5-2018

Hierarchical Bayesian Regression with Application in Spatial Modeling and Outlier Detection

Ghadeer Mahdi
University of Arkansas, Fayetteville

Hierarchical Bayesian Regression with Application in Spatial Modeling and Outlier Detection

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Mathematics

by

Ghadeer Mahdi
University of Baghdad
Bachelor of Science in Mathematics, 2004
University of Baghdad
Master of Science in Mathematics, 2007
University of Arkansas
Master of Science in Mathematics, 2015
University of Arkansas
Master of Science in Statistics and Analytics, 2016

May 2018
University of Arkansas

This dissertation is approved for recommendation to the Graduate Council.

Mark Arnold, Ph.D.
Dissertation Director

Avishek Chakraborty, Ph.D.
Dissertation Director

Giovanni Petris, Ph.D.
Committee Member

Qingyang Zhang, Ph.D.
Committee Member

Abstract

This dissertation makes two important contributions to the development of Bayesian hierarchical models. The first contribution is focused on spatial modeling. Spatial data observed on a group of areal units is common in scientific applications. The usual hierarchical approach for modeling this kind of dataset is to introduce a spatial random effect with an autoregressive prior. However, the usual Markov chain Monte Carlo scheme for this hierarchical framework requires the spatial effects to be sampled from their full conditional posteriors one-by-one resulting in poor mixing. More importantly, it makes the model computationally inefficient for datasets with large number of units. In this dissertation, we propose a Bayesian approach that uses the spectral structure of the adjacency to construct a low-rank expansion for modeling spatial dependence. We develop a computationally efficient estimation scheme that adaptively selects the functions most important to capture the variation in response. Through simulation studies, we validate the computational efficiency as well as predictive accuracy of our method. Finally, we present an important real-world application of the proposed methodology on a massive plant abundance dataset from Cape Floristic Region in South Africa. The second contribution of this dissertation is a heavy tailed hierarchical regression to detect outliers. We aim to build a linear model that can allow for small as well as large magnitudes of residuals through observation-specific error distribution. t-distribution is specifically suited for that purpose as we can parametrically control its degrees of freedom (df) to tune the heaviness of its tail - large df values represent observations in normal range and small ones represents potential outliers with high error magnitudes. In a hierarchical structure, we can write t-distribution as a scale mixture of a Gaussian distribution so that the standard MCMC algorithm for Gaussian setting can still be used. Post-MCMC, the posterior mean of degrees of freedom for any observation acts as a measure of outlyingness of that observation. We implemented this method on a real dataset consisting of biometric records.

Acknowledgements

I would like to express my sincerest gratitude to my advisors, Dr. Mark Arnold and Dr. Avishek Chakraborty, for their continued help, care and support during my academic studies. I respect and admire them for providing me with insights, feedback and encouragement through the research. They guided me to achieve my goal in writing this dissertation. Beside my advisors, I would like to extend my thanks to my dissertation committee members, Dr. Giovanni Petris and Dr. Qingyang Zhang for the help and support.

My special thanks also goes to the faculty and staff in the Graduate School and Department of Mathematical Sciences at the University of Arkansas. Also, I would like to thank the Higher Committee for Education Development (HCED) for providing me the opportunity to pursue my Ph.D. from the University of Arkansas in US.

I sincerely thank Professors Anthony Rebelo, John Silander, Andrew Latimer and Adam Wilson for allowing me to use the Ecological and Environmental Datasets from Cape Floristic Region for my analysis in Chapter 4. I am grateful to Dr. Judith Weber and Dr. Mallik Rettiganti for giving me the opportunity to work as a research assistant in the NIH-funded project (NIH - 1P20GM109096-01A1) on outlier detection. I also want to thank Dr. Anthony Goudie for sharing the biometric dataset used for analysis in Chapter 5.

I owe big thanks to all my firends and family members who always encourage and support me, especially my wife, Zahraa Al-Sharea.

Table of Contents

1 Introduction to Bayesian Inference 1

1.1 Motivation and Key Concepts . . . 1

1.2 Monte Carlo Methods . . . 2

1.3 Markov Chain Monte Carlo Methods . . . 3

1.3.1 Metropolis Hasting Algorithm . . . 4

1.3.2 Gibbs Sampler . . . 5

1.4 Preview of Following Work . . . 6

2 Spatial Data 8

2.1 Introduction . . . 8

2.2 Modeling Areal-level Spatial Data . . . 11

2.3 Creating of Neighborhood Structure . . . 12

2.4 Empirical Measures of Spatial Association . . . 15

2.5 Prior Distribution for Spatial Effect . . . 16

2.6 Conditionally Auto-Regressive Prior . . . 18

3 Krylov Subspace Methods 20

3.1 Introduction . . . 20

3.2 Krylov Subspace . . . 21

3.2.1 Ritz Approximation . . . 23

3.3 Arnoldi Methods . . . 24

3.4 Lanczos Methods . . . 27

3.5 Arnoldi Algorithm with Restarting . . . 29

3.5.1 Explicit Restart . . . 30

3.5.2 Implicit Restart . . . 31

3.6 Convergence of The Restarted Arnoldi Algorithm 32

3.7 Implicit Restarted Lanczos 34

3.7.1 Shift Selection 35

3.7.2 Implicit Restarted Lanczos with Exact Shifts 37

3.7.3 Implicit Restarted Lanczos with Leja Shifts 38

3.8 Conclusion 40

4 A Computationally Efficient Hierarchical Model for Large Areal Data 41

4.1 Introduction 41

4.2 Method Based on Spectral Structure 43

4.3 Adaptive Selection of Eigenvectors 46

4.4 Simulation Studies 52

4.4.1 Simulation Study I 52

4.4.2 Simulation Study II 53

4.5 Model for Plant Abundance 57

4.6 Discussion 69

5 Hierarchical Regression model for Outlier Detection 71

5.1 Introduction 71

5.2 Regression with Heavy-tailed Error Distribution 72

5.2.1 MCMC Algorithms for Outlier Detection Using t-residual 74

5.3 Simulation Studies 80

5.3.1 Model for Simulation 81

5.3.2 Accuracy of coefficient estimation in presence of outliers 83

5.3.3 Model Application on the Simulated Datasets 85

5.3.4 Comparison Among Methods for Detection of Outliers 90

5.3.5 Criterion for Outlier Determination 92

5.4 Comparison Against Existing Methods 94

5.4.1 Bonferroni Outlier Test 94

5.4.2 Bayesian Test for Outliers Detection 94

5.4.3 Comparison of Simulation Datasets 95

5.5 Application to a Real Dataset 98

5.5.1 Data Description 98

5.5.2 Formulation of the Problem 98

5.5.3 Applying the Methods on the Dataset 99

5.5.4 Comparison of Thresholds for Determining Outliers 100

5.5.5 Exploring Temporal Pattern 101

5.5.6 Analysis of Outlying Observations 102

5.6 Conclusion 105

Bibliography **106**

List of Tables

4.1 Computational time (mm:ss) for IRL-LSFLE and QR method for 100 smallest eigenpairs . . . 52

4.2 IRL-LSFLE computation times (mm:ss) for variation in n, k and sparsity of L 53

4.3 Comparison of predictive and computational performance . . . 55

4.4 Significance of covariate effects on presence-absence of a species . . . 65

5.1 Prior and posterior probabilities distribution for ν_i . . . 77

5.2 Parameter comparison among all methods across the three simulation models 86

5.3 Number of outliers in first 20 and 100 ranked ν values . . . 92

5.4 Sensitivity and specificity comparison for different choices of ν_0 . . . 93

5.5 Number of outliers in first 20 and 100 observations . . . 97

5.6 Number of matches in 100 and 1000 most outlying observations . . . 100

5.7 Values of threshold parameter for obtaining comparable numbers of outliers 100

5.8 Posterior mean and 95% credible intervals for β . . . 101

5.9 Number of matches outliers in 100 and 1000 most outlying observations . . . 102

5.10 Number of matches in 100 and 1000 most outlying observations . . . 102

5.11 The 1000 and 10000 most outlying observations across different gender and grades . . . 105

List of Figures

2.1 PM2.5 moutoring sites in 3 states showing average levels in 2001 (Figure 1.1 in Banerjee et al. (2014)) 9

2.2 Age-adjusted obesity rates by U.S. county (Figure 1 in Thomas (2013)) . . . 10

2.3 Point pattern data showing commerce robbery(A) and passerby robbery(B) at various scales. (Figure 2 in de Melo et al. (2015)) 11

2.4 A Graph with 7 nodes 14

2.5 A 3 × 3 Grid Cell & Adjacency (Figure 5.1 page 144 in Bhark (2011)) 14

4.1 (Left) Cells within the CFR that have at least one observation from the Protea Atlas dataset are shown in light grey, while cells with no observations are shown in dark grey. (Right) Proportion of untransformed land inside the CFR. Most of the transformation is due to agriculture, but includes dense stands of alien invasive species. 59

4.2 PRCYNA: Posterior histogram of number of Laplacian eigenvectors chosen by (top) PA model and (bottom) CA model for different choices of k_{max} parameter 63

4.3 PRREPE: Posterior histogram of number of Laplacian eigenvectors chosen by (top) PA model and (bottom) CA model for different choices of k_{max} parameter 64

4.4 PRPUNC: Posterior histogram of number of Laplacian eigenvectors chosen by (top) PA model and (bottom) CA model for different choices of k_{max} parameter 64

4.5 Protea cynaroides: Spatial maps of marginal posterior abundance probabilities for category 0 (top) to 3 (bottom) with and without accounting for land transformation 66

4.6 Protea repens: Spatial maps of marginal posterior abundance probabilities for category 0 (top) to 3 (bottom) with and without accounting for land transformation 67

4.7 Protea punctata: Spatial maps of marginal posterior abundance probabilities for category 0 (top) to 3 (bottom) with and without accounting for land transformation 68

5.1 Datasets with 20 outliers: the true outliers are marked with red stars, and the blue line represents the trend of the data points. 82

5.2 Datasets with 100 outliers: the true outliers are marked with red stars, and the blue line represents the trend of the data points. 83

5.3 Outlier detection using posterior sample mean for ν for Method-I, IIa and IIb on SI (Top) with 20 outliers and (Bottom) with 100 outliers: The actual outliers are marked as red stars. 87

5.4 Outlier detection using posterior sample mean for ν for Method-I, IIa and IIb on SIIa (Top) with 20 outliers and (Bottom) with 100 outliers: The actual outliers are marked as red stars. 88

5.5 Outlier detection using posterior sample mean for ν for Method-I, IIa and IIb on SIIb (Top) with 20 outliers and (Bottom) with 100 outliers: The actual outliers are marked as red stars. 89

5.6 The number of observations that should be considered to detect different proportions of the actual outliers on datasets with 20 outliers 90

5.7 The number of observations that should be considered to detect different proportions of the actual outliers on datasets with 100 outliers 91

5.8 The number of observations that should be considered to detect different proportions of the actual outliers on datasets with 20 outliers 96

5.9 The number of observations that should be considered to detect different proportion of the actual outliers on datasets with 100 outliers 97

5.10 The time series of Z-scores for 3 students (The numbers indicate the rank of changes w.r.t. all outliers) 103

5.11 Z-scores from one student with unusually large values between all grades (The numbers indicate the rank of changes w.r.t. all outliers) 104

Chapter 1

Introduction to Bayesian Inference

1.1 Motivation and Key Concepts

In probability theory, we can use prior knowledge of conditions to describe the probability of an event. This is called Bayes' rule (or Bayes' theorem which is named after Reverend Thomas Bayes, 1701–1761. Mathematically, we can state the theorem as follows,

$$\pi(\theta|D) = \frac{L(D|\theta)\pi(\theta)}{\pi(D)}$$

where D is the data, θ is the parameter, $L(D|\theta)$ is the likelihood of D given θ, $\pi(\theta)$ is the probability distribution of the parameter before observing the data, and $\pi(D)$ is the marginal distribution of the data. The goal from the Bayesian inference is to learn about parameters given the dataset.

The probability distribution of any parameter before observing the data is called the prior distribution, and the probability distribution of our data given parameters is called Likelihood. Before we collect the data, prior distribution of any parameter gives us an idea about its possible values. We need to use both the prior and the likelihood to learn about parameters θ.

Suppose, we have a data $D = \{y_1, y_2, \ldots, y_n\}$ such that $y_i \overset{iid}{\sim} f(\cdot|\theta)$, then the likelihood function becomes,

$$L(D|\theta) = \prod_{i=1}^{n} f(y_i|\theta)$$

A probability distribution needs to be specified for θ, $\theta \sim \pi(\theta)$. Our goal is to estimate the posterior distribution of θ given the data.

In general, the posterior distribution of $\theta|D$ can be written as a product of the Likelihood of the data given parameters and prior distribution of these parameters, i.e.,

$$\pi(\theta|D) \quad \propto \quad L(D|\theta) \times \pi(\theta) = \prod_{i=1}^{n} f(y_i|\theta) \times \pi(\theta)$$

Sometimes, we can have parameters inside these priors, and they are called hyper-parameters, and their distribution are called hyper-priors.

By analyzing the posterior density function, $\pi(\theta|D)$, we obtain various information about θ. For example, we can compute different properties of θ such as, Mean $= \int \theta\pi(\theta|D)d\theta$, and Median $= M$, where $\int_{\theta \leq M} \pi(\theta|D)d\theta = 0.5$. Also, for any region R, one can compute $\pi(\theta \in R|D)$. The $(1-\alpha)$ credible intervals for a parameter $\{\theta : \theta \in R\}$ can be evaluated by using the following, $\int_R \pi(\theta|D)d\theta = 1 - \alpha$. However, whenever θ is a vector, many integrals become analytically difficult to solve, and hence an alternative numerical method can be used, such as the Monte Carlo method.

1.2 Monte Carlo Methods

Monte Carlo (MC) is commonly used in Mathematical and Physical problems (Del Moral, 2013; Robert, 2004). It can be utilized to obtain numerical results for problems involving numerical integration, function optimization, and characteristics of a probability distribution. For example, by using the Strong Law of Large Number (SLLN) theorem one can approximate the mean by computing the average from large samples.

SLLN: *If $Y_1, Y_2, \ldots, Y_n$ are iid random variables with $E(Y_j) = \mu$, then $\frac{1}{n}\sum_{i=1}^{n} Y_i \to \mu$, almost surely as $n \to \infty$.*

For example, if θ is a vector, *e.g.* $\theta = (\theta_1, \theta_2, \ldots, \theta_n)$, then for given data points, our goal is to sample from $\pi(\theta|D)$. If $\pi(\theta|D)$ is a standard distribution, then we can draw directly from it. However, most of the time, it is not possible to draw from the joint distribution $\pi(\theta|D)$ directly, specifically when the hierarchical model is complex and involves many parameters; in those cases we can still use the MC method through a Markov chain.

