computational complexity. The deterministic approximation of the log-determinant using the ICT factorization as well as the stochastic Chebyshev approximation yielded satisfying results for the direct approximation of the log-determinant. However, with respect to the final results the MCMC error was not able to compensate the error introduced by these approximations.

The simulation studies conducted in Chapter 5 have demonstrated situations in which the proposed MCMC framework produces reliable results. Especially in settings where sufficient information in the data is present the estimation of large-scale effects is well supported. In this case results can be generated that are extremely close to the one obtained by the INLA approach. However, in situations where the information in the data is sparse problems with convergence of Markov chains of precision parameters can be observed. Particularly, the strategies which approximate the log-determinants show a poor behavior in these situations. Furthermore, the modified sampling scheme requires more iterations to explore the corresponding posteriors sufficiently.

In the last chapter it has been shown that the proposed MCMC framework can be applied successfully to a broad range of applications. First, a supervised approach for lesion segmentation was presented. Here, a regression model with a high-dimensional spatial effect was fitted. In Section 6.2 a Gaussian response model with five of these effects was applied in the context of voxel-based morphometry. Finally, a spatial poisson process was used to analyze the spatial distribution of genes in the nucleus of a human cell.

## 7.2 Outlook

The framework described in this thesis is not restricted to the analysis of three-dimensional medical or biological images as performed in the last chapter. Instead, the MCMC algorithm is applicable to a wide range of different applications. For example, the data of different medical imaging techniques such as EEG signals can be processed by generalized additive regression models (Meulman et al., 2015) and, thus, present a possible application. Also, inverse problems within the analysis of non-medical digital images, such as deblurring and sharpening noisy images (Bardsley, 2012), could be analyzed. Further applications may be found when modeling large time series as occurred previously when forecasting electricity consumption (Gaillard and Goude, 2015), in the field of articulography (Wieling et al., 2015), and monitoring environmental data (Elayouty et al., 2016).

Having shown how the classical Gibbs sampler can be applied to large-scale problems, the improvement of the algorithm has to be the next logical step. Of particular importance are situations where Markov chains show slow mixing properties. This applies particularly to situations where the information of the data is sparse, such as the second simulation setup in Chapter 5 and the microscopy application in Section 6.3. Over the years different methods have been proposed to improve mixing of Markov chains. For example, a non-centered version of the Gibbs sampler (Papaspiliopoulos and Roberts, 2003) can be used to dissolve the dependency of $\boldsymbol{\gamma}$ and $\kappa$. Note that connections between the preconditioned Lanczos sampler and the non-centered parametrization are already discussed in Simpson et al. (2013). Further promising methods to improve mixing of Markov chains are the partially collapsed Gibbs sampler (van Dyk and Park, 2008) and the interweaving strategy proposed by Yu and Meng (2011). However, care must be taken as these approaches need to fulfill all necessary requirements for working on large-scale problems (Section 4.1.1). Note that updating $\boldsymbol{\gamma}$ and $\kappa$ jointly as proposed by Knorr-Held and Rue (2002) and performed in the simulation studies in Chapter 5 did not result in improved mixing behavior. Most probably, better approximations for the log-determinant are required in order to apply this strategy successfully. In addition, the results of Knorr-Held and Rue suggests that larger blocks may be necessary. However, this would increase the computational requirements dramatically.

Increasing the efficiency of the MCMC algorithm also includes an improved implementation in order to speed up the fitting process. This can be achieved in various ways. As a first step the algorithm could be implemented in a lower-level programming language. In addition, new technological achievements offer different possibilities of parallelization and thus promising a significant gain in computation time. For example, the most demanding step within the Krylov subspace methods is the computation of matrix-vector products. Therefore, parallelization of matrix multiplications seems to be a good starting point. Note that the Matlab implementation used in this thesis already makes use of this feature. Another approach is to parallelize the Gibbs sampler for which different suggestions have been made in the literature, see for example Doshi-Velez et al. (2009)

and Gonzalez et al. (2011). However, it needs to be proven that these approaches do not counteract on crucial assumptions that are necessary for working in large-scale settings. Probably the simplest and most naive approach of parallelization is the generation of multiple chains on independent processing units. For example, sampling 10,000 realizations distributed over 100 independent MCMC chains only requires the generation of 100 samples for each chain, a task which can be performed rather quickly. Depending on the computational requirements such massive parallelizations can be achieved by using GPUs or simple single board computers which can usually be purchased for a reasonable price. However, fast convergence of the MCMC algorithm is the main requirement for this parallelization strategy. Otherwise, the advantages of parallel setups may be undermined by long burn-in periods.

An interesting extension of STAR models can be achieved by the framework of generalized additive models for location, scale and shape (GAMLSS, Rigby and Stasinopoulos, 2005). Here, in addition to the conditional mean other distributional parameters, in particular the variance, skewness and kurtosis, can be modeled by additive predictors and specific link functions. First attempts within Bayesian setups seem promising (Klein et al., 2015). With respect to the analysis of medical images the GAMLSS framework is of particular interest for voxel-wise regression models as performed in Section 6.2. Here, independent dispersion parameters have been estimated for each voxel. Due to the spatial structure of the data it seems natural to include this information in the estimation of these parameters.

# Appendix

# A Implementation details

## A.1 Working stations and software

All calculations by the proposed MCMC algorithm have been performed on the following machine:

> **Working station A:** MacBook Pro (Retina, 13″, late 2013) running on OS X El Capitan (Version 10.11.4). Processor: 2.8 GHz Intel Core i7. Memory: 16 GB 1600 MHz DDR3.

Due to limited memory of this machine another working station has been used in order to obtain the results for the INLA approach in Chapter 5. This machine has the following specifications:

> **Working station B:** Self-made desktop PC running Ubuntu Linux (Version 12.04). Processor: AMD Phenom II x4 955. Memory: 32 GB 667 MHz DDR3.

All code was implemented in MATLAB[1] version 8.3.0.532 (R2014a) on working station A, and version 7.14.0.739 (R2012a) on working station B.

## A.2 Krylov subspace methods

Up to three different Krylov subspace methods are used in this thesis: A preconditioned version of the conjugate gradient method for solving linear systems, a preconditioned Lanczos algorithm for approximate sampling, and a Lanczos algorithm for the approximation of extreme eigenvalues. Pseudocode for the first two algorithms is provided next, for the latter and the stochastic Chebyshev expansion see Section A.3.

[1] www.mathworks.com/products/matlab/.

**Algorithm 2** Preconditioned conjugate gradient algorithm.

```
Set $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{Q}\boldsymbol{\mu}_0$, $\tilde{\boldsymbol{r}}_0 = \boldsymbol{M}^{-1}\boldsymbol{r}_0$, and $\boldsymbol{p}_0 = \boldsymbol{M}^{-T}\tilde{\boldsymbol{r}}_0$.
for $j = 1, \dots, r$ do
    $\alpha_j = ||\tilde{\boldsymbol{r}}_j||_2 / \langle \boldsymbol{Q}\boldsymbol{p}_j, \boldsymbol{p}_j \rangle$
    $\boldsymbol{\mu}_{j+1} = \boldsymbol{\mu}_j + \alpha_j \boldsymbol{p}_j$
    $\tilde{\boldsymbol{r}}_{j+1} = \tilde{\boldsymbol{r}}_j - \alpha_j \boldsymbol{M}^{-1}\boldsymbol{Q}\boldsymbol{p}_j$
    $\beta_j = ||\tilde{\boldsymbol{r}}_{j+1}||_2 / ||\tilde{\boldsymbol{r}}_j||_2$
    $\boldsymbol{p}_{j+1} = \boldsymbol{M}^{-T}\tilde{\boldsymbol{r}}_{j+1} + \beta_j \boldsymbol{p}_j$
end for
```

### A.2.1 Preconditioned conjugate gradients

A version of the preconditioned conjugate gradients algorithm for solving the linear system $\boldsymbol{Q}\boldsymbol{\mu} = \boldsymbol{b}$ for symmetric preconditioners $\boldsymbol{M} = \boldsymbol{M}_L = \boldsymbol{M}_R$ is displayed in Algorithm 2. This version is taken from Saad (2003, Algorithm 9.2). Note that there exist a wide variety of different – and possibly more efficient – implementations of the preconditioned conjugate gradients algorithm. In particular, versions that use parallelization can be found in the literature, see for example O'Leary (1987) and Di Brozolo and Robert (1989) for early applications as well as Saad (2003, Chapter, 11) for an overview. Often, special implementations can be found for specific software packages. In this thesis, Matlabs implementation in the `pcg` function was used. For the computation of the IC and ICT preconditioners Matlabs `ichol` function has been used.

**Algorithm 3** Preconditioned Lanczos algorithm for approximate sampling.

```
Set $\boldsymbol{v}_0 = \boldsymbol{0}$, $\beta_1 = 0$ and initialize $\boldsymbol{v}_1$
for $j = 1, \dots, r$ do
    $\boldsymbol{a} = \boldsymbol{M}^{-1}\boldsymbol{v}_j$
    $\boldsymbol{b} = \boldsymbol{Q}\boldsymbol{a}$
    $\boldsymbol{w} = \boldsymbol{M}^{-T}\boldsymbol{b} - \beta_j \boldsymbol{v}_{j-1}$
    $\alpha_j = \boldsymbol{w}'\boldsymbol{v}_j$
    $\boldsymbol{w} = \boldsymbol{w} - \alpha_j \boldsymbol{v}_j$
    $\beta_{j+1} = ||\boldsymbol{w}||_2$
    $\boldsymbol{v}_j = \boldsymbol{w} / \beta_{j+1}$
end for
```

**Algorithm 4** Lanczos algorithm for approximation of extreme eigenvalues

```
Set $\beta_1 = 0$ and initialize $\boldsymbol{v}_0 = \boldsymbol{u}/||\boldsymbol{u}||_2$ with $\boldsymbol{u} \sim \mathrm{U}(0, 1)$
for $j = 1, \ldots, r$ do
    $\boldsymbol{w} = \boldsymbol{Q}\boldsymbol{v}_j - \beta_j \boldsymbol{v}_{j-1}$
    $\alpha_j = \boldsymbol{w}'\boldsymbol{v}_j$
    $\boldsymbol{w} = \boldsymbol{w} - \alpha_j \boldsymbol{v}_j$
    $\beta_{j+1} = ||\boldsymbol{w}||_2$
    $\boldsymbol{v}_j = \boldsymbol{w}/\beta_{j+1}$
end for
```