1.3 Markov Chain Monte Carlo Methods

The Markov Chain Monte Carlo (MCMC) methods are the most popular techniques in Bayesian estimation (Gilks, 2005). Estimation using MCMC methods are often applied to solve numerical approximations of multi-dimensional integrals and optimization problems in spaces with large dimension. For example, it has been used in computational Physics (Brubaker et al., 2012), computational Biology(Bouckaert et al., 2014), machine learning (Andrieu et al., 2003) and Economics(Lin and Huang, 2002).

In Statistics, MCMC techniques are used to sample from a probability distribution when the direct sampling is difficult. In MCMC, we can simulate an observation from a Markov distribution conditional from previous draws. If the Markov distribution is properly constructed and the chain is run a large number of times, then the draws approximately follow the target distribution.The draws can be used to approximate the target joint or marginal distribution, or to compute an integral with respect to that distribution.

The large hierarchical models that involve several parameters can be estimated by the MCMC algorithms. However, we need to (i) throw away the initial few draws to eliminate the effect of the initial values, and (ii) thin the remaining draws at a certain interval to remove correlations between successive draws. By increasing the thinning width, correlations can be reduced further. To get a reasonable estimate of any desired distribution, we need a large number of draws.

In the last few decades, several kinds of MCMC algorithms have been developed. For example: Metropolis-Hasting algorithm (Metropolis et al., 1953; Hastings, 1970), Gibbs sampler (Geman and Geman, 1984), Monte Carlo EM (Baum et al., 1970), Slice sampler

(Neal, 2003), and Reversible jump MCMC (Andrieu et al., 2003).

There are several ways to check whether the Markov chain converges. For more information about convergence, see (Zhu et al., 2003; Banerjee et al., 2014).

In the rest of this chapter, we are going to discuss the most important MCMC algorithms that we are going to use in this work.

1.3.1 Metropolis Hasting Algorithm

In general, the Metropolis Hasting (MH) algorithm can be used to sample from any target distribution. At iteration i, with a target density $p(\theta|D)$, we generate a candidate $\theta^{proposed}$ for the next sample by drawing from a proposal distribution $q(\theta^{proposed}|\theta^{(i)})$. Then we calculate the acceptance ratio, R, where

$$R = R(\theta^{(i)} \to \theta^{proposed}) = \min(1, \frac{p(\theta^{proposed})q(\theta^{(i)}|\theta^{proposed}))}{p(\theta^{(i)})q(\theta^{proposed}|\theta^{(i)}))})$$

Then we set $\theta^{(i+1)} = \theta^{proposed}$ if $u \leq R$; Otherwise, $\theta^{(i+1)} = \theta^{i}$, where u is an uniform random number on (0, 1).

In Algorithm 1, we summarize the MH algorithm:

Algorithm 1 MH

```
INPUT: An initial value $\theta^{(i)} = (\theta_1^{(i)}, \theta_2^{(i)} \dots, \theta_n^{(i)})$ and proposal density $q(x)$
OUTPUT: A new sample $\theta^{(i+1)} = (\theta_1^{(i+1)}, \theta_2^{(i+1)} \dots, \theta_n^{(i+1)})$
assume $\theta^{proposed} \sim q(\theta^{proposed}|\theta)$
compute the acceptance ratio (R) where
$R = R(\theta^{(i)} \to \theta^{proposed}) = \min(1, \frac{p(\theta^{(proposed)})q(\theta^{(i)}|\theta^{proposed})}{p(\theta^{(i)})q(\theta^{proposed}|\theta^{(i)})})$
generate $u$ from unif(0,1)
if $u \leq R$, then
    $\theta^{(i+1)} = \theta^{proposed}$
else
    $\theta^{(i+1)} = \theta^{(i)}$
end if
```

In the MH algorithm, it is not easy to find the right proposal distribution, so that we have a reasonable acceptance rate. If we take the proposal distribution to have a very small variance, then we are accepting a lot of samples, but we are not exploring the parameter space adequately. On the other hand, if we choose a large value of variance, we move very quickly, but we are accepting very few samples. Hence, it is important to have an acceptance rate which is neither too large nor too small, maybe in the range 20–40%. This problem gets bigger when we have a high dimensional parameter. In those cases, a different approach, the Gibbs sampler, can be used.

1.3.2 Gibbs Sampler

In multivariate posterior distribution, the Gibbs sampler is useful method because, rather than choosing a sample for the entire parameter vector at once, a new sample for each dimension (or for a block of parameters) will be drawn conditionally on the current state of other parameters. For us to be able to draw exactly, the Gibbs sampler requires such conditional distributions to be in standard form. For $\theta = (\theta_1, \theta_2, \ldots, \theta_n)$, our target is the full conditional distribution $p(\theta_j|\theta_{-j})$ where $\theta_{-j} = \theta_{1:n\backslash\{j\}}$. We can view this as a special case of the MH algorithm, where for $j = 1, 2, \ldots, n$, we use the following proposed distribution,

$$q(\theta_j^{proposed}|\theta_j^{(i)}) = p(\theta_j^{proposed}|\theta_{-j}^{(i)})$$

Therefore, the corresponding acceptance probability is 1 for each proposal, as we can see below,

$$\begin{aligned}
q(\theta_j^{proposed}|\theta_j^{(i)}) &= \min\left(1, \frac{p(\theta_j^{proposed})q(\theta_j^{(i)}|\theta^{proposed})}{p(\theta_j^{(i)})q(\theta_j^{proposed}|\theta^{(i)})}\right) \\
&= \min\left(1, \frac{p(\theta^{proposed})p(\theta_j^{(i)}|\theta_{-j}^{(i)})}{p(\theta^{(i)})p(\theta_j^{proposed}|\theta_{-j}^{proposed})}\right) \\
&= \min\left(1, \frac{p(\theta_j^{proposed})}{p(\theta_j^{(i)})}\right) = 1
\end{aligned}$$

The following algorithm, Algorithm 2, summarizes the Gibbs sampler method.

Algorithm 2 Gibbs Sampler

```
INPUT: An initial value $\theta^{(i)} = (\theta_2^{(i)}, \theta_3^{(i)}, \ldots, \theta_n^{(i)})$
OUTPUT: A new sample $\theta^{(i+1)} = (\theta_1^{(i+1)}, \theta_2^{(i+1)} \ldots, \theta_n^{(i+1)})$
for $k = 1, 2, \ldots, N$ do
    draw $\theta_1^{(i+1)}$ from $p(\theta_1 | \theta_2^{(i)}, \theta_3^{(i)}, \ldots, \theta_n^{(i)})$
    draw $\theta_2^{(i+1)}$ from $p(\theta_2 | \theta_1^{(i+1)}, \theta_3^{(i)}, \ldots, \theta_n^{(i)})$
    $\vdots$
    draw $\theta_n^{(i+1)}$ from $p(\theta_n | \theta_1^{(i+1)}, \theta_2^{(i+1)}, \ldots, \theta_{n-1}^{(i+1)})$
end for
```

1.4 Preview of Following Work

We are going to use the Bayesian methods in context of two different problems: one in spatial modeling and the other in outlier detection. Our key contribution for this dissertation is detailed in the work that is presented in Chapter 4 and Chapter 5. Here, we give summary of the work in the next four chapters.

In chapter 2, we discuss the basics of spatial data with several examples. Then, we elaborate on the modeling details of the Areal-level spatial datasets as our key contribution is focused on that kind of data.

In chapter 3, we describe the Krylov subspace methods and some general subspace iteration algorithms. We include a brief discussion of basic Arnoldi and Lanczos algorithms, and some developed Arnoldi algorithms.

In chapter 4, we derived a hierarchical model for species using effecient dimension reduction for lattice data. We propose an alternative approach using the spectral properties of the adjacency matrix. We applied our work on the species abundance datasets from Cape Floristic Region in South Africa.

In chapter 5, we developed two outlier detection methods by using the hierarchical regression model based on heavy tailed error distributions. We test our methods, and compared them to two existent methods by using some simulation datasets. The real data implementations of our methods consists of outliers detection in the records of heights of Arkansas school students.

We want to introduce some of the notations used throughout this work. For an event A, $1[A]$ denotes a random variable which takes the value 1 if A occurs and 0 if A does not occur. For a real number a, δ_a denotes a probability distribution that puts entire mass at a. MVN and IG are the notations for multivariate normal distribution and inverse-gamma distributions, respectively. $\Phi(A; \mu, \sigma^2)$ denotes the probability enclosed in set A under univariate normal distribution with mean μ and variance σ^2. The notations $0_l, 1_l$ and I_l refer to the $l \times 1$ vector of all zeros, all ones and the identity matrix of order $l \times l$, respectively. All computations conducted for this chapter are run in R (https://cran.r-project.org/) on a single processor without using any explicit distributed computing.

Chapter 2

Spatial Data

2.1 Introduction

Spatial data includes information that identifies the geographic location of features and boundaries on Earth, such as natural or constructed features, oceans, public health, school districts, etc (Banerjee et al., 2014). It usually can be mapped and stored as coordinates and topology. Also, when the information about a physical object can be represented by numerical values in a geographic coordinate system, it is called spatial or geospatial data. The use of spatial data analysis has risen in many different scientific fields such as: Public Health (Nucci et al., 2016), Biological Sciences (Weston et al., 2012), and Geological Science (Rampaso et al., 2016).

The spatial data can be classified into three types:

- Point-referenced data: If the dataset consists of a vector of measurements $y(s_1), y(s_2)$, $\ldots, y(s_2)$ at locations $s_1, s_2, \ldots, s_n \in D \subseteq \mathbb{R}^n$, then we call it point reference data. For example, Figure 2.1 shows the 2001 PM2.5 level at 114 locations in Illinois, Indiana, and Ohio (Banerjee et al., 2014). The reader can consult Stein et al. (2004) for more information about Point-level models which are related to this kind of data.

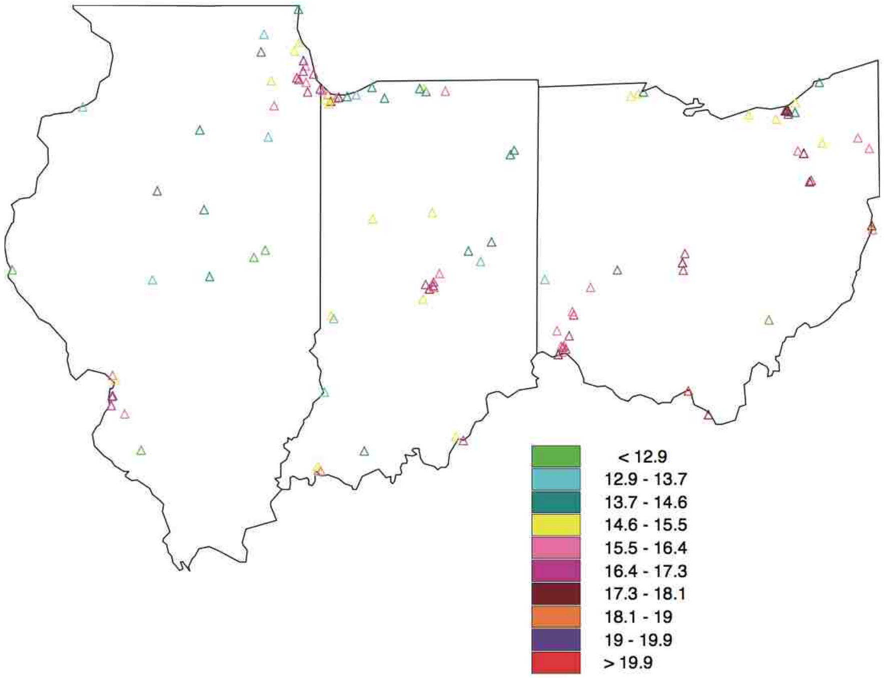


Figure 2.1: PM2.5 moutoring sites in 3 states showing average levels in 2001 (Figure 1.1 in Banerjee et al. (2014))

- Areal data: This type of dataset arises when a fixed region D is partitioned into multiple regular/irregular units and each unit corresponds to one measurement in the dataset. Figure 2.2 shows the 2008 age-adjusted obesity rates by U.S. county. The percentage of people have been indicated by different colors in each area according to the percentage of the population with a Body Mass Index (BMI) greater than or equal to 30.

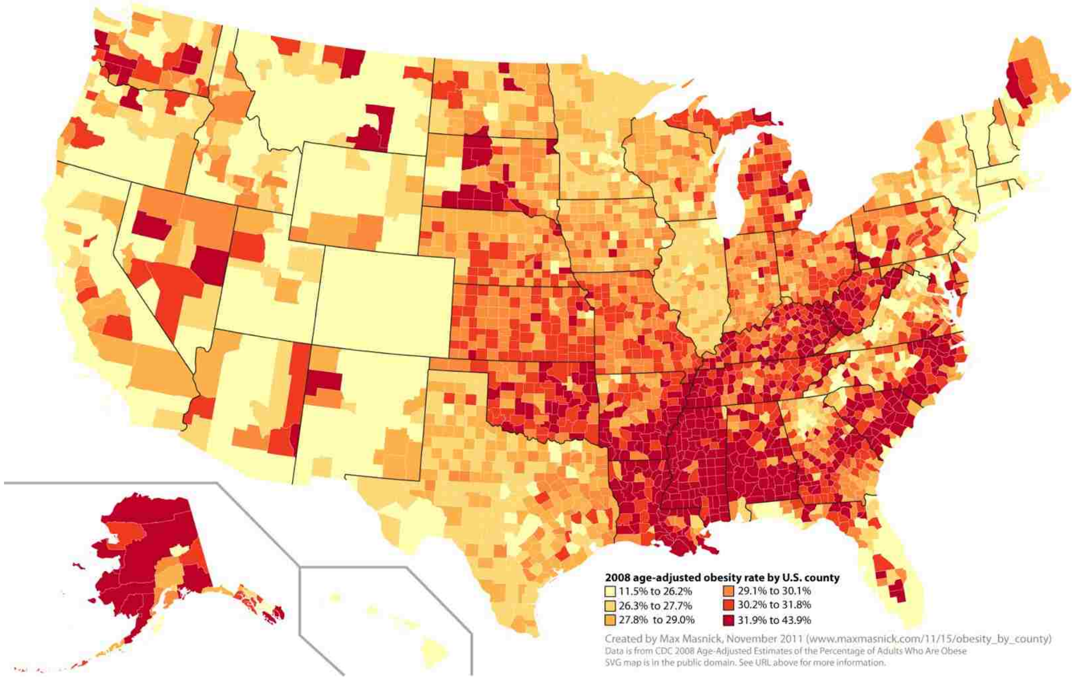

2008 age-adjusted obesity rate by U.S. county
11.5% to 26.2%
26.3% to 27.7%
27.8% to 29.0%
29.1% to 30.1%
30.2% to 31.8%
31.9% to 43.9%
Created by Max Masnick, November 2011 (www.maxmasnick.com/11/15/obesity_by_county)
Data is from CDC 2008 Age-Adjusted Estimates of the Percentage of Adults Who Are Obese
SVG map is in the public domain. See URL above for more information.