### A.2.2 Preconditioned Lanczos algorithm for approximate sampling

For approximate sampling the preconditioned Lanczos algorithm as given in Algorithm 3 has been used, which is Algorithm 1 extended by the preconditioning steps discussed in Section 4.2.3. The same IC and ICT preconditioners as for Algorithm 2 were used. The result of Algorithm 3 are coefficients $\alpha_j$ and $\beta_j$, $j = 1, \ldots, r$, which are used to create matrix $\boldsymbol{T}_r$ given in (4.3). Subsequently, an approximate sample $\boldsymbol{x}$ from $\mathrm{N}(\boldsymbol{0}, \boldsymbol{Q}^{-1})$ is obtained by calculating $\tilde{\boldsymbol{x}}^*$ according to (4.4) and correcting for preconditioning, i.e. $\boldsymbol{x} = \boldsymbol{M}^{-1}\tilde{\boldsymbol{x}}^*$. As a stopping rule for Algorithm 3 Chow and Saad (2014) suggest to monitor the relative change between $\tilde{\boldsymbol{x}}^*_j$ and $\tilde{\boldsymbol{x}}^*_{j-1}$.

## A.3 Approximation of log-determinants

### A.3.1 Stochastic Chebyshev expansion

The complete algorithm for the stochastic Chebyshev expansion of the log-determinant of $\boldsymbol{Q}$ can be found in pseudocode in Han et al. (2015). In addition, Matlab code is available online[2].

### A.3.2 Lanczos algorithm for extreme eigenvalues

For the stochastic Chebyshev expansion it is required to provide rough estimates for the smallest and largest eigenvalues of $\boldsymbol{Q}$. This was the original task of the algorithm proposed by Lanczos (1950). Thus, the Lanczos algorithm in its original form (Algorithm 4) can be used for this task. As a results this algorithm generates coefficients $\alpha_j$ and $\beta_j$, $j = 1, \ldots, r$,

[2] https://sites.google.com/site/mijirim/logdet_code.zip, visited on January 13th, 2016.

which are used to formulate matrix $\boldsymbol{T}_r$ according to (4.3). The extreme eigenvalues of $\boldsymbol{Q}$ can then be approximated by the smallest and largest eigenvalues of $\boldsymbol{T}_r$. The algorithm is stopped if no more significant changes in these values are observed.

# List of Figures

# References

Aït-Ali, L. S., Prima, S., Hellier, P., Carsin, B., Edan, G., and Barillot, C. (2005). Strem: a robust multidimensional parametric method to segment MS lesions in MRI. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2005*, pages 409–416. Springer.

Akselrod-Ballin, A., Galun, M., Gomori, J. M., Filippi, M., Valsasina, P., Basri, R., and Brandt, A. (2009). Automatic segmentation and classification of multiple sclerosis in multichannel MRI. *Biomedical Engineering, IEEE Transactions on*, 56(10):2461–2469.

Albert, J. H. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679.

Anbeek, P., Vincken, K. L., van Osch, M. J., Bisschops, R. H., and van der Grond, J. (2004). Probabilistic segmentation of white matter lesions in MR imaging. *NeuroImage*, 21(3):1037–1044.

Andrews, D. F. and Mallows, C. L. (1974). Scale mixtures of normal distributions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(1):99–102.

Ashburner, J. and Friston, K. J. (2000). Voxel-based morphometry–the methods. *Neuroimage*, 11(6):805–821.

Aune, E., Eidsvik, J., and Pokern, Y. (2013). Iterative numerical methods for sampling from high dimensional Gaussian distributions. *Statistics and Computing*, 23(4):501–521.

Aune, E., Simpson, D. P., and Eidsvik, J. (2014). Parameter estimation in high dimensional Gaussian distributions. *Statistics and Computing*, 24(2):247–263.

Bai, Z., Fahey, G., and Golub, G. (1996). Some large-scale matrix computation problems. *Journal of Computational and Applied Mathematics*, 74(1):71–89.

Bai, Z. and Golub, G. H. (1996). Bounds for the trace of the inverse and the determinant of symmetric positive definite matrices. *Annals of Numerical Mathematics*, 4:29–38.

Bardsley, J. M. (2012). MCMC-based image reconstruction with uncertainty quantification. *SIAM Journal on Scientific Computing*, 34(3):A1316–A1332.

Barry, R. P. and Pace, K. R. (1999). Monte Carlo estimates of the log determinant of large sparse matrices. *Linear Algebra and its applications*, 289(1):41–54.

Benzi, M. and Tuma, M. (1999). A comparative study of sparse approximate inverse preconditioners. *Applied Numerical Mathematics*, 30(2):305–340.

Besag, J. (1989). Towards Bayesian image analysis. *Journal of Applied Statistics*, 16(3):395–407.

Besag, J. and Higdon, D. (1999). Bayesian analysis of agricultural field experiments. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(4):691–746.

Besag, J., York, J., and Mollié, A. (1991). Bayesian image restoration, with two applications in spatial statistics. *Annals of the Institute of Statistical Mathematics*, 43(1):1–20.

Bezdek, J. C., Hall, L., and Clarke, L. (1992). Review of MR image segmentation techniques using pattern recognition. *Medical physics*, 20(4):1033–1048.

Bickel, P. J., Brown, J. B., Huang, H., and Li, Q. (2009). An overview of recent developments in genomics and associated statistical methods. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 367(1906):4313–4337.

Bishop, C. (2006). *Pattern Recognition and Machine Learning.* Springer-Verlag New York.

Bolzer, A., Kreth, G., Solovei, I., Koehler, D., Saracoglu, K., Fauth, C., Müller, S., Eils, R., Cremer, C., Speicher, M. R., and Cremer, T. (2005). Three-dimensional maps of all chromosomes in human male fibroblast nuclei and prometaphase rosettes. *PLoS Biol*, 3(5):e157.