Figure 2.2: Age-adjusted obesity rates by U.S. county (Figure 1 in Thomas (2013))

- Point pattern data: Unlike the previous two type of spatial data here the locations are random, so the dataset consists of location of a particular event of interest with in a fixed region D. Residences of persons suffering from a particular disease or the locations of a certain species of tree in a forest are examples of point pattern data. As an example, Figure 2.3 taken from de Melo et al. (2015) shows the southeast region of Brazil, where the Campinas is located. Part(A) shows commerce robbery, and part(B) shows passerby robbery. The similarity appears in the spatial patterns in both parts even though commerce robbery has a lower amount of offenses than passerby robbery. More information about Point pattern data and related model is found in Lawson and Denison (2002) and Moller and Waagepetersen (2003).

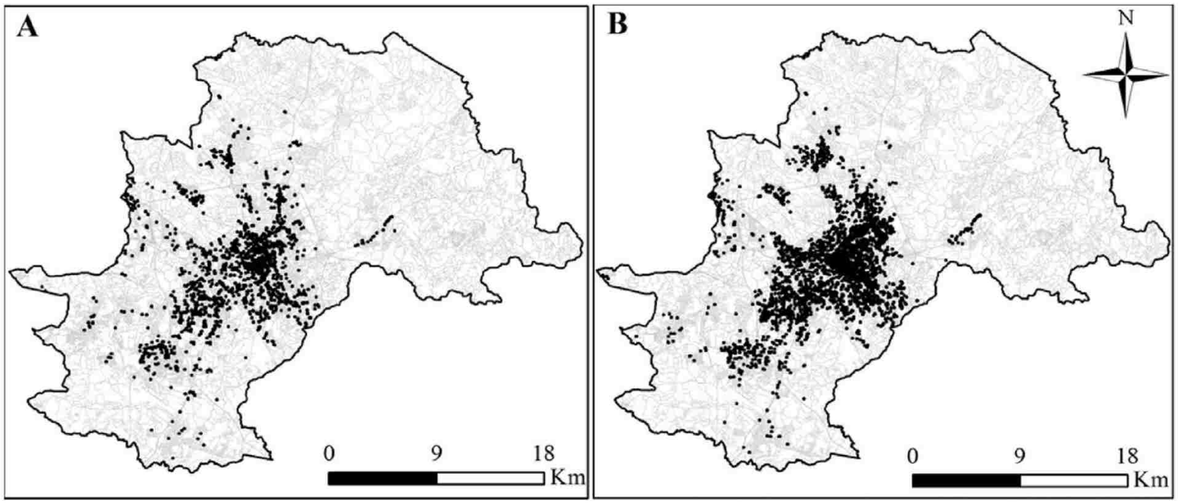


Figure 2.3: Point pattern data showing commerce robbery(A) and passerby robbery(B) at various scales. (Figure 2 in de Melo et al. (2015))

2.2 Modeling Areal-level Spatial Data

In this dissertation, we specifically focus on Areal-level spatial data. In a spatial setting, we deal with response and covariate variables that are connected to regions. Let us denote the region as D, so whenever we have n units, we call them $A_1, A_2, \ldots, A_n$. Therefore, we can write the Areal-level spatial model as

$$y(A_i) = \beta_0 + \beta_1 x_1(A_i) + \beta_2 x_2(A_i) + \cdots + \beta_p x_p(A_i) + \epsilon(A_i) \tag{2.1}$$

where $y(A_i)$ is the response, $x_1(A_i), x_2(A_i), \ldots, x_p(A_i)$ are the covariates, and $\epsilon(A_i)$ is zero mean Gaussian noise from the i^{th} unit with variance σ^2. Eq. 2.1 is called a multiple linear regression. Now, consider the situation where $x_1, x_2, \ldots, x_p$ are poor predictors in explaining y, (i.e., in the above multivariate regression model, R^2 is too low). Some possible reasons for this are

- Response and covariates do not a have strong linear relation.
- There may be additional covariates on which we have no data.

- There may be similarity between measurements from adjacent units that can not be explained by covariates.

The model above is not sufficient to explain how the response, y, changes from one region to another. Hence, we need a stronger model that can explain y better. We can define a new model as follows,

$$y(A_i) = \beta_0 + \beta_1 x_1(A_i) + \beta_2 x_2(A_i) + \cdots + \beta_p x_p(A_i) + \theta(A_i) + \epsilon(A_i) \tag{2.2}$$

where $\theta(A_i)$ is called a spatial effect. When we have n areal units $(A_1, A_2, \ldots, A_n)$, we will have n spatial effects $(\theta(A_1), \theta(A_2), \ldots, \theta(A_n))$. Eq. 2.2 is called Areal-level spatial model.

2.3 Creating of Neighborhood Structure

For any two regions which are adjacent, *e.g.* A_i and A_j, we have the following two models:

$$\begin{aligned} y(A_i) &= \beta_0 + \beta_1 x_1(A_i) + \beta_2 x_2(A_i) + \cdots + \beta_p x_p(A_i) + \theta(A_i) + \epsilon(A_i) \\ y(A_j) &= \beta_0 + \beta_1 x_1(A_j) + \beta_2 x_2(A_j) + \cdots + \beta_p x_p(A_j) + \theta(A_j) + \epsilon(A_j) \end{aligned}$$

If the data has a strong spatial pattern, we expect $y(A_i)$ and $y(A_j)$ have close or similar values if A_i and A_j are adjacent. If this is the case, we say that θ has a strong spatial pattern. But if y does not have a strong spatial pattern, we expect θ to behave like random error, ϵ. Hence θ makes sense only when there is a spatial pattern. We want to build a model for $\theta(A_i)$. First, we have to define the relation among units by a matrix which is called an adjacency matrix, we denote it with W, where for n units, $W = [w_{ij}]$, $i, j = 1, \ldots, n$, where $w_{ij} = 1$ if A_i and A_j are neighboring and 0 otherwise. We can explain the relation between two units, by specifying that there is an edge between them if $w_{ij} = 1$. Some of the common approaches to identify the edges are

- Sharing a boundary: we say two units, *e.g.* A_i and A_j, are neighbors, if they share a boundary. This is called first order neighbor. It is called second order neighbor if A_i and A_j share boundaries or they have a common unit, *e.g.* A_k, that shares a boundary with both of them. In the second order neighboring case, the adjacency matrix will be less sparse.

- Distance between unit centers: in some situations, geographical adjacency may not be appropriate, so the distance between unit centers is a better measurement. Two units A_i and A_j have an edge between them if the distance between centers of A_i and A_j is less than a certain distance, *e.g.* d, which is the user choice.

- Minimum distance between two units: neighbors can be defined by working with the minimum distance between two units, $d_{ij} = \min\{d(x, y) : x \in A_i \text{ and } y \in A_j\}$. We say two units are neighbors if the distance between them is less than d.

We can view the neighborhood relationship as edges in a graph. If we consider the case where W has all zero entries, the graph has no edge and no connection adjacent $\theta(A_i)$. In this case, there is no point of using the spatial effect, θ. However, what extended spatial pattern is present in data can vary from one example to another, and looking at the map is an exploratory approach to get an idea. $\epsilon(A_i)$ in a spatial model represents variation in y that is not connected across an adjacent region, so we refer to it as "pure error" or "random error." Before building a model for $\theta(A_i)$, we want to discuss some adjacency matrix examples.

In general, if a graph has n nodes, the adjacency matrix W is $n \times n$. For example, if a graph has only 7 nodes, as we can see in Figure 2.4, we have a 7×7 symmetric non-negative matrix.

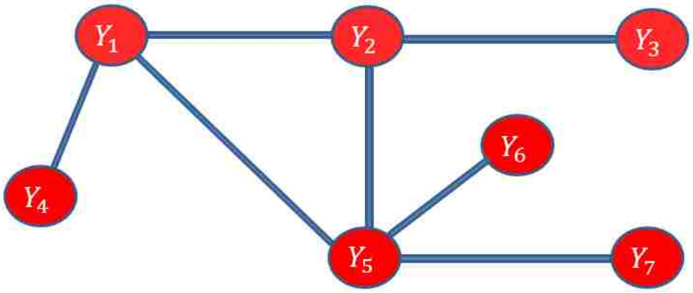


Figure 2.4: A Graph with 7 nodes

All the adjacency matrix entries are 0 or 1 depending on whether there is an edge between any two nodes, and it can be written as follows,

$$W = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

In some cases, instead of putting just 1 and 0 on the edges for present and absent, respectively, one can use edges with weights. An example, of using weights for the edge instead of 0 or 1 can be found in Bhark (2011).

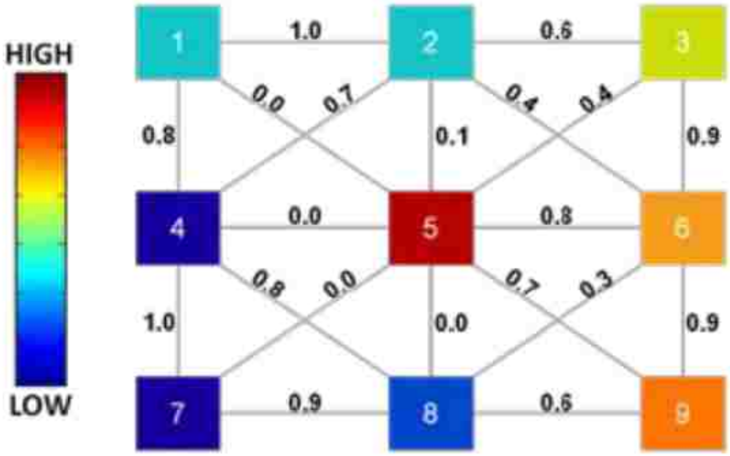


Figure 2.5: A 3 × 3 Grid Cell & Adjacency (Figure 5.1 page 144 in Bhark (2011))

As we can see in Figure 2.5, the weight among the 9 nodes are values between 0 and 1. Its adjacency matrix can be represented as follows,

$$W = \begin{bmatrix} 0 & 1 & 0 & 0.8 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0.6 & 0 & 0.1 & 0.4 & 0 & 0 & 0 \\ 0 & 0.6 & 0 & 0 & 0.4 & 0.9 & 0 & 0 & 0 \\ 0.8 & 0 & 0 & 0 & 0 & 0 & 1 & 0.8 & 0 \\ 0 & 0.1 & 0.4 & 0 & 0 & 0.8 & 0 & 0 & 0.7 \\ 0 & 0.4 & 0.9 & 0 & 0.8 & 0 & 0 & 0.3 & 0.9 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0.9 & 0 \\ 0 & 0 & 0 & 0.8 & 0 & 0.3 & 0.9 & 0 & 0.6 \\ 0 & 0 & 0 & 0 & 0.7 & 0.9 & 0 & 0.6 & 0 \end{bmatrix}$$

To constrict an adjacency matrix from a map, first, we identify the nodes. The nodes are the original units (states, counties, cities, etc) in a map. Next, we identify the edges among the nodes by using one of the three approaches that we discussed above. By following the constructed edges, the adjacency matrix can be formulated as discussed above.

2.4 Empirical Measures of Spatial Association

Before working with spatial models, the association among the units should be checked by using one of the empirical measures of spatial association. The two standard statistical measurements are Moran's I and Geary's C (Banerjee et al., 2014):

- Moran's I is given by the following form

$$I = \frac{n \sum_i \sum_j w_{ij}(\theta_i - \bar{\theta})(\theta_j - \bar{\theta})}{(\sum_{i \neq j} w_{ij}) \sum_i (\theta_i - \bar{\theta})^2}$$

where $I \in$ [-1,1]. If I is significantly different from zero, there is a spatial dependence. But, if I is approximately equal to 0, there is no spatial dependence. I is asymptotically normal with mean equal to $\frac{-1}{n-1}$ under the hypothesis of independence.

- Geary's C is given by the following form

$$C = \frac{(n-1)\sum_i \sum_j w_{ij}(\theta_i - \theta_j)^2}{2(\sum_{i \neq j} w_{ij}) \sum_i (\theta_i - \bar{\theta})^2}$$

where $C \in [0,1]$. If C is significantly different from 1, there is a spatial dependence. But, if C is approximately equal to 1, there is no spatial dependence. C is asymptotically normal under the null hypothesis with mean equal to 1. Both Moran's I and Geary's C are techniques for areal data analogues to what empirical correlation function would represent for point-level data.

2.5 Prior Distribution for Spatial Effect

Since each θ_i is defined on an areal unit, it is not reasonable to use a distent based correlation model for them. Instead, to obtain the joint distribution of θ_i, we use a technical result which is called Brook's Lemma.

Lemma 2.5.1 (Brook's Lemma)**.** *If* $\{\pi(\theta_i|\theta_{-i}), i = 1, 2, \ldots n\}$ *is a set of compatible full conditional distributions and* $\theta_0 = (\theta_{10}, \theta_{20}, \ldots, y_{n0})$ *is any fixed point in the support of* $\pi(\theta_1, \ldots, \theta_n)$*, then*

$$\begin{aligned}\pi(\theta_1, \theta_2, \ldots, \theta_n) &= \frac{\pi(\theta_1|\theta_2, \ldots, \theta_n)}{\pi(\theta_{10}|\theta_2, \ldots, \theta_n)} \times \frac{\pi(\theta_2|\theta_{10}, \ldots, \theta_n)}{\pi(\theta_{20}|\theta_{10}, \ldots, \theta_n)} \\ &\times \;\cdots \times \frac{\pi(\theta_n|\theta_{10}, \ldots, \theta_{n-1,0})}{\pi(\theta_{n0}|\theta_{10}, \ldots, \theta_{n-1,0})} \times \pi(\theta_{10}, \ldots, \theta_{n0}).\end{aligned}$$

It implies the joint distribution can be expressed in term of full conditional distributions. Brook's Lemma gives us the joint distribution up to a normalizing constant, and if $\pi(\theta_1, \theta_2, \ldots, \theta_n)$ is proper, then the normalizing constant is determined by the fact that it integrates to 1. The full conditional distributions need to be specified in away so that they are compatible and simple enough and yet yield useful spatial structure.

To develop these full conditional distributions we need to define the Markov property. By Markov property of θ, we mean the conditional distribution of $\theta(A_i)$ given all other nodes depend only on θ values at all nodes with an edge to A_i. In other words, $\theta\big(A_i|\theta(A_{-i})\big)$ is equivalent to $\theta\big(A_i|\theta(A_{N(i)})\big)$, where $A_{-i} = \bigcup_{j=1}^{n} A_j \setminus A_i$, and $A_{N(i)} = \bigcup_{w_{ij}=1} A_j$.

These conditional distributions may or may not lead to a valid joint distribution. So, we define a Markov Random Field (MRF) as the collection $\{\pi(\theta_i|\theta_{-i}) : i = 1, 2, \ldots, n\}$ that lead to a valid joint distribution $\pi(\theta_1, \theta_2, \ldots, \theta_n)$.

Next, we mention some important definitions and theorems which are going to be used later. In an MRF, we can have a clique, a subset of nodes where there is an edge from every node to every node, for many different orders. In general, if a clique includes k nodes, it is called a clique of order k (potential of order k).

Definition 2.5.1 (Banerjee et al. (2014))**.** A function of k arguments that is exchangeable in these arguments is called a potential function of order k (or simply a potential).

Definition 2.5.2 (Banerjee et al. (2014))**.** The function $\pi(\theta_1, \theta_2, \ldots, \theta_n)$ only through potentials is called a Gibbs distribution.

$$\pi(\theta_1, \theta_2, \ldots, \theta_n) \propto \exp\Big(\gamma \sum_{k} \sum_{\alpha \in M_k} \phi^k(\theta_{\alpha_1}, \theta_{\alpha_2}, \ldots, \theta_{\alpha_k})\Big) \tag{2.3}$$

ϕ^k is a potential of order k, and M_k is the collection of all subsets of size k. α indexes this set, and $\gamma > 0$ is a scale parameter.

Theorem 2.5.2 (Hammersley-Clifford (Besag, 1974))**.** *If we have an MRF, (i.e., if the conditional defines a unique joint distribution), then this joint distribution is a Gibbs distribution.*

Theorem 2.5.3. *(Geman and Geman, 1984) Every Gibbs distribution is an MRF.*

2.6 Conditionally Auto-Regressive Prior

A common choice for a joint distribution for continuous data on R is the Conditionally Auto-Regressive (CAR) prior which was introduced by Besag (1974). In fact, CAR is a Gibbs distribution on potential of order 1 and 2. We derive the Gaussian (or auto-normal) CAR prior following Banerjee et al. (2014). First, let us set the full conditionals,

$$\theta|\theta_i, j \neq i \sim N\Big(\sum_j b_{ij}\theta_i, \tau^2\Big), i = 1, 2, \ldots, n. \tag{2.4}$$

By using Brook's Lemma, we get,

$$\pi(\theta_1, \ldots, \theta_n) \propto \exp\left\{-\frac{1}{2}\theta' D^{-1}(I - B)\theta\right\},$$

where $B = \{b_{ij}\}$ and D is diagonal with $D_{ij} = \tau_i^2$. Eq. 2.4 is a joint multivariate normal distribution for θ with mean and variance equal to 0 and $\Sigma_{\boldsymbol{\theta}} = (I - B)^{-1}D$, respectively. To ensure $\Sigma_{\boldsymbol{\theta}}^{-1}$ is symmetric, we need $\frac{b_{ij}}{\tau_i^2} = \frac{b_{ji}}{\tau_j^2}$ for all i and j; So, we are going to set $b_{ij} = \frac{b_{ij}}{w_{i+}}$ and $\tau_i^2 = \frac{\tau^2}{w_{i+}}$ where $w_{i+} = \#\{j : w_{ij} = 1\}$.

Hence, Eq. 2.4 becomes,

$$\pi(\theta_i|\theta_i, i \neq j) = N\Big(\sum_j \frac{w_{ij}\theta_i}{w_{i+}}, \frac{\tau^2}{w_{i+}}\Big),\ i = 1, 2, \ldots, n.$$

The resulting multivariate prior distribution for θ is:

$$\pi(\theta) \propto \exp\left\{-\frac{1}{2\tau^2}\sum_{i<j} w_{ij}\big(\theta_i - \theta_j\big)^2\right\}$$

This density, being invariant to translation, is improper.

We denote,

$$\theta \sim \mathrm{CAR}\big(\tau^2, W\big).$$

We discuss some of disadvantages of CAR approach in Chapter 4.2 to motivate an alternative model development.

Chapter 3

Krylov Subspace Methods

3.1 Introduction

The computation of a matrix's eigenpairs was improved by using various techniques in the last century. The design of the algorithm depends on the matrix size. There are three different size categories from the eigenvalue algorithms, and they can be classified as follows,

- The small matrices algorithms: these algorithms can compute all of the eigenvalues and eigenvectors for a given matrix. The most notable algorithms are the QR-algorithm (Francis, 1962) for the general and Hermition matrices, and the divide-and-conquer (Cuppen, 1980), and Jacobi algorithms (Parlett and Scott, 1979) methods for Hermition matrices.

- The medium matrices algorithms: in this category, we can follow the algorithms including Jacobi-Davidson (Sleijpen and Van der Vorst, 2000) or various other methods based on the shift-and-invert principle. Also, in this category we can use the same algorithms as in the following category.

- The very large matrices algorithms: the solution to very large system is expensive, so we can reduce the dimension by projecting the very large matrices to small matrices by the following algorithm that applies an application of $matrix \times vector$ operation Ax. The most common algorithms are Arnoldi algorithm for general and Hermition matrices and Lanczos algorithm for Hermitian matrices.

In this chapter, we are going to discuss the Arnoldi and Lanczos algorithms which can be used to evaluate the eigenpairs for the very large matrices. In Section 3.2, we introduce the

Krylov subspace and the general subspace iteration algorithm. Section 3.3 gives a description of Arnoldi algorithm and the basic Arnoldi algorithm. In Section 3.4, we discuss the Lanczos algorithm. Arnoldi algorithms with explicit and implicit restarting are explained in Section 3.5. Convergence of the restarted Arnoldi algorithm is introduced in Section 3.6. Finally, in Section 3.7, we discuss the Implicit Lanczos algorithm with exact and Leja shifts.

3.2 Krylov Subspace

In the power iterative method, one of the oldest methods used to compute eigenpairs for large matrices, we can generate a sequence of vectors $x^{(0)}, Ax^{(0)}, A^2x^{(0)}, \ldots$. The additional information can be ignored by the single vector power iteration, but it can be extracted through various linear combinations of the power sequence (Saad, 2003). The Krylov subspace can be considered to attempt to formulate the best possible approximation of the eigenpairs.

The k-dimensional subspace, spanned by a given vector v and with increasing powers of A applied to v until the $(k-1)$-th power, is called the k-dimensional Krylov subspace and can be defined as, $K_k(A, v) = span\{v, Av, A^2v, \ldots, A^{k-1}v\}$. Notice that there is an interesting connection between Krylov subspaces, so for a fixed matrix A, depending on the starting vector v, there exist some $k \leq n$ so that:

$$K_1(v) \subset K_2(v) \subset K_3(v) \subset \cdots \subset K_k(v) = K_{k+1}(v) = \cdots \tag{3.1}$$

The iterative methods, Arnoldi and Lanczos methods, are based on Krylov subspace projection to extract a small number of eigenpairs from large sparse matrices.

A Galerkin condition, a Ritz vector $x \in K_k(A, v1)$ corresponding Ritz value θ if the Galerkin condition $\langle w, Ax - \theta x\rangle = 0$ for all $w \in K_k(A, v_1)$ is satisfied, can be used to construct an approximate eigenpair from the Krylov subspace, where for any two vector u and v, $\langle u, v\rangle$ is the inner or dot product.

In other words, the eigenpairs (θ, x) can be obtained by imposing the Galerkin condition: $\langle w, AVy - \theta Vy\rangle = 0$, for all $v_j \in V = (v_1, \ldots, v_k)$, where $V^H V = I$, A is $n \times n$ matrix, $x = Vy$ and K an m-dimension subspace.

Therefore, y and θ must satisfy $B_k y = \theta y$ where $B_k = V^H AV$. Each eigenvalue θ_i of B_k is called Ritz value and Vy_i is called Ritz vector. When y_i is the eigenvector of B_k associated with θ_i, this is called Rayleigh-Ritz procedure (Saad, 2003).

The following algorithm, Algorithm 3, shows us the general sketch that all algorithms follow for finding approximate eigenpairs from the large matrices.

Algorithm 3 General Subspace Iteration

```
INPUT: An subspace $X^{(1)}$, integers $g$ and $N$
OUTPUT: $g$ eigenpair

for $k = 1, 2, \ldots, N$ do
    seclect $g$ eigenpair from $X^{(k)}$;
    if all $g$ eigenpair are converged then
        output the approximations and exit;
    else
        using information from $X^{(k)}$, build a new subspace $X^{(k+1)}$;
    end if
end for
write misconvergence and exit;
```

Theorem 3.2.1 (Saad (2003))**.** *For k defined by Eq. 3.1, The subspace $K_k(v)$ is the smallest A-invariant subspace of $\mathbb{C}^n$ that contains the vector v.*

- *Let μ be the nonzero polynomial of smallest possible degree such that $\mu(A)v = 0$, then $deg(\mu) = k$.*

- *If A is diagonalizable, then $K_k(v) = span\{P_\lambda v : P_\lambda v \neq 0, \lambda \in \Lambda(A)\}$, where P_λ is the spectral projector associated with the eigenvalue λ.*

The methods that are used to evaluate an approximate eigenpairs using the Krylov space are called the Krylov subspace methods. We can classify the Krylov subspace methods into four different families. This is dependent upon the manner of identifying $x \in K_k(A; v)$ (Saad, 2003).

1. Ritz-Galerkin: in this family, u^k is constructed in a way so that the residual must be orthogonal to the Krylov subspace, i.e., $r^{(k)} \perp K_k(A; r^{(0)})$, where $r^{(k)} = Au^{(k)} - \lambda^{(k)} u^{(k)}$.

2. Petrov-Galerkin: in this family, u^k is constructed so that the residual is orthogonal to other subspaces, i.e., $r^{(k)} \perp L_k$.

3. Residual norm minimization: in this family, u^k is constructed so that $||r^{(k)}||_2$ is minimized over $K_k(A; r^{(0)})$.

4. Error norm minimization: in this family, u^k is constructed so that $||r^{(k)}||_2$ is minimized over $K_k(A^T; r^{(0)})$.

3.2.1 Ritz Approximation

Now, we have to construct an orthonormal basis for Krylov subspace to compute a Ritz approximation. The important properties of the basis are obtained from the following theorem:

Theorem 3.2.2 (Bujanović (2011)). *Let the $dim(K_m(A; v)) = m$, and let $v_1, v_2, \ldots, v_m$ denote the sequence of vectors generated by the Gram-Schmidt procedure when run on vectors $v, Av, A^2v, \ldots, A^{m-1}v$, respectively. Then,*

1. *For $j < m$, $Av_j \in K_{j+1}(A; v)$, and $AV_j \notin K_j(A; v)$.*

2. *H_m is an unreduced $m \times m$ upper Hessenberg matrix, i.e., none of the elements β_1, β_2, $\ldots$, β_{m-1} on its first subdiagonal is equal to zero.*

3. *The following equality holds for all $j < m$:*

$$AV_j = V_j H_j + \beta_j v_{j+1} e_j^* \tag{3.2}$$

In particular, the last column reveals a recursive relation:

$$\beta_j v_{j+1} = Av_j - V_j h_j = Av_j - V_j(V_j^* Av_j) = (I - V_j V_j^*)Av_j \tag{3.3}$$

where h_j is the last column of H_j.

4. *Suppose $\tilde{V_j}$ is a $n \times j$ orthonormal matrix, $\tilde{H_j}$ is an unreduced $j \times j$ upper Hessenberg matrix with positive subdiagonal elements and $\tilde{r_j}$ is a vector in C^n such that $A\tilde{V_j} = \tilde{V_j}\tilde{H_j} + \tilde{r_j}e_j^*$. If the first column of $\tilde{V_j}$ is equal to v_1, then $\tilde{V_j} = V_j, \tilde{H_j} = H_j$ and $\tilde{r_j} = \beta_j v_{j+1}$. This is called the "Implicite-Q theore".*

5. *Let (θ, x) denote a Ritz Pair from $K_j(A; v)$; let $x = V_j y$. Then*

$$||Ax - \theta x|| = |\beta_j|\,|e_j^* y|. \tag{3.4}$$

Eq. 3.2 is called an Arnoldi decomposition, where V_j and H_j are defined in (4). Eq. 3.3 describes how to build the Arnoldi algorithm, which is going to be described in the following section, by computing the matrices V_j and H_j. To check that a single Ritz pair is converged, we determine the size of the residual norm in Eq. 3.4. Extensive research has been done to study the convergence of actual values and *Ritz* approximations from Krylov subspace (Saad, 2003).

3.3 Arnoldi Methods

In 1951, Aroldi introduced a method to transform a general matrix into the Hessenberg form of dimension $k \leq n$. This method constructs an orthogonal basis of the Krylov subspace. When a matrix is large and sparse, this method will be suitable.

For a fixed k and a unit vector $||v||_2 = 1$, the classical Gram-Schmidt procedure can be used to perform the mentioned orthogonalization process.

So, for $k = 1, 2, 3, \dots, m$.

$$h_{ik} = (v_i, Av_k), i = 1, 2, \dots, k \tag{3.5}$$

$$\hat{v}_{k+1} = Av_k - \sum_{i=1}^{k} h_{ik} v_i, \tag{3.6}$$

$$h_{k+1,k} = ||\hat{v}_{k+1}||_2 \tag{3.7}$$

$$v_{k+1} = \hat{v}_{k+1} / ||\hat{v}_{k+1}||_2 \tag{3.8}$$

In Eq. 3.7, if $h_{k+1,k} = 0$, the algorithm stops because Eq. 3.8 is undefined. The procedure is called the classical Gram-Schmidt orthogonalization procedure, and the vectors $v_1, v_2, \dots, v_{k+1}$ are called Arnoldi vectors.

Due to the presence of rounding errors, the loss of orthogonality appears when we use the classical Gram-Schmidt procedure. So, the modified Gram-Schmidt procedure is a simple remedy for this situation and can be explained by the following: Given a vector v_1 with $||v_1||_2 = 1$,

```
For $k = 1, 2, \dots, m$ , do:

    $w = Av_k$
        For $i = 1, 2, \dots, k$:
        $h_{ik} = (w, v_i)$
        $w = w - h_{ik} v_i$
        End for
    $h_{k+1,k} = ||w||_2$        $v_{k+1} = w / h_{k+1,k}$
End for.
```

Both procedures, classical and modified Gram-Schmidt, perform the same arithmetic operations, so they have the same computational costs. It is easy to verify that they are equivalent in exact arithmetic (without rounding errors). Householder reflectors are another implementation used to achieve the required orthogonality (Walker, 1988). It is more accurate, but more expensive because the operation count is increased.

In the Arnoldi algorithm, we get $v_1, v_2, \ldots, v_{k+1}$ is an orthogonal basis for $K_k(A; v)$ when k steps are run. So, if we define a $n \times k$ matrix satisfying $V_k^T V_k = I_k$, such that

$$V_k = [v_1, v_2, \ldots, v_k] \in \mathbb{R}^{n \times k} \tag{3.9}$$

Then, we have

$$V_k^T A V_k = H_{kk} \tag{3.10}$$

and

$$V_{k+1}^T A V_k = H_{k+1,k} \tag{3.11}$$

where $H_{k+1,k} \in \mathbb{R}^{(k+1) \times k}$ is the upper Hessenberg matrix with h_{ij} computed by Arnoldi algorithm.