Brezger, A. and Lang, S. (2006). Generalized structured additive regression based on Bayesian P-splines. *Computational Statistics & Data Analysis*, 50(4):967–991.

Chan, T., Gilbert, J. R., and Teng, S.-H. (1995). Geometric spectral partitioning. Technical Report CSL-94-15, Xerox PARC.

Chen, Z. (1993). Fitting multivariate regression functions by interaction spline models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 55(2):473 – 491.

Chow, E. and Saad, Y. (1998). Approximate inverse preconditioners via sparse-sparse iterations. *SIAM Journal on Scientific Computing*, 19(3):995–1023.

Chow, E. and Saad, Y. (2014). Preconditioned Krylov subspace methods for sampling multivariate Gaussian distributions. *SIAM J. Sci. Comput.*, 36(2):A588 – A608.

Clayton, D. G. (1996). Generalized linear mixed models. In Gilks, W. R., Richardson, S., and Spiegelhalter, D., editors, *Markov Chain Monte Carlo in practice.* Chapman & Hall.

Compston, A. and Coles, A. (2008). Multiple sclerosis. *The Lancet*, 372(9648):1502–1517.

Cosgrove, J., Diaz, J., and Griewank, A. (1992). Approximate inverse preconditionings for sparse linear systems. *International Journal of Computer Mathematics*, 44(1-4):91–110.

Cowles, M. K. and Carlin, B. P. (1996). Markov Chain Monte Carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, 91(434):883—904.

Crayton, H., Heyman, R. A., and Rossman, H. S. (2004). A multimodal approach to managing the symptoms of multiple sclerosis. *Neurology*, 63(11 suppl 5):S12–S18.

Cuthill, E. and McKee, J. (1969). Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 1969 24th National Conference*, ACM '69, pages 157–172, New York, NY, USA. ACM.

de Boor, C. (1978). *A Practical Guide to Splines.* Springer, Berlin.

Di Brozolo, G. R. and Robert, Y. (1989). Parallel conjugate gradient-like algorithms for solving sparse nonsymmetric linear systems on a vector multiprocessor. *Parallel Computing*, 11(2):223–239.

Diggle, P. and Ribeiro, P. J. (2007). *Model-based Geostatistics.* Springer New York.

Diggle, P. J., Moraga, P., Rowlingson, B., Taylor, B. M., et al. (2013). Spatial and spatio-temporal log-gaussian cox processes: extending the geostatistical paradigm. *Statistical Science*, 28(4):542–563.

Doshi-Velez, F., Mohamed, S., Ghahramani, Z., and Knowles, D. A. (2009). Large scale nonparametric Bayesian inference: Data parallelisation in the indian buffet process. In *Advances in Neural Information Processing Systems*, pages 1294–1302.

Dunn, K. W., Kamocka, M. M., and McDonald, J. H. (2011). A practical guide to evaluating colocalization in biological microscopy. *American Journal of Physiology-Cell Physiology*, 300(4):C723–C742.

Eilers, P. H. C. and Marx, B. D. (1996). Flexible smoothing with B-splines and penalties. *Statistical Science*, 11(2):89—102.

Eklund, A., Nichols, T. E., and Knutsson, H. (2016). Cluster failure: Why fMRI inferences for spatial extent have inflated false-positive rates. *Proceedings of the National Academy of Sciences*, 113(28):7900–7905.

Elayouty, A., Scott, M., Miller, C., Waldron, S., and Franco-Villoria, M. (2016). Challenges in modeling detailed and complex environmental data sets: a case study modeling the excess partial pressure of fluvial $CO_2$. *Environmental and Ecological Statistics*, 23(1):65–87.

Evans, A. C., Collins, D. L., Mills, S., Brown, E., Kelly, R., and Peters, T. M. (1993). 3d statistical neuroanatomical models from 305 mri volumes. In *Nuclear Science Symposium and Medical Imaging Conference, 1993., 1993 IEEE Conference Record.*, pages 1813–1817. IEEE.

Fahrmeir, L. and Kneib, T. (2011). *Bayesian smoothing and regression for longitudinal, spatial and event history data.* Oxford University Press.

Fahrmeir, L., Kneib, T., and Lang, S. (2004). Penalized structured additive regression for space-time data: a Bayesian perspective. *Statistica Sinica*, 14(3):731–762.

Fahrmeir, L. and Tutz, G. (2001). *Multivariate Statistical Modelling Based on Generalized Linear Models.* Springer, New York, 2nd edition.

Ferrari, S. and Cribari-Neto, F. (2004). Beta regression for modelling rates and proportions. *Journal of Applied Statistics*, 31(7):799–815.

Fischer, B. and Freund, R. W. (1994). On adaptive weighted polynomial preconditioning for Hermitian positive definite matrices. *SIAM Journal on Scientific Computing*, 15(2):408–426.

Fletcher, R. (1976). Conjugate gradient methods for indefinite systems. In *Numerical analysis*, pages 73–89. Springer.

Freifeld, O., Greenspan, H., and Goldberger, J. (2009). Multiple sclerosis lesion detection using constrained GMM and curve evolution. *Journal of Biomedical Imaging*, 2009:14.

Friston, K. and Penny, W. (2003). Posterior probability maps and SPMs. *Neuroimage*, 19(3):1240–1249.