Algorithm 4 Basic Arnoldi

```
INPUT: Initial unit vector $v_1$, integers $m$ and $k$
OUTPUT: $k$ dominant eigenpairs from $K_m(v_1; A)$
set $V_1 = [v_1]$;
for $j = 1, 2, \ldots, m$ do
    compute the last column $h_j$ of $H_j$ and define $t = Av_j$;
    compute $h_j = V_j^*(Av_j) = V_j^* t$;
    if $j > 1$ then
        $H_j = \begin{bmatrix} H_{j-1} & h_j(1:j-1) \\ \beta_{j-1}e_j^* & h_j(j) \end{bmatrix};$
    else
        $H_1 = h_1 = v_1^* A v_1$;
    end if
    compute the orthogonal projection onto $\mathrm{Im} V_j^{\perp}$:
    define $r_j = (I - V_j V_j^*)(Av_j) = t - V_j h_j$, $\beta_j = ||r_j||$;
    if $\beta_j = 0$ then
        $V_j$ is $A$-invariant; report convergence and exit.
    end if
    $v_{j+1} = r_j/\beta_j$, $V_{j+1} = [V_j v_{j+1}]$;
    compute Ritz pairs from $K_j(v_1; A)$
    calculate the dominant eigenvectors $(\lambda_1, y_1), (\lambda_2, y_2), \ldots, (\lambda_k, y_k)$ of $H_j$;
    if $j \geq k$ and error of Ritz pairs $(\lambda_i, V_j y_i)$ are small then
        write convergence and exit
    end if
end for
```

3.4 Lanczos Methods

For a Hermition matrix, A, we can use the Lanczos algorithm, which is a reduced form of the Arnoldi algorithm. In this algorithm, the matrix H_j becomes a tridiagonal matrix since it is a Hermition upper Hessenberg matrix.

Suppose,

$$H_j = \begin{bmatrix} \alpha_1 & \beta_1 & & & & \\ \beta_1 & \alpha_2 & \beta_2 & & & \\ & \beta_2 & \alpha_3 & \beta_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \beta_{j-2} & \alpha_{j-1} & \beta_{j-1} \\ & & & & \beta_{j-1} & \alpha_j \end{bmatrix}$$

where $\beta_i \in \mathbb{R}^+$, for $i = 1, \ldots, j-1$.

From the Arnoldi decomposition, we have:

$$AV_j = V_j H_j + \beta_j v_{j+1} e_j^*$$

In the Lanczos algorithm, we are going to have the three term recursive relation,

$$\beta_j v_{j+1} = Av_j - \beta_{j-1} v_{j-1} - \alpha_j v_j \tag{3.12}$$

In exact arithmetic we need to orthogonalize Av_j only against v_{j-1} and v_j to get v_{j+1}.

When we use Eq. 3.12 in finite precision to find v_{j+1}, the orthogonality may be lost against $v_1, v_2, \ldots, v_{j-2}$. Hence, extensive research has been conducted in this area to resolve this problem. The most common solutions are:

- Complete reorthogonalization: [See Parlett (1980)]
- Partial and selective reorthogonalization: [See Parlett and Scott (1979)]
- No additional reorthogonalization: [See Cullum and Willoughby (1981)]

Algorithm 5 Basic Lanczos

```
INPUT: Initial unit vector $v_1$, integers $m$ and $k$.
OUTPUT: k dominant eigenpairs from $K_m(v_1; A)$
$\beta_0 = 0$;
for $j = 1, 2, \ldots, m$ do
    compute the element in the last column in $H_j$ and $t = Av_j$;
    $\alpha_j = v_j^*(Av_j) = v_j^* t$;
    if $j > 1$ then
        $H_j = \begin{bmatrix} H_{j-1} & \beta_{j-1}e_j \\ \beta_{j-1}e_j^* & \alpha_j \end{bmatrix};$
    else
        $H_1 = \alpha_1$;
    end if
    compute the orthogonal projection onto $ImV_j^{\perp}$:
    compute $r_j = (I - V_jV_j^*)(Av_j) = t - \beta_{j-1}v_{j-1} - \alpha_j v_j$ and $\beta_j = ||r_j||$;
    if $\beta_j = 0$ then
        $V_j = [v_1 v_2 \cdots v_j]$ is $A-$invariant; report convergence and exit.
    end if
    set $v_{j+1} = r_j/\beta_j$ and evaluate Ritz pairs from $K_j(v_1; A)$
    compute the dominant eigenvector $(\lambda_1, y_1), (\lambda_2, y_2), \ldots, (\lambda_k, y_k)$ of $H_j$;
    if $j \geq k$ and residuals of Ritz pairs $(\lambda_i, V_j y_i)$ are small enough then
        report convergence and exit
    end if
end for
```

3.5 Arnoldi Algorithm with Restarting

The goal of the large scale algorithm is to determine a few eigenvalues and their corresponding eigenvectors. It is not easy to determine the number of steps that the Arnoldi algorithm will require to compute the desired, g, eigenpairs. Also, the computer's capacity may not be enough to store the basis for K_k when the matrix is very large. The time, moreover, will be increased because the dimension of the projection matrix increases. Hence, with certain properties, the number of unwanted Ritz approximation becomes larger and larger when we want to compute the g eigenpairs exactly. Therefore, we restart the Arnoldi algorithm to avoid these unwanted effects.

3.5.1 Explicit Restart

After approximating some eigenpairs by completing m steps of the basic Arnoldi algorithm, we choose a new initial vector, $v^{(1)}$ and run additional m steps. We continue repeating this procedure until we get the desired eigenpairs. It is clear that when we follow this approach, the algorithm will require a fixed amount of memory.

If the subspace $K_m(v; A)$ is invariant (the initial vector v belongs to a subspace spanned by m eigenvectors of the matrix A), the Ritz values will exactly coincide with the eigenvalues of matrix A.

Let us denote the eigenpairs that we want to compute by: $(\lambda_1^*, u_1^*), (\lambda_2^*, u_2^*), \ldots, (\lambda_g^*, u_g^*)$, and the other eigenpairs by: $(\lambda_1', u_1'), (\lambda_2', u_2'), \ldots, (\lambda_{n-g}', u_{n-g}')$ where $g \leq m$. Let

$$v^* = \sum_{j=1}^{g} \xi_j u_j^*, \qquad 0 \neq \xi_j \in C$$

be the ideal starting vector that requires at most m basic Arnoldi steps for computing the wanted eigenpairs (Sorensen, 1992).

If the Arnoldi algorithm starts with $v = \sum\limits_{j=1}^{g} \xi_j^* u_j^* + \sum\limits_{j=1}^{n-g} \xi_j' u_j'$, then a logical choice for the initial vector is

$$v^{(1)} = \pi(A)v = \sum_{j=1}^{g} \xi_j^* \pi(\lambda_j^*) u_j^* + \sum_{j=1}^{n-g} \xi_j' \pi(\lambda_j') u_j', \text{ where } |\pi(\lambda_j')| \gg |\pi(\lambda_k')|$$

Here, π is called a polynomial filter.

Algorithm 6 includes all the development steps for the explicit Arnoldi algorithm.

Algorithm 6 Arnoldi with Explicit Restart

```
INPUT: Initial vector $v^{(0)}$, integers $g$, $m$, $p$ and $A_{n\times n}$.
OUTPUT: Approximations of $g$ wanted eigenpairs.
$i = 0$;
while TRUE do
    run $m$ steps of the Arnoldi algorithm with initial vector $v^{(i)}$:
    $AV_m^{(i)} - H_m^{(i)} V_m^{(i)} = \beta_m^{(i)} v_{m+1}^{(i)} e_m^*$;
    compute Ritz pairs $(\lambda_{i,j}^*, v_{i,j}^*)$ and $(\lambda'_{i,j}, v'_{i,j})$;
    if all $g$ Ritz pairs $(\lambda_{i,j}^*, v_{i,j}^*)$ are accurate approximations of the eigenpairs then
        report Ritz pairs as eigenpair approximations and exit the loop;
    else
        using the computed Ritz pairs, choose a polynomial $\pi \in P_p$; $v^{(i+1)} = \pi(A)v^{(i)}; i = i + 1$;
    end if
end while
```

3.5.2 Implicit Restart

In each step of the Arnoldi algorithm with an explicit restart, we require $m+p-1$ matrix-vector multiplication. The m Arnoldi steps to build up the Krylov subspace require $m-1$ multiplication, and the p additional multiplications come from restarting with a polynomial of degree p. In 1992, Sorensen showed a way of restarting so that in each step, only p matrix-vector multiplications are required. In the Implicitly Restarted Arnoldi (IRA) algorithm with exact shift, the eigenvalues of the matrix H_m are the unwanted Ritz eigenvalues. Assume σ_i is an eigenvalue of the matrix H_m, then $H_m - \sigma_i I$ is a singular matrix, and the bottom right element of the tridiagonal vector R^i in QR-factorization, $H_m - \sigma_i I = Q^i R^i$ is equal to zero.

In Algorithm 7, we see the important steps of the IRA algorithm.

Algorithm 7 Arnoldi with implicit Restart

```
INPUT: Initial vector $v^{(0)}$, integers $g$, $m$ and $p$.
OUTPUT: $g$ eigenpairs.
$i = 0$;
compute $m$ steps of the Arnoldi algorithm with initial vector $v^{(0)}$:
$AV_m^{(0)} = H_m^{(0)} V_m^{(0)} + r_m^{(0)} e_m^*$;
compute Ritz pairs $(\lambda_{i,j}^*, v_{i,j}^*)$ and $(\lambda_{i,j}', v_{i,j}')$;
while not all $g$ Ritz pairs $(\theta_{i,j}^*, v_{i,j}^*)$ are accurate approximations of eigenpairs of $A$ do
    choose shifts $\sigma_1, \sigma_2, \ldots, \sigma_p \in \mathbb{C}$ using the computed Ritz pairs;
    $Q = I_m$;
    for $j = 1, 2, \cdots, p$ do
        modify $H_m^{(i)}$ and $Q$ by calling $(H_m^{(i)}, \sigma_j, Q)$
    end for
    $\hat{\beta} = Q_{m,m-p}$; $\tilde{\beta} = H_{m-p+1,m-p}$, and $\tilde{r}_{m-p} = \hat{\beta} r_m^{(i)} + \hat{\beta} V_m^{(i)} Q e_{m-p+1}$;
    $\tilde{V}_{m-p} = (V_m^{(i)}.Q)(:, 1 : m-p)$, and $\tilde{H}_{m-p} = H_m^{(i)}(1 : m-p, 1 : m-p)$;
    $i = i + 1$;
    compute $p$ more steps of the Arnoldi algorithm started with
    $A\tilde{V}_{m-p} = \tilde{V}_{m-p} \tilde{H}_{m-p} + \tilde{r}_{m-p} e_{m-p}^*$;
    denote $AV_m^{(i)} = H_m^{(i)} V_m^{(i)} + r_m^{(i)} e_m^*$;
    compute Ritz pairs $(\lambda_{i,j}^*, v_{i,j}^*)$ and $(\lambda_{i,j}', v_{i,j}')$
end while
```

3.6 Convergence of The Restarted Arnoldi Algorithm

Extensive research has been conducted on the convergence theory of the restarted Arnoldi algorithm. Let

$$AV_m^{(i)} = V_m^{(i)} H_m^{(i)} + \beta_m^{(i)} v_{m+1}^{(i)} e_m^* \tag{3.13}$$

be a sequence of the Arnoldi output where the columns of $V_m^{(i)}$ consist of an orthogonal basis for a Krylov subspace, $K_m(A; v^{(i)})$. Let $v^{(i)}$ be a starting vector such that

$$v^{(i+1)} = \pi_i(A) v^{(i)} \tag{3.14}$$

where π_i is a polynomial filter at i^{th} restarted step. For all i and some polynomial π of degree p, if the filter stays the same at π and π_i, then the convergence is obvious.

Now, suppose $\lambda_1, \lambda_2, \ldots, \lambda_n$ are ordered eigenvalues for the diagonalizable matrix A such that:

$$|\pi(\lambda_1)| \geq |\pi(\lambda_2)| \geq \cdots \geq |\pi(\lambda_{m-p})| > |\pi(\lambda_{m-p+1})| \geq |\pi(\lambda_{m-p+2})| \geq \cdots \geq |\pi(\lambda_n)|.$$

Let $u_1, u_2, \ldots, u_n$ be the associated eigenvectors, and for the initial vector $v^{(1)}$, we have:

$$v^{(1)} = \sum_{j=1}^{n} \xi_j u_j, \text{ where } \quad \xi_1, \xi_2, \ldots, \xi_{m-p} \neq 0$$

With some normalizing vector, the restarted initial vector $v^{(i)}$ satisfies the following:

$$v^{(i)} = \alpha_i \sum_{j=1}^{n} \xi_j (\frac{|\pi(\lambda_i)|}{|\pi(\lambda_m - p)|})^i u_j,$$

The restarted Arnoldi method will converge because the components of the vectors $u_1, u_2, \ldots, u_{m-p}$ begin to dominate, forcing $v^{(i)}$ ultimately into an invariant subspace. In other words, the angle between the Krylov subspace and the eigenspace associated with $\lambda_1, \lambda_2, \ldots, \lambda_{m-p}$ converges to zero.

In 1992, Sorensen showed that when A is a Hermitian Matrix, the restarted Arnoldi algorithm converges with such filters.

Theorem 3.6.1 (Sorensen (1992)). *Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ be the eigenvalues of the Hermitian matrix A and $u_1, u_2, \ldots, u_n$ the associated eigenvectors. Consider a sequence of Arnoldi decomposition Eq. 3.13 and Eq. 3.14 such that the polynomial filter π_i has degree p and roots $\theta^{(i)}_{m-p+1}, \theta^{(i)}_{m-p+2}, \ldots, \theta^{(i)}_{n}$. Here, $\theta^{(i)}_1 \geq, \theta^{(i)}_2 \geq, \ldots, \theta^{(i)}_m$ are the Ritz values computed from the Arnoldi decomposition Eq. 3.13 . Suppose that the starting vector $v^{(1)}$ has a non-zero component in the direction of each of $u_1, u_2, \ldots, u_{m-p}$.*

Furthermore, suppose that $H^{(i)}_{m-p}(j+1, j) > \epsilon > 0$ for all i and j. Then $\forall i$, $\theta^{(i)}_j \longrightarrow \lambda_j$ for all $j = 1, 2, \ldots, m-p$.

However, in 2009, Embree showed that Theorem 3.4 is not valid for the Non-Hermiatian matrices (Embree, 2009).