Friston, K. J., Holmes, A. P., Worsley, K. J., Poline, J.-P., Frith, C. D., and Frackowiak, R. S. (1994). Statistical parametric maps in functional imaging: a general linear approach. *Human brain mapping*, 2(4):189–210.

Gaillard, P. and Goude, Y. (2015). Forecasting electricity consumption by aggregating experts; how to design a good set of experts. In *Modeling and Stochastic Learning for Forecasting in High Dimensions*, pages 95–115. Springer.

Gamerman, D. (1997). Sampling from the posterior distribution in generalized linear mixed models. *Statistics and Computing*, 7(1):57–68.

Ge, T., Müller-Lenke, N., Bendfeldt, K., Nichols, T. E., and Johnson, T. D. (2014). Analysis of multiple sclerosis lesions via spatially varying coefficients. *The annals of applied statistics*, 8(2):1095–1118.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, 1(3):515–533.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2014). *Bayesian data analysis*, volume 2. Taylor & Francis.

Gelman, A. and Hill, J. (2007). *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical science*, pages 457–472.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741.

George, A. (1973). Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis*, 10(2):345–363.

George, A. and Liu, J. W. (1989). The evolution of the minimum degree ordering algorithm. *SIAM Review*, 31(1):1–19.

Geremia, E., Clatz, O., Menze, B. H., Konukoglu, E., Criminisi, A., and Ayache, N. (2011). Spatial decision forests for MS lesion segmentation in multi-channel magnetic resonance images. *NeuroImage*, 57(2):378–390.

Ghosh, A., Pal, N. R., and Pal, S. K. (1991). Image segmentation using a neural network. *Biological Cybernetics*, 66(2):151–158.

Gilbert, J. R., Moler, C., and Schreiber, R. (1992). Sparse matrices in MATLAB: design and implementation. *SIAM Journal on Matrix Analysis and Applications*, 13(1):333–356.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (1996). Introducing Markov Chain Monte Carlo. In Gilks, W. R., Richardson, S., and Spiegelhalter, D., editors, *Markov Chain Monte Carlo in practice*. Chapman & Hall.

Gilks, W. R. and Roberts, G. O. (1996). Strategies for improving MCMC. In Gilks, W. R., Richardson, S., and Spiegelhalter, D., editors, *Markov Chain Monte Carlo in practice*. Chapman & Hall.

Giraud, L. and Gratton, S. (2006). On the sensitivity of some spectral preconditioners. *SIAM journal on matrix analysis and applications*, 27(4):1089–1105.

Gonzalez, J., Low, Y., Gretton, A., and Guestrin, C. (2011). Parallel Gibbs sampling: From colored fields to thin junction trees. In *International Conference on Artificial Intelligence and Statistics*, pages 324–332.

Good, C. D., Johnsrude, I., Ashburner, J., Henson, R. N., Friston, K. J., and Frackowiak, R. S. (2001). Cerebral asymmetry and the effects of sex and handedness on brain structure: a voxel-based morphometric analysis of 465 normal adult human brains. *Neuroimage*, 14(3):685–700.

Gössl, C., Auer, D. P., and Fahrmeir, L. (2000). Dynamic models in fMRI. *Magnetic Resonance in Medicine*, 43(1):72–81.

Gössl, C., Auer, D. P., and Fahrmeir, L. (2001). Bayesian spatio-temporal inference in functional magnetic resonance imaging. *Biometrics*, 57:554–562.

Green, P. J. (1987). Penalized likelihood for general semi-parametric regression models. *International Statistical Review/Revue Internationale de Statistique*, pages 245–259.

Hafez, M. M., Ōshima, K., and Kwak, D. (2010). *Computational fluid dynamics review 2010*. World Scientific.

Hammersley, J. and Clifford, P. (1971). Markov fields on finite graphs and lattices. Unpublished.

Han, I., Malioutov, D., and Shin, J. (2015). Large-scale log-determinant computation through stochastic Chebyshev expansions. In Blei, D. and Bach, F., editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 908–917. JMLR Workshop and Conference Proceedings.

Harville, D. A. (1974). Bayesian inference for variance components using only error contrasts. *Biometrika*, 61(2):383–385.

Harville, D. A. (1997). *Matrix Algebra From a Statistician's Perspective.* Springer New York.

Hastie, T. and Tibshirani, R. (1993). Varying-coefficient models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 55(4):757–796.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109.

Helmuth, J. A., Paul, G., and Sbalzarini, I. F. (2010). Beyond co-localization: inferring spatial interactions between sub-cellular structures from microscopy images. *BMC bioinformatics*, 11(1).

Herskovits, E., Bryan, R., and Yang, F. (2008). Automated Bayesian segmentation of microvascular white-matter lesions in the ACCORD-MIND study. *Advances in medical sciences*, 53(2):182.

Hestenes, M. R. and Stiefel, E. (1952). *Methods of conjugate gradients for solving linear systems*, volume 49. NBS.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Holmes, C. C. and Held, L. (2006). Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian Anal.*, 1(1):145–168.

Hutchinson, M. F. (1990). A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 19(2):433–450.

Hutton, C., Draganski, B., Ashburner, J., and Weiskopf, N. (2009). A comparison between voxel-based cortical thickness and voxel-based morphometry in normal aging. *Neuroimage*, 48(2):371–380.

Ilić, M., Turner, I., and Anh, V. (2008). A numerical solution using an adaptively preconditioned Lanczos method for a class of linear systems related with the fractional Poisson equation. *International Journal of Stochastic Analysis*, vol. 2008:26 pages.