3.7 Implicit Restarted Lanczos

The Basic Lanczos Algorithm (BLA), Algorithm 6, is a suitable algorithm for finding the few largest or smallest eigenpairs related to very large symmetric matrices. But the accuracy of computing approximated eigenpairs can be reduced because of the loss of orthogonality of computing the Krylov subspace. Also, the BLA suffers from large storage requirements. However, the Implicit Restarted Lanczos (IRL) algorithm has been studied to address these difficulties. The IRL algorithm is a special case of the IRA algorithm, and it can be obtained by specializing the IRA to symmetric case. After k steps of the factorization, we have

$$AV = VH + fe_k^T \tag{3.15}$$

where $V \in \mathbb{R}^{n\times k}$, $V^TV = I_k$, $H \in \mathbb{R}^{k\times k}$ is symmetric and tridiagonal and $f \in \mathbb{R}^n$ with $V^Tf = 0$. Eq. 3.15 also can be written as

$$AV = (V, v)(H\beta e_k^T)^T,\ \beta = ||f||,\ v = \frac{1}{\beta}f \tag{3.16}$$

Eq. 3.15 is a truncation of the complete reduction of matrix A $\in \mathbb{R}^{n\times n}$ to the tridiagonal form.

Let (θ, y) be an eigenpair of matrix H. Then the vector $x = Vy$ is referred to as a Ritz vector and θ as a Ritz value of A when Eq. 3.17 is satisfied.

$$||Ax - x\theta|| = ||(AV - VH)y|| = |\beta e_k^T y| \tag{3.17}$$

It is clear that the residual error, Eq. 3.17, associated with the Ritz pair (θ, x) can be determined by evaluating $|\beta e_k^T y|$ without computing the Ritz vector x explicitly.

Using iterative refinement to orthogonalized f against V as it is computed is the best way to solve the loss of orthogonality in the BLA because the computations required can be expressed in terms of the BLA and because the columns of V can be updated with the orthogonal transformations required to perform the implicitly shifted QR-steps.

If g is the number of the desired eigenpairs, then the Lanczos algorithm replaces the general matrix, $A_{n\times n}$, with an orthonormal matrix, $V_{(k+p)\times(k+p)}$, and a tridiagonal matrix, $H_{(k+p)\times(k+p)}$, where p is not much larger, and may be smaller, than k, such that:

$$AV_{k+p} = V_{k+p}H_{k+p} + r_{k+p}e_{k+p}^T \tag{3.18}$$

This can be done after $k+p$ iterations. For more information, see Calvetti et al. (1994).

3.7.1 Shift Selection

The convergence of the IRL algorithm can be determined by the selection of the shifts; the values $\nu_1, \nu_2, \ldots, \nu_n$ are called shifts, where $p(z) = (z-\nu_1)(z-\nu_1)\cdots(z-\nu_1)$. The two common kinds of shifts that can be used with the IRL algorithm are exact shifts and Leja shifts. Exact shifts can be obtained by selecting p eigenvalues from all the computed eigenvalues of the tridiagonal matrix (Saad, 2003). The Leja shifts are Leja points for an interval on the real axis that contains unwanted eigenvalues.

Before we introduce the algorithm that can be used to compute the p Leja shifts (Leja points) we need to give some basic definitions. Let $\mathbb{C}$ denote the complex plane, $\mathbb{R}$ real numbers and let $\Omega \subset \mathbb{C}$ be a compact set. $G(s,t)$ is called a Green's function, and it is uniquely determined by the requirements (i) $\Delta G(s,t) = 0$ in Ω, (ii) G(s,t) = 0 on $d\Omega$ and (iii) $\int_{d\Omega} \frac{\partial}{\partial n} G(s,t) d\sigma = 1$, where $\frac{\partial}{\partial n}$ denotes the normal derivative direction into Ω and $d\sigma$

stands for the element of arc length. The nonnegative number denoted by

$$c = \lim_{|z|\to\infty} |z| \exp(-G(s,t)),\ z = s + it$$

is called the capacity of K, and it depends on the size of K.

Let $w(z)$ be the continuous weight function on K, such that

$$\alpha \leq w(z) \leq \beta,\ z \in K \tag{3.19}$$

for some constants $0 < \alpha \leq \beta < \infty$, and introduce a sequence of points, $\{z_j\}_{j=1}^{\infty}$ of points in K as follows. Let z_0 be a point such that

$$w(z_0)|z_0| = \max_{z\in K} w(z)|z|,\ z_0 \in K \tag{3.20}$$

and let z_j satisfy

$$w(z_j)\prod_{l=0}^{j-1}|z_j - z_l| = \max_{z\in K} w(z)\prod_{l=1}^{j-1}|z - z_l|,\ z_j \in K,\ j = 1, 2, \ldots \tag{3.21}$$

The sequence of points $\{z_j\}_{j=1}^{j=\infty}$ that satisfy Eq. 3.19 and Eq. 3.20 is called weighted Leja points or Leja shifts for K. The following algorithm, Algorithm 6, shows the main steps for computing the Leja shifts.

Algorithm 8 Compute p Leja shifts for K_j given points $\{z_k\}_{k=0}^{r-1}$

```
INPUT: Initial values $a_j, b_j, \lambda_{k+1}, r, \{z_k\}_{k=0}^{r-1}$.
OUTPUT: Leja shifts $\{z_k\}_{k=r}^{p+r-1}$
 1: $k = r$;
 2: if $k = 0$ then
 3:     set $z_0 = b_j$
 4: else
 5:     determine $z_k \in K$, so that
        $w(z_j) \prod_{l=0}^{j-1} |z_j - z_l| = \max_{z \in K} w(z) \prod_{l=1}^{j-1} |z - z_l|$, where $w(z) = |z - \lambda_{k+1}|$
 6: end if
 7: set $k = k + 1$
 8: if $k < p + r$ then
 9:     GO TO STEP 2
10: else
11:     STOP;
12: end if
```

3.7.2 Implicit Restarted Lanczos with Exact Shifts

The IRL method is a polynomial acceleration scheme, and on the choice of acceleration polynomials, the rate at which eigenvalues and invariant subspaces are determined depends on the IRL algorithm with exact shifts, implicitly shifted QR-algorithm, and yield convergence. However, this convergence can be very slow (Calvetti et al., 1994). The best choices for exact shifts are the remaining eigenvalues of the $(k+p) \times (k+p)$ symmetric tridiagonal matrix that we got after Lanczos factorization. Let us assume the eigenvalues are ordered according to

$$\theta_1 < \theta_2 < \cdots < \theta_{(k+p)} \tag{3.22}$$

Let (θ_j, x_j) be a Ritz eigenpair for matrix A. Then from Eq. 3.17 we get,

$$||Ax_j - x_j\theta_j|| = |\beta_{k+p} e_{k+p}^T y_j|, \; 1 \leq j \leq k+p \tag{3.23}$$

where β_{k+p} is defined by Eq. 3.18. In the following algorithm, Algorithm 9, we explain the main steps of the IRL with exact shifts.

Algorithm 9 IRL-ES

```
INPUT: Initial values $A, k, p, v_1, \epsilon$;
OUTPUT: Set of eigenpairs $\{\hat{\lambda_j}, \hat{v_j}\}_{j=1}^{k}$;
determine the Lanczos factorization Eq. 3.18
compute the eigenvalues Eq. 3.22 of the $H_{k+p}$
if $\max\limits_{1 \le j \le k+p} ||Ax_j - x_j\lambda_j|| = |\beta_{k+p} e_{k+p}^T y_j| \le \epsilon|\lambda_j|$, where $\epsilon > 0$, then STOP;
apply the exact shifts: $\mu = \lambda_j,\ j = k+1, \ldots k+p$
advance Lanczos factorization p steps in order to obtain Eq. 3.18
```

3.7.3 Implicit Restarted Lanczos with Leja Shifts

There are two kinds of IRL algorithms with Leja shifts. Specifying how to choose the endpoints a and b of the interval K is the key point to obtain an IRL algorithm based on the Leja shifts. Let $K = [a, c] \cup [d, b]$ be the union of two real intervals with end points a, b, c, d. The first algorithm uses Leja shifts for a sequence of nested intervals $K_j = [a_j, b_j]$. First, when we determine the eigenvalues of the symmetric tridiagonal matrix $H_{(k+p)}$ in the algorithm, we define the initial interval $K_0 = [a_0, b_0]$ with

$$a_0 = \theta_{k+1}, \quad b_0 = \theta_{k+p} \tag{3.24}$$

For each new eigenvalue, we update $K_j = [a_j, b_j]$ by $K_{j+1} = [a_{j+1}, b_{j+1}]$, where

$$a_{j+1} = \min\{a_j, \theta_{k+1}\},\ b_{j+1} = \max\{b_j, \theta_{k+p}\}, \quad j = 0, 1, 2, \ldots \tag{3.25}$$

The eigenvalues of the symmetric tridiagonal matrix, H_{k+p}, satisfy

$$\lambda_k \le \theta_k, \quad \theta_{k+p} \le \lambda_n$$

The sequence of intervals $K_j = [a_j, b_j],\ j \geq 0$, defined by Eq. 3.24 and Eq. 3.25 satisfies:

$$K_j = [a_j, b_j] \subset K_{j+1} = [\lambda_{k+1}, \lambda_n]$$

The following algorithm, Algorithm 9, shows the main steps of the IRL for Leja Shifts with Nested Intervals (IRL-LSNI)(Calvetti et al., 1994).

Algorithm 10 IRL-LSNI

```
INPUT: Initial values $A, k, p, v_1, \epsilon$;
OUTPUT: Set of eigenpairs $\{\hat{\lambda_j}, \hat{u_j}\}_{j=1}^{k}$;
1: determine the Lanczos factorization Eq. 3.18
2: compute the eigenvalues Eq. 3.22 of the $H_{k+p}$
3: If $\max\limits_{1 \leq j \leq k+p} ||Ax_j - x_j\lambda_j|| = |\beta_{k+p} e_{k+p}^T y_j| \leq \epsilon|\lambda_j|$, where $\epsilon > 0$, then STOP;
4: if $j = 0$ then define the interval $K_j = [a_j, b_j]$ by Eq. 3.24 else by Eq. 3.25
5: compute p Leja shifts $\{z_k\}_{k=jp}^{(j+1)p-1}$ for $K_j$ by Algorithm 8
6: apply shifts $\mu = z_k,\ k = jp, jp+1, \ldots, (j+1)p-1$
7: apply the exact shifts: $\mu = \lambda_j, j = k+1, \ldots k+p$
8: advance Lanczos factorization p steps in order to obtain Eq. 3.18; j = j+1; Go to 2;
```

An updated version of the IRL-LSNI algorithm is the IRL for Leja Shifts with Free Left Endpoints (IRL-LSFLE) algorithm, where the endpoints of the intervals $K_j = [a_j, b_j]$ are updated according to

$$a_{j+1} = \theta_{k+1}, \quad b_{j+1} = \max\{b_j, \theta_{k+p}\}, \quad j = 0, 1, 2, \ldots \tag{3.26}$$

In this case, the sets of K_j will be smaller than the set that we get in the IRL-LSNI algorithm. In the following algorithm, Algorithm 11, we see the main steps for the IRL-LSFLE algorithm (Lehoucq et al., 1998).

Algorithm 11 IRL-LSFLE

```
INPUT: Initial values $A, k, p, v_1, \epsilon$;
OUTPUT: Set of eigenpairs $\{\hat{\lambda}_j, \hat{u_j}\}_{j=1}^{k}$;
1: determine the Lanczos factorization Eq. 3.18
2: compute the eigenvalues Eq. 3.22 of the $H_{k+p}$
3: if $\max_{1 \leq j \leq k+p} ||Ax_j - x_j\lambda_j|| = |\beta_{k+p} e_{k+p}^T y_j| \leq \epsilon|\lambda_j|$, where $\epsilon > 0$, then STOP;
4: if $j = 0$ then define the interval $K_j = [a_j, b_j]$ by Eq. 3.24 else by Eq. 3.26
5: compute p Leja shifts $\{z_k\}_{k=jp}^{(j+1)p-1}$ for $K_j$ by Algorithm 8
6: apply shifts $\mu = z_k$, $k = jp, jp+1, \ldots, (j+1)p-1$
7: apply the exact shifts: $\mu = \lambda_j, j = k+1, \ldots k+p$
8: advance Lanczos factorization p steps in order to obtain Eq. 3.18; j = j+1; GO TO 2.
```

3.8 Conclusion

The complexity of computing all eigenpairs of any $n \times n$ matrix using the QR method is $\mathcal{O}(n^3)$. If we are interested in finding only k smallest or largest eigenvalues of a symmetric matrix, we can use Lanczos algorithm. The idea of the Lanczos algorithm is to replace the real symmetric $n \times n$ matrix by a real symmetric tridiagonal $(k+p) \times (k+p)$ matrix where k is the number of desired eigenvalues and p is close to k. This substantially improves the computational efficiency as computing all eigenvalues for an $n \times n$ tri-diagonal matrix is of $\mathcal{O}(nk^2)$ complexity. However, the basic Lanczos algorithm can suffer from large storage requirement and memory issues. The IRL algorithm has been studied to address some of these problems. The IRL can compute k largest or smallest eigenvalues and the associated eigenvectors. In our work, the IRL-LSFLE is going to be used, and it was implemented using RSpectra package (Qiu and Mei, 2016).

Chapter 4

A Computationally Efficient Hierarchical Model for Large Areal Data

4.1 Introduction

As we have described in Chapter 2, datasets indexed by geographical features are common in different fields of scientific applications. For many of these datasets, the measurements from adjacent spatial units may exhibit significant association. This association arises due to the similarity of many underlying factors between adjacent units that influence the response. However, in practice, it is likely that many of these factors are actually unobserved or difficult to quantify or to measure. Hence, the available pool of covariates may not be adequate to capture the association. The role of a spatial stochastic model in this setting is to account for this additional correlation by using a vector of random effects, one for each unit. In a hierarchical setting, this can be achieved by imposing a prior joint distribution on these random effects that allows for correlation based on geographical proximity. Use of a spatial random effect significantly improves the predictive accuracy of the model as it allows for borrowing of information from observed locations to unsampled regions.

There is well-established literature on the hierarchical framework for spatial data collected from adjacent areal units. The most common approach is to introduce an MRF - type joint prior where the conditional distribution of each random effect is dependent only on random effects from areal units pre-classified as *neighbors* - this allows for spatial smoothing. A frequently used class of MRF distribution is a CAR prior that specifies each of these conditional distributions as univariate normal with the mean being average of spatial effects at neighboring units. The CAR prior is a common choice for many Bayesian approaches because (i) it yields a standard, easy-to-sample posterior distribution for simulation of spatial effects using MCMC in linear and generalized linear models and (ii) since the joint distribution is specified through univariate full conditionals, one does not have to deal with high-dimensional matrix computation (as in Gaussian process-based methods). However, with a large number of areal units, estimating these effects one-by-one from conditional distributions becomes computationally intensive; suffers from strong correlation and poor mixing performance.