Ilić, M., Turner, I. W., and Simpson, D. P. (2010). A restarted Lanczos approximation to functions of a symmetric matrix. *IMA Journal of Numerical Analysis*, 30(4):1044–1061.

Illian, J. B., Sørbye, S. H., and Rue, H. (2012). A toolbox for fitting complex spatial point process models using integrated nested laplace approximation (inla). *The Annals* *of Applied Statistics*, pages 1499–1530.

Johnstone, I. M. and Titterington, D. M. (2009). Statistical challenges of high-dimensional data. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 367(1906):4237–4253.

Khayati, R., Vafadust, M., Towhidkhah, F., and Nabavi, M. (2008). Fully automatic segmentation of multiple sclerosis lesions in brain MR FLAIR images using adaptive mixtures method and Markov random field model. *Computers in biology and medicine*, 38(3):379–390.

Klein, N., Kneib, T., and Lang, S. (2015). Bayesian generalized additive models for location, scale, and shape for zero-inflated and overdispersed count data. *Journal of the American Statistical Association*, 110(509):405–419.

Kneib, T. (2006). *Mixed model based inference in structured additive regression.* Dr. Hut Verlag, Munich.

Knorr-Held, L. (1999). Conditional prior proposals in dynamic models. *Scandinavian Journal of Statistics*, 26(1):129–144.

Knorr-Held, L. (2000). Bayesian modelling of inseparable space-time variation in disease risk. *Statistics in Medicine*, 19(17-18):2555–2567.

Knorr-Held, L. and Rue, H. (2002). On block updating in Markov random field models for disease mapping. *Scandinavian Journal of Statistics*, 29(4):597–614.

Koch-Henriksen, N. and Sørensen, P. S. (2010). The changing demographic pattern of multiple sclerosis epidemiology. *The Lancet Neurology*, 9(5):520–532.

Krige, D. (1951). A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society of South Africa*, 52(6):119–139.

Lachmanovich, E., Shvartsman, D., Malka, Y., Botvin, C., Henis, Y., and Weiss, A. (2003). Co-localization analysis of complex formation among membrane proteins by computerized fluorescence microscopy: application to immunofluorescence co-patching studies. *Journal of microscopy*, 212(2):122–131.

Lang, S. and Brezger, A. (2004). Bayesian P-splines. *Journal of computational and graphical statistics*, 13(1):183–212.

LeSage, J. P. and Pace, R. K. (2009). *Introduction to Spatial Econometrics*. Chapman & Hall/CRC.

Liesen, J. and Strakos, Z. (2012). *Krylov subspace methods: principles and analysis*. Oxford University Press.

Lin, X. and Zhang, D. (1999). Inference in generalized additive mixed modelsby using smoothing splines. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(2):381–400.

Lindgren, F., Rue, H., and Lindström, J. (2011). An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498.

Lindquist, M. A. et al. (2008). The statistical analysis of fMRI data. *Statistical Science*, 23(4):439–464.

Manders, E., Stap, J., Brakenhoff, G., Van Driel, R., and Aten, J. (1992). Dynamics of three-dimensional replication patterns during the S-phase, analysed by double labelling of DNA and confocal microscopy. *Journal of cell science*, 103(3):857–862.

Manders, E., Verbeek, F., and Aten, J. (1993). Measurement of co-localization of objects in dual-colour confocal images. *Journal of microscopy*, 169(3):375–382.

Markaki, Y., Smeets, D., Fiedler, S., Schmid, V. J., Schermelleh, L., Cremer, T., and Cremer, M. (2012). The potential of 3D-FISH and super-resolution structured illumination microscopy for studies of 3D nuclear architecture. *Bioessays*, 34(5):412–426.

Martin, R. J. (1992). Approximations to the determinant term in Gaussian maximum likelihood estimation of some spatial models. *Communications in Statistics - Theory and Methods*, 22(1):189–205.

Martins, T. G., Simpson, D., Lindgren, F., and Rue, H. (2013). Bayesian computing with INLA: new features. *Computational Statistics & Data Analysis*, 67:68–83.

Mason, J. C. and Handscomb, D. C. (2002). *Chebyshev polynomials*. CRC Press.

McDonald, W. I., Compston, A., Edan, G., Goodkin, D., Hartung, H.-P., Lublin, F. D., McFarland, H. F., Paty, D. W., Polman, C. H., Reingold, S. C., et al. (2001). Recommended diagnostic criteria for multiple sclerosis: guidelines from the international panel on the diagnosis of multiple sclerosis. *Annals of neurology*, 50(1):121–127.

Meijerink, J. A. and van der Vorst, H. A. (1977). An iterative solution method for linear systems of which the coefficient matrix is a symmetric m-matrix. *Mathematics of computation*, 31(137):148–162.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.

Meulman, N., Wieling, M., Sprenger, S. A., Stowe, L. A., and Schmid, M. S. (2015). Age effects in L2 grammar processing as revealed by ERPs and how (not) to study them. *PLoS ONE*, 10(12):1–27.

Minka, T. P. (2001). Expectation propagation for approximate Bayesian inference. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, UAI '01, pages 362–369, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Monteiro, R. D., O'Neal, J. W., and Nemirovski, A. (2004). A new conjugate gradient algorithm incorporating adaptive ellipsoid preconditioning. *Report, School of ISyE, Georgia Tech, USA.*

Moore, G. E. et al. (1975). Progress in digital integrated electronics. In *Electron Devices Meeting*, volume 21, pages 11–13.

Mortazavi, D., Kouzani, A. Z., and Soltanian-Zadeh, H. (2012). Segmentation of multiple sclerosis lesions in MR images: a review. *Neuroradiology*, 54(4):299–320.

Nelder, J. A. and Wedderburn, R. W. M. (1972). Generalized linear models. *Journal of the Royal Statistical Society. Series A*, 135:370 – 384.

Ohkura, K., Nishizawa, H., Obi, T., Hasegawa, A., Yamaguchi, M., and Ohyama, N. (2000). Unsupervised image segmentation using hierarchical clustering. *Optical Review*, 7(3):193–198.

O'Leary, D. P. (1987). Parallel implementation of the block conjugate gradient algorithm. *Parallel Computing*, 5(1):127–139.

O'Leary, D. P. (1991). Yet another polynomial preconditioner for the conjugate gradient algorithm. *Linear algebra and its applications*, 154:377–388.

Pace, R. K. and LeSage, J. P. (2004). Chebyshev approximation of log-determinants of spatial weight matrices. *Computational Statistics & Data Analysis*, 45(2):179–196.

Paige, C. C. and Saunders, M. A. (1975). Solution of sparse indefinite systems of linear equations. *SIAM journal on numerical analysis*, 12(4):617–629.

Papaspiliopoulos, O. and Roberts, G. O. (2003). Non-centered parameterisations for hierarchical models and data augmentation. *Bayesian Statistics*, 7:307 – 326.

Patterson, H. D. and Thompson, R. (1971). Recovery of inter-block information when block sizes are unequal. *Biometrika*, 58(3):545–554.

Penny, W. D., Trujillo-Barreto, N. J., and Friston, K. J. (2005). Bayesian fMRI time series analysis with spatial priors. *NeuroImage*, 24(2):350–362.

Pinheiro, J. and Bates, D. (2000). *Mixed-effects models in S and S-PLUS.* Springer New York.

Polman, C. H., Reingold, S. C., Banwell, B., Clanet, M., Cohen, J. A., Filippi, M., Fujihara, K., Havrdova, E., Hutchinson, M., Kappos, L., et al. (2011). Diagnostic criteria for multiple sclerosis: 2010 revisions to the McDonald criteria. *Annals of neurology*, 69(2):292–302.

R Core Team (2016). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria.

Reusken, A. (2001). Approximation of the determinant of large sparse symmetric positive definite matrices. *SIAM J. Matrix Anal. Appl.*, 23(3):799–818.

Rigby, R. A. and Stasinopoulos, D. M. (2005). Generalized additive models for location, scale and shape. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 54(3):507–554.

Robert, C. and Casella, G. (2011). A short history of markov chain monte carlo: Subjective recollections from incomplete data. *Statist. Sci.*, 26(1):102–115.

Rue, H. (2001). Fast sampling of Gaussian Markov random fields. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):325–338.

Rue, H. and Held, L. (2005). *Gaussian Markov random fields: theory and applications.* CRC Press.

Rue, H. and Martino, S. (2007). Approximate Bayesian inference for hierarchical Gaussian Markov random fields models. *Journal of Statistical Planning and Inference*, 137:3177–3192.

Rue, H., Martino, S., and Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the royal statistical society: Series b (statistical methodology)*, 71(2):319–392.

Rzezak, P., Squarzoni, P., Duran, F. L., de Toledo Ferraz Alves, T., Tamashiro-Duran, J., Bottino, C. M., Ribeiz, S., Lotufo, P. A., Menezes, P. R., Scazufca, M., and Busatto, G. F. (2015). Relationship between brain age-related reduction in gray matter and educational attainment. *PloS one*, 10(10):e0140945.

Saad, Y. (1994). ILUT: A dual threshold incomplete LU factorization. *Numerical linear algebra with applications*, 1(4):387–402.

Saad, Y. (1995). Preconditioned Krylov subspace methods for CFD applications. In Habashi, W., editor, *Solution Techniques for Large-Scale CFD Problems*, pages 139–158. Wiley, New York.

Saad, Y. (2003). *Iterative methods for sparse linear systems*. Siam.

Saad, Y. (2011). *Numerical Methods for Large Eigenvalue Problems*. Society for Industrial and Applied Mathematics.

Saad, Y. and Schultz, M. H. (1986). Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869.

Schmid, V. J. (2004). *Bayesianische Raum-Zeit-Modellierung in der Epidemiologie*. Dr. Hut Verlag, Munich.

Schmidt, P., Gaser, C., Arsic, M., Buck, D., Förschler, A., Berthele, A., Hoshi, M., Ilg, R., Schmid, V. J., Zimmer, C., Hemmer, B., and Mühlau, M. (2012). An automated tool for detection of FLAIR-hyperintense white-matter lesions in multiple sclerosis. *Neuroimage*, 59(4):3774–3783.

Schmidt, P., Mühlau, M., and Schmid, V. (2017). Fitting large-scale structured additive regression models using Krylov subspace methods. *Computational Statistics & Data Analysis*, 105:59 – 75.

Schmidt, P., Schmid, V. J., Gaser, C., Buck, D., Bührlen, S., Förschler, A., and Mühlau, M. (2013). Fully Bayesian inference for structural MRI: application to segmentation and statistical analysis of T2-hypointensities. *PloS one*, 8(7):e68196.