The key contribution of this chapter is to propose an alternative adjacency-based Bayesian hierarchical model for areal-level datasets using a low-rank approximation of the spatial effects. Such approximations are common in analyzing large point-referenced (Higdon, 1998; Banerjee et al., 2008) or point pattern (Chakraborty et al., 2011) datasets due to their computational efficiency. However, there has been limited development of this kind of approach in the areal framework. Treating the areal units as vertices of a graph, we studied the spectral properties of its Laplacian matrix and established the link between ordering of its eigenvectors and frequency of spatial variation. Subsequently, we proposed to replace the original set of spatial effects with a truncated linear combination of these eigenvectors so that high-frequency spatial patterns are approximated with a white noise. To enhance the flexibility of our approach and to encourage sparsity, we treated the included set of eigenvectors as well as their number as parameters and updated them using a reversible jump Markov chain Monte Carlo (RJMCMC; Richardson and Green, 1997) scheme.

We illustrate the use of our method to address an important topic of ecological research-modeling of species abundance. Our focus is the presence-absence setting where abundance information is aggregated at an areal resolution in terms of raw counts or ordinal categories. The goal of a stochastic model in this setting is to explain how the variation in environmental features influences degree of abundance and to predict the abundance distribution at unsampled part of the region. The presence of spatial random effect in the abundance distribution can account for ecological phenomena such as localized dispersal, as well as influence of omitted/unobserved geographical features with spatial pattern that, at sufficiently high resolution, correlate the abundance of a species at one location with its abundance at neighboring locations (Ver Hoef et al., 2001). As a result, a spatial model improves the predictive quality of the model by borrowing information across locations (Gelfand et al., 2006). Abundance datasets that come from a large study region usually have two common characteristics: (i) a significant part of the region remains unsampled due to difficulty of accessibility but (ii) on the other hand, the accessible part of the region is sampled at a dense resolution. Whereas the latter allows for fine-scale analysis of species-environment relationship, the former implies we need a very large number of spatial random effects to cover the entire study region at that scale. The usual spatial model for such datasets becomes computationally challenging and that is where our method can be established as an efficient alternative.

The organization of this chapter is as follows. In Section 4.2, we developed the main hierarchical model in detail. Parameter estimation from this model using an adaptive MCMC algorithm is mentioned in Section 4.3. Multiple simulation and comparison studies are presented in Section 4.4 followed by the real data application in Section 4.5. We conclude this chapter in Section 4.6 with discussion of possible extensions.

4.2 Method Based on Spectral Structure

The CAR specification, described in Chapter 2.6, is convenient in terms of simple and intuitive posterior full conditional distribution for each spatial effect and is frequently used in areal data modeling. However, it suffers from some important limitations. First, when the number of such areal units is very large, we have to sequentially sample each component of θ at each iteration of the MCMC, which increases the computation time significantly. Second, use of posterior full conditionals introduces strong correlation across successive draws of the MCMC which results in lower effective sample size and/or larger thinning interval to obtain approximately independent samples from posterior distribution, again increasing the computation time. Third, similar to its prior, the posterior joint distribution of θ is improper as well - Banerjee et al. (2014) discusses two possible solutions for this. One could use a ρ-CAR with $\rho < 1$ implying propriety. However, they have shown that a reasonably strong spatial correlation would need ρ to be very close to 1 which would again lead to numerical impropriety. Another commonly used solution is to do an adhoc mean-centering for the θ vector at each iteration, referred to as 'centering on the fly'.

Based on the above discussion, our objective in this chapter is to propose an alternative specification for multivariate prior distribution for θ that is based on neighborhood structure W like CAR but is computationally efficient for large regions. For this purpose, we use the graph Laplacian as described below.

Consider the data structure described in Chapter 2.2. If we think of the areal units as n vertices of a graph G, then W acts as the adjacency matrix of G. ($w_{ij} = 1$ is equivalent to having an undirected edge between A_i and A_j.) Then, the graph laplacian for G is defined to be an $n \times n$ matrix $L = D_w - W$ where D_w is a diagonal matrix containing the number of neighbors for each areal unit: $D_w[i, i] = w_{i+}$.

Characteristics of L matrix are well-explored in literature (Chung, 1997). It is a singular, non-negative definite with number of zero eigenvalues equal to the number of connected components in G. In the following development, we assume a connected graph so L has only one zero eigenvalue with eigenvector 1_n. Our goal is to utilize its properties in the prior distribution for θ. For that, we present the following result. Suppose, f is a function defined on the vertices of G, so we can denote $f_G = \big(f(A_1), f(A_2), \ldots, f(A_n)\big)^T$. Define the total variability of f on G as $\sigma^2_G(f) = \frac{1}{2}\sum_{w_{ij}=1}\big(f(A_i) - f(A_j)\big)^2$, the total squared difference between the values of f at any pair of neighboring units (the factor of 1/2 adjusts for counting any (i, j) pair twice). Then, we show that σ^2_G can be written as a quadratic form in f_G involving the Laplacian L as we proved in the following result.

Result: For a function f defined on a graph G with Laplacian L, $\sigma^2_G(f) = f^T_G L f_G$.

Proof:

$$\begin{aligned}
\sigma^2_G(f) &= \frac{1}{2}\sum_i\sum_j w_{ij}(f(v_i) - f(v_j))^2 \\
&= \frac{1}{2}\Big(\sum_i\sum_j f(v_i)^2 w_{ij} + \sum_i\sum_j f(v_j)^2 w_{ij} - 2\sum_i\sum_j f(v_i)f(v_j)w_{ij}\Big) \\
&= \frac{1}{2}\Big(\sum_i f(v_i)^2 \sum_j w_{ij} + \sum_j f(v_j)^2 \sum_{ij} w_{ij} - 2\sum_i\sum_j f(v_i)f(v_j)w_{ij}\Big) \\
&= \frac{1}{2}\Big(\sum_i f(v_i)^2 w_{i+} + \sum_j f(v_j)^2 w_{j+} - 2\sum_i\sum_j f(v_i)f(v_j)w_{ij}\Big) \\
&= \frac{1}{2}\Big(2\sum_i f(v_i)^2 w_{i+} - 2\sum_i\sum_j f(v_i)f(v_j)w_{ij}\Big) \\
&= \sum_i f(v_i)^2 w_{i+} - \sum_i\sum_j f(v_i)f(v_j)w_{ij} \\
&= f^T_G L f_G
\end{aligned}$$

Let us define the spectral decomposition of L as $L = \sum_{i=1}^n \lambda_{(i)} v_i v_i^T$ with $\lambda_{(i)}$ being the i^{th} smallest eigenvalue of L with corresponding eigenvector v_i (there may be eigenvalues with multiplicity > 1). Using Courant-Fischer Minimax Theorem (Courant and Hilbert, 1965), the above result implies that for σ^2_G to be smallest: without any constraint, f_G needs to a multiple of v_1; constraining f_G to be orthogonal to v_1, it needs to be a multiple of v_2; $\ldots$, constraining f_G to be orthogonal to $v_1, v_2, \ldots, v_{i-1}$, it needs to be a multiple of v_i.

It implies that linear combination of the eigenvectors corresponding to smaller eigenvalues of L generate functions with small σ^2_G - they are mostly similar across different units indicative of strong spatial association. In fact, for a connected graph, the (unnormalized) eigenvector corresponding to the smallest eigenvalue is the vector 1_n, which is a spatially constant function with $\sigma^2_G = 0$. On the other hand, eigenvectors associated to larger eigenvalues of L can represent functions that rapidly change from one unit to its neighboring unit

- functions that behave more like random noise with weak association.

To link these characteristics to the specification of θ, first observe that it can be written as an exact linear combination $\theta = \sum_{m=1}^{n} \eta_m v_m$ for a set of coefficients $\eta_1, \eta_2, \ldots, \eta_n$. When θ shows strong spatial association, it follows from the above discussion that, we expect η_m to be significant for small values of m. For identifiability constraint, we set $\eta_1 = 0$. Then, we propose a low-rank representation of θ as follows:

$$\theta = \sum_{m=2}^{k+1} \eta_m v_m \tag{4.1}$$

The above representation of θ adds new interpretation to the role of the nugget term ϵ. We are expressing a n dimensional random vector using a random linear combination of k fixed eigenvectors with $k \ll n$. In addition to capturing measurement error, ϵ serves two purposes here. First, since we are excluding $v_{k+2}, v_{k+3}, ..., v_n$ from the expansion of θ, we are basically eliminating high-frequency spatial variation from the model. The nugget term is expected to represent the average variability due to those excluded high-frequency components. Second, although the distribution of θ is now rank-deficient, addition of ϵ implies that the response vector z still has a full-rank marginal dispersion as a noisy version of the low-rank spatial structure. The adequacy of above expansion as well as its computational advantage depend heavily on choice of the truncation level k, that we discuss in detail in the following section.

4.3 Adaptive Selection of Eigenvectors

Instead of using a pre-fixed value of number of eigenvectors k in Eq. 4.1, we treat it as a parameter. We allow variability in two directions to encourage a sparse representation of θ if supported by the data. First, instead of using a fixed value of k, we fix an upper bound k_{max}. So, at any stage during MCMC, the number of nonzero η coefficients $= k \leq k_{max}$.

Hence, even if we conservatively set a large value of k_{max}, we allow the data to choose a small number of terms if adequate. Second, at any iteration, the k indices of η coefficients do not need to be consecutive, instead they could be any k numbers from a set $\{2, 3, \ldots, k_0 + 1\}$ where k_0 is another pre-fixed threshold. It implies, the MCMC algorithm can add or delete eigenvectors from the expansion of θ flexibly, depending on the evidence from data. Using smaller values of k_0 or k_{max} allow us to focus on simpler models and prevents overfitting.

Letting k to vary will imply that the MCMC will move between parameters sets of varying dimensions. To achieve this, we employ the RJMCMC scheme of Richardson and Green (1997). Below, we develop the algorithm when z is Gaussian. For any non-Gaussian z following a generalized linear model, one can replicate the same technique on some appropriate parameter from the distribution of z. Further note that, for the Gaussian case, one should not use a noise term directly in the distribution of z as the expansion of θ in Eq. 4.1 already contains a nugget.

Suppose, at a particular iteration of MCMC, the nonzero coefficients in the expansion of θ are $\eta_{s_1}, \eta_{s_2} \ldots \eta_{s_k}$ where $k \leq k_{max}$ and $2 \leq s_i \leq k_0+1$. If we define an $n \times (p+k)$ dimensional matrix $X_k = [X \; v_{s_1} \; v_{s_1} \cdots v_{s_k}]$ and another $(p + k)$ dimensional vector $\tilde{\beta}_k = (\beta_1, \beta_2, \ldots, \beta_p, \eta_{s_1} \ldots, \eta_{s_k})^T$, then it follows from above that $z = (z_1, z_2, \ldots, z_n)^T \sim MVN(X_k\tilde{\beta}_k, \sigma^2 I_n)$. To make the dimension-changing move, we first marginalize out $\tilde{\beta}$ and σ^2 from the model. Thus, to compute the new likelihood for a proposed move, it suffices to know how the X_k matrix is going to change under the proposal without the need to propose coherent updates to $\tilde{\beta}_k$ and σ^2. This is expected to improve the performance of RJMCMC as the moves are accepted or rejected based on increase in marginal likelihood of z. The marginal likelihood computation, performed under a multivariate normal prior for $\tilde{\beta}_k$ and an IG prior for σ^2, is shown below. We start with the hierarchical structure

$$z \sim MVN(X_k\tilde{\beta}_k, \sigma^2 I_n), \;\; \tilde{\beta}_k \sim MVN_{p+q}(\tilde{\beta}_{k_0}, \sigma^2\tau^2 I_{p+k}), \;\; \sigma^2 \sim IG(a_1, b_1)$$

Then, the marginal density of z is calculated as

$$\begin{aligned}
f(z) &= \iint f(z|\tilde{\beta}, \sigma^2)\pi(\tilde{\beta}, \sigma^2) d\tilde{\beta} d\sigma^2 \\
&= \iint f(z|\tilde{\beta}_k, \sigma^2)\pi(\tilde{\beta}_k|\sigma^2)\pi(\sigma^2) d\sigma^2 d\tilde{\beta}_k \\
&\propto \iint (\sigma^2)^{\frac{-n}{2}} \exp\left\{\frac{-(z - X_k\tilde{\beta}_k)^T(z - X_k\tilde{\beta}_k)}{2\sigma^2}\right\} (\sigma^2\tau^2)^{\frac{-(k+p)}{2}} \\
&\quad \times \exp\left\{\frac{-(\tilde{\beta}_k - \tilde{\beta}_{k_0})^T(\tilde{\beta}_k - \tilde{\beta}_{k_0})}{2\sigma^2\tau^2}\right\} (\sigma^2)^{-(a_1+1)} \exp\left\{\frac{-n}{\sigma^2}\right\} d\sigma^2 d\tilde{\beta}_k \\
&= \iint (\sigma^2)^{-(a_1+\frac{n}{2}+\frac{k+p}{2}+1)} (\tau^2)^{\frac{-(k+p)}{2}} \exp\left\{\frac{-(z - X_k\tilde{\beta}_k)^T(z - X_k\tilde{\beta}_k)}{2\sigma^2}\right\} \\
&\quad \times \exp\left\{\frac{-(\tilde{\beta}_k - \tilde{\beta}_{k_0})^T(\tilde{\beta}_k - \tilde{\beta}_{k_0})}{2\sigma^2\tau^2}\right\} \exp\left\{\frac{-b_1}{\sigma^2}\right\} d\sigma^2 d\tilde{\beta}_k
\end{aligned}$$

We can identify the part involving σ^2 as another IG density:

$$IG\left(a_1 + \frac{n}{2} + \frac{k+p}{2}, b_1 + \frac{(z - X_k\tilde{\beta}_k)^T(z - X_k\tilde{\beta}_k)}{2} + \frac{(\tilde{\beta}_k - \tilde{\beta}_{k_0})^T(\tilde{\beta}_k - \tilde{\beta}_{k_0})}{2\tau^2}\right)$$

Its normalizing constant will be:

$$\left[b_1 + \frac{(z - X_k\tilde{\beta}_k)^T(z - X_k\tilde{\beta}_k)}{2} + \frac{(\tilde{\beta}_k - \tilde{\beta}_{k_0})^T(\tilde{\beta}_k - \tilde{\beta}_{k_0})}{2\tau^2}\right]^{-(a_1+\frac{n+k+p}{2})}$$