Schulz, G. (1933). Iterative Berechnung der reziproken Matrix. *Zeitschrift für Angewandte Mathematik und Mechanik*, 13:57 – 59.

Simmons, J. P., Nelson, L. D., and Simonsohn, U. (2011). False-positive psychology undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological science*, pages 1359–1366.

Simpson, D. P., Turner, I. W., and Pettitt, A. (2007). Fast sampling from a Gaussian Markov random field using Krylov subspace approaches. Technical report, Queensland University of Technology.

Simpson, D. P., Turner, I. W., Strickland, C. M., and Pettitt, A. N. (2013). Scalable iterative methods for sampling from massive Gaussian random vectors. *arXiv preprint arXiv:1312.1476*.

Sonneveld, P. (1989). CGS, a fast Lanczos-type solver for nonsymmetric linear systems. *SIAM journal on scientific and statistical computing*, 10(1):36–52.

Thron, C., Dong, S. J., Liu, K. F., and Ying, H. P. (1998). Padé-$Z_2$ estimator of determinants. *Phys. Rev. D*, 57:1642–1653.

Thron, C., Liu, K., and Dong, S. (1996). The PZ method for estimating determinant ratios, with applications. *Nuclear Physics. B, Proceedings Supplements*, pages 977 – 979.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

Tierney, L. and Kadane, J. B. (1986). Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, 81(393):82–86.

Tofts, P. S. (1997). Modeling tracer kinetics in dynamic Gd-DTPA MR imaging. *Journal of Magnetic Resonance Imaging*, 7(1):91–101.

van den Eshof, J. and Hochbruck, M. (2006). Preconditioning Lanczos approximations to the matrix exponential. *SIAM Journal on Scientific Computing*, 27(4):1438–1457.

van Dyk, D. A. and Park, T. (2008). Partially collapsed gibbs samplers: Theory and methods. *Journal of the American Statistical Association*, 103(482):790–796.

van Leemput, K., Maes, F., Vandermeulen, D., Colchester, A., and Suetens, P. (2001). Automated segmentation of multiple sclerosis lesions by model outlier detection. *Medical Imaging, IEEE Transactions on*, 20(8):677–688.

Varga, R. S. (1960). Factorization and normalized iterative methods. In Langer, R. E., editor, *Boundary Problems in Differential Equations*. University of Wisconsin Press.

Wallis, J. W., Miller, T. R., Lerner, C. A., and Kleerup, E. C. (1989). Three-dimensional display in nuclear medicine. *Medical Imaging, IEEE Transactions on*, 8(4):297–230.

Wang, S., Zhu, W., and Liang, Z.-P. (2001). Shape deformation: Svm regression and application to medical image segmentation. In *ICCV 2001. Proceedings of the Eighth IEEE International Conference on Computer Vision.*, volume 2, pages 209–216. IEEE.

Welford, B. (1962). Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3):419–420.

Wells III, W. M., Grimson, W. E. L., Kikinis, R., and Jolesz, F. A. (1996). Adaptive segmentation of MRI data. *Medical Imaging, IEEE Transactions on*, 15(4):429–442.

West, M. and Harrison, J. (1997). *Bayesian Forecasting and Dynamic Models*. Springer, New York, 2nd edition.

Wieling, M., Tomaschek, F., Arnold, D., and Tiede, M. (2015). Investigating dialectal differences using articulography. In *Proceedings of ICPhS 2015*. International Phonetic Association.

Winkler, G. (2003). *Image analysis, random fields and Markov chain Monte Carlo methods: a mathematical introduction*, volume 27. Springer.

Wood, S. N., Goude, Y., and Shaw, S. (2015). Generalized additive models for large data sets. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 64(1):139–155.

World Health Organization (2008). Atlas: Multiple sclerosis resources in the world 2008.

Worsley, K. J. and Friston, K. J. (1995). Analysis of fMRI time-series revisited—again. *Neuroimage*, 2(3):173–181.

Wu, Y., Warfield, S. K., Tan, I. L., Wells, W. M., Meier, D. S., van Schijndel, R. A., Barkhof, F., and Guttmann, C. R. (2006). Automated segmentation of multiple sclerosis lesion subtypes with multichannel MRI. *NeuroImage*, 32(3):1205–1215.

Xing, F. and Yang, L. (2016). Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: A comprehensive review. *IEEE Reviews in Biomedical Engineering*, 9:234–263.

Yu, Y. and Meng, X.-L. (2011). To center or not to center: That is not the question—an ancillarity–sufficiency interweaving strategy (ASIS) for boosting MCMC efficiency. *Journal of Computational and Graphical Statistics*, 20(3):531–570.

Zhang, R., Czado, C., Sigloch, K., et al. (2013). A Bayesian linear model for the high-dimensional inverse problem of seismic tomography. *The Annals of Applied Statistics*, 7(2):1111–1138.

Zhang, Y., Leithead, W., Leith, D., and Walshe, L. (2008). Log-det approximation based on uniformly distributed seeds and its application to Gaussian process regression. *Journal of Computational and Applied Mathematics*, 220(1–2):198 – 214.

Zhang, Y. and Leithead, W. E. (2007). Approximate implementation of the logarithm of the matrix determinant in Gaussian process regression. *journal of Statistical Computation and Simulation*, 77(4):329–348.

# Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12. Juli 2011, § 8 Abs. 2 Pkt. 5)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

Berlin, den 04. November 2016

Paul Schmidt