Hence the above integral is:

$$\begin{aligned}
&= (\tau^2)^{\frac{-(k+p)}{2}} \int \left[b_1 + \frac{(z - X_k\tilde{\beta}_k)^T(z - X_k\tilde{\beta}_k)}{2} + \frac{(\tilde{\beta}_k - \tilde{\beta}_{k_0})^T(\tilde{\beta}_k - \tilde{\beta}_{k_0})}{2\tau^2}\right]^{-(a_1+\frac{n+k+p}{2})} d\tilde{\beta}_k \\
&= (\tau^2)^{\frac{-(k+p)}{2}} \int \left[\tilde{\beta}_k^T A\tilde{\beta}_k - 2\tilde{\beta}_k^T B + C\right]^{-(a_1+\frac{n+k+p}{2})} d\tilde{\beta}_k
\end{aligned}$$

where: $A = \frac{1}{2}(X_k^T X_k + \frac{I_{k+p}}{\tau^2})$, $B = \frac{1}{2}(X_k^T z + \frac{\tilde{\beta}_{k0}}{\tau^2}))$, $C = b_1 + \frac{z^T z}{2} + \frac{\tilde{\beta}_{k0}^T \tilde{\beta}_k}{2\tau^2}$

$$= \quad (\tau^2)^{\frac{-(k+p)}{2}} \int \left[(\tilde{\beta}_k - A^{-1}B)^T A(\tilde{\beta}_k - A^{-1}B) + (C - B^T A^{-1}B)\right]^{-(a_1 + \frac{n+k+p}{2})} d\tilde{\beta}_k$$

Denote: $\tilde{\mu} = A^{-1}B$, $\tilde{\Sigma} = A^{-1}$, $\tilde{c} = C - B^T A^{-1}B$. Then, the integral becomes

$$\begin{aligned} &= \quad (\tau^2)^{\frac{-(k+p)}{2}} \int \left[\tilde{c} + (\tilde{\beta}_k - \tilde{\mu})^T \tilde{\Sigma}^{-1}(\tilde{\beta}_k - \tilde{\mu})\right]^{-(a_1 + \frac{n+k+p}{2})} d\tilde{\beta}_k \\ &= \quad (\tau^2)^{\frac{-(k+p)}{2}} \tilde{c}^{-(a_1 + \frac{n+k+p}{2})} \int \left[1 + (\tilde{\beta}_k - \tilde{\mu})^T \frac{\tilde{\Sigma}^{-1}}{\tilde{c}}(\tilde{\beta}_k - \tilde{\mu})\right]^{-(a_1 + \frac{n+k+p}{2})} d\tilde{\beta}_k \end{aligned}$$

Let us define $d = 2a_1 + n$. Then the integral becomes:

$$= \quad (\tau^2)^{\frac{-(k+p)}{2}} \tilde{c}^{-(\frac{d+k+p}{2})} \int \left[1 + \frac{1}{d}(\tilde{\beta}_k - \tilde{\mu})^T (\frac{\tilde{c}\tilde{\Sigma}}{d})^{-1}(\tilde{\beta}_k - \tilde{\mu})\right]^{-(\frac{d+k+p}{2})} d\tilde{\beta}_k$$

The quantity inside the integral is proportional to the density function of a $(k+p)$-variate t random variable with $df = d$. So the integral will be its normalizing constant

$$(\tau^2)^{\frac{-(k+p)}{2}} \tilde{c}^{\frac{-(d+k+p)}{2}} |\tilde{c}\tilde{\Sigma}|^{\frac{1}{2}} \; = \; (\tau^2)^{\frac{-(k+p)}{2}} \tilde{c}^{\frac{-d}{2}} |\tilde{\Sigma}|^{\frac{1}{2}}$$

It is evident from the above derivation that the marginal likelihood calculation involves working with only $(p+k)$-dimensional matrices which provides a significant computational advantage when $n \gg (p+k)$. Post-reversible jump, conditional on the selected indices, one can update components of $\tilde{\beta}_k$ and σ^2 using standard distributions.

The key parameters in the dimension-changing move are $\{k, s_1, s_2, \ldots, s_k\}$. We propose to move them between two successive iterations in three possible ways:

- *Birth*: propose to increase k to $k+1$ and choose one of the currently excluded indices.

- *Death*: propose to reduce k to $k-1$ and delete one of the k currently included indices.

- *Swap*: keep k unchanged, but replace one of the currently included indices with a currently excluded index.

The prior for k is chosen to be a Poisson distribution truncated at k_{max}. When k reaches k_{max}, we allow only moves of the type (ii) or (iii). Given k, the prior on the indices $(s_1, s_2, \ldots, s_k)$ is taken to be uniform over all k-length subsets of $\{2, \ldots, k_0+1\}$. Alternatively, one may opt for non-uniform probability mass functions to treat larger and smaller indices differently. Below we present the details of RJMCMC for variable-dimension coefficient estimation.

Denote by $\nu_k = \{s_1, s_2, \ldots, s_k\}$ the collection of included indices in the model. Using a suitable proposal distribution, q, propose a dimension changing move $(k, \nu_k) \to (k', \nu_{k'})$. The birth, death and swap imply addition, deletion and replacement of eigenvectors ot at a time. Thus $k' \in \{k-1, k, k+1\}$.
The acceptance ratio for such a move is given by

$$p_{k \to k'} = \min \left\{ 1, \ \frac{f\left(z|k', \nu_{k'}, \ldots\right)}{f(z|k, \nu_k, \ldots)} \ \frac{p\left(\nu_{k'}, k'\right)}{p\left(\nu_k, k\right)} \ \frac{q\left((k', \nu_{k'}) \to (k, \nu_k)\right)}{q\left((k, \nu_k) \to (k', \nu_{k'})\right)} \right\}$$

We specify a prior for (k, ν_k) in the form of $p(\nu_k|k)p(k)$. As mentioned above, k has a Poisson(λ) prior truncated to the right at k_{max}. Conditional on k, prior for ν_k assigns equal probability to any k-length subset of $\{1, 2, \ldots, k_0\}$. Hence we have,

$$\pi(\nu_k|k) = \frac{1}{\binom{k_0}{k}} = \frac{k!(k_0-k)!}{k_0!} \propto k!(k_0-k)!$$

Next, we specify the proposal distribution q for each of the three moves. For a birth move, we need to draw a new index from $(k_0 - k)$ excluded indices. We are going to propose a birth with probability $p_{b,k}$ and, if a birth is proposed, randomly include a specific eigenvector with probability $\frac{1}{k_0-k}$. Hence, for a birth step, $k' = k+1$, and

$$q[(k, \nu_k) \to (k+1, \nu_{k+1})] = p_{b,k} \times \frac{1}{k_0 - k}$$

Second, for a death move, we want to remove one of the k indices currently in the model. We are going to propose a death with probability $p_{d,k}$ and, if we propose the death, then the chance that we remove a particular eigenvector is $\frac{1}{k}$. Hence, for a death step, $k' = k-1$, and

$$q[(k, \nu_k) \to (k-1, \nu_{k-1})] = p_{d,k} \times \frac{1}{k}$$

Finally, we may want to keep k eigenvectors, but replace one of the existing eigenvectors with a new one. We are going to propose it with probability $p_{s,k} = 1 - p_{b,k} - p_{d,k}$, and if we propose the swap, then the chance that we delete and include a specific pair of eigenvectors is $\frac{1}{k(k_0-k)}$. Hence, for a swap step, $k' = k$, ν'_k denotes the updated set of k indices and

$$q[(k, \nu_k) \to (k, \nu'_k)] = p_{s,k} \times \frac{1}{k(k_0 - k)}$$

The acceptance ratios for different types of move can be worked out from the above prior and proposal distributions. Set $k = k', \nu_k = \nu_{k'}$ if the move is accepted, leave unchanged otherwise. Subsequently, $\tilde{\beta}_k$ can be updated using the $(k+p)$-variate t distribution with degrees of freedom d, mean $\tilde{\mu}$, dispersion $\frac{\tilde{c}\tilde{\Sigma}}{d}$ whose expressions are derived above.

We note that, to have $v_2, ..., v_{k_0}$ available, one needs to compute the k_0 smallest eigenvalues of L and their associated eigenvectors. However, this computation needs to be done only once before the MCMC (Section 4.6 mentions a setting where this is not true). When n very large, it takes significant time to compute the entire spectral decomposition to determine the k_0 smallest eigenvalues. In that case, one may use a more efficient algorithm as outlined in Chapter 3. We present a simulation study in the following section.

4.4 Simulation Studies

We conduct multiple simulation studies to for assessing computational and predictive properties of proposed method. In the following, we first explore variation in computational time in obtaining the eigenvectors for different choices of n, k and sparsity of W. Next, we present predictive and computational comparison of the proposed approach against CAR.

4.4.1 Simulation Study I

To find out the k smallest eigenvalues of the graph-Laplacian L in this work, we used the IRL-LSEFE with shift-and-invert mode (Chapter 3). Here, instead of finding eigenvectors corresponding to k smallest eigenvalues of L, we (equivalently) find eigenvectors that correspond to k largest eigenvalues of $(L - aI)^{-1}$ where a is a fixed real number of our choice. We chose $a = -0.01$ since all eigenvalues of L are non-negative. The reason for adopting this approach is the fact that the Lanczos algorithm converges faster to the eigenvalues with largest magnitude. Note that, we cannot choose $a = 0$ as L is singular.

Now, we present simulation studies that numerically illustrate the benefit of abovementioned approach with respect to much-improved computation time. In each case, we define a regular grid with large number of cells, define a neighborhood structure with predefined distance threshold, compute the Laplacian and find its k smallest eigenvectors. In Table 4.1, we compare the IRL-LSEFE algorithm against the general method of doing a complete eigen decomposition as the graph size increases.

Table 4.1: Computational time (mm:ss) for IRL-LSFLE and QR method for 100 smallest eigenpairs

n	5000		15000		25000	
# of neighbors	8	20	8	20	8	20
IRL-LSFLE	0:01	0:01	0:01	0:01	0:02	0:03
QR method	2:42	2:42	9:43	69:44	23:16	323:36

In Table 4.2, we explore the computational sensitivity of IRL-LSEFE approach to the grid size as well as sparsity. If we increases the distance threshold for defining neighbors, that reduces the sparsity of the graph structure. Here, we analyzed four different grid sizes (substantially larger than those in Table 4.1), each with two different sparsity levels.

Table 4.2: IRL-LSFLE computation times (mm:ss) for variation in n, k and sparsity of L

n		25000		50000		100000		200000	
# of neighbors		8	20	8	20	8	20	8	20
k	50	0:02	0:03	0:04	0:05	0:06	0:14	0:16	0:29
	100	0:02	0:03	0:05	0:07	0:09	0:14	0:28	0:46
	500	1:46	1:54	3:30	3:51	6:37	7:16	12:33	13:59
	1000	7:20	7:34	14:01	14:40	25:45	27:01	44:11	50:52
	2000	33:24	33:53	59:35	60:35	104:59	107:15	190:06	196:40

As we have expected, computation time increases if we make one or more of the following changes: (i) increasing the grid size, (ii) collecting a larger number of eigenvectors, (iii) reducing the sparsity of the graph by using a larger threshold for defining neighbors.

4.4.2 Simulation Study II

We focus on comparing the proposed adaptive approach against the CAR-based spatial regression with respect to predictive accuracy as well as computational efficiency. Sensitivity of this comparison to variation in size of spatial network as well as density of neighborhood structure will also be investigated. We begin by dividing the unit-square into identical gridcells of dimension 100×100 and 150×150. Next, we consider a spatially-varying function f such that,

$$f(s) = 2s_1^2 \log(1 + s_2) - 3\frac{s_1}{1 + s_2^2} + 3\cos 2\pi s_1, \;\; s = (s_1, s_2) \in [0,1] \times [0,1].$$

For any cell, the response value is constructed by evaluating f at its center point and adding a Gaussian noise with standard deviation $= 0.5$. Hence, we obtain two spatial datasets with a total of $10,000$ and $22,500$ areal units corresponding to the first and second grid, respectively. To construct the adjacency matrix W, we first define neighbors based on distance between two cell centers being less than a threshold. We propose two different values of threshold - resulting in about 8 and 25 neighbors, respectively, for majority of the cells. Hence, sparsity of both L and W matrices differ significantly between these two settings.

For validation of predictive performance, we randomly selected 10% of the observations as a test set and fit both models on the remaining observations. The CAR model was estimated using the CARBayes (Lee, 2013) R package. Post-MCMC, we simulate from posterior predictive distribution for each of the test observations and validate the accuracy using three measures: (i) absolute bias computed as the difference between the true response and its posterior median, (ii) uncertainty measured as the width of two-sided 90% credible interval (C.I.) and (iii) empirical coverage calculated as proportion of responses in the test set that lie inside their respective 90% C.I. For evaluating computational performance, we chose two measures. The first one is the time required for MCMC estimation. We ran the algorithm for 20,000 iterations discarding initial 10,000 draws and thinning the rest at every 5^{th} iteration. The second one is effective sample size that is calculated by taking into account the autocorrelation within MCMC sample. Low value of effective sample size relative to number of posterior draws indicates high autocorrelation and poor mixing. We calculated the effective sample size (ESS) from posterior samples of log likelihood of both models using coda (Plummer et al., 2006) R package and report it as proportion of number of MCMC draws retained after burn-in and thinning. We replicate this procedure for 10 different choices of the test set and, in Table 4.3, summarize these measures across replications for both models.

Table 4.3: Comparison of predictive and computational performance

Criterion	Absolute Bias	90% C.I. Width	Empirical Coverage	Duration (in minutes)	ESS Proportion	Effective Time (in minutes)
	Grid size: 100 × 100 , Avg. no of neighbors: ∼ 8					
CAR prior	0.412	1.693	0.893	4	0.163	25
Adaptive Bayes	0.392	1.785	0.929	6	0.425	14
	Grid size: 100 × 100, Avg. no of neighbors: ∼ 25					
CAR prior	0.406	1.719	0.904	14	0.005	2800
Adaptive Bayes	0.397	1.803	0.926	9	0.296	30
	Grid size: 150 × 150 , Avg. no of neighbors: ∼ 8					
CAR prior	0.406	1.670	0.899	20	0.197	102
Adaptive Bayes	0.398	1.795	0.926	20	0.734	27
	Grid size: 150 × 150 , Avg. no of neighbors: ∼ 25					
CAR prior	0.410	1.684	0.897	105	0.003	35000
Adaptive Bayes	0.399	1.800	0.929	24	0.622	39

Note: Duration refers to time for the full MCMC with 20,000 iterations. However, ESS proportion is calculated only using 2,000 draws retained after burn-in and thinning described above.

In terms of predictive accuracy, the two models performed comparably. The proposed approach produces marginally smaller bias than CAR model. Whereas both models have the coverage rate near or above the desired level of 90%, the adaptive method consistently covered about 2% more test observations due to slightly wider credible intervals resulting in larger uncertainty. Comparison of MCMC characteristics reflects a more contrasting behavior. When the grid was sparse with about 8 neighbors on an average, the CAR model took either less or as much as time compared to the proposed approach for same number of iterations. However, if we look at corresponding ESS proportions for log-likelihood under two models, it reveals that the latter has about 2.5–5 times more effective samples than CAR. It implies that to get similar effective sample size, the MCMC for CAR model needs to be run with approximately 2.5–5 times more iterations than our method which significantly skews the computational advantage. For the dense grid, the skew is stronger and clearly visible. The duration of CAR estimation was more than 1.5 and 4 times the duration for proposed approach fr the smaller and larger grids, respectively. However, the correlation and lack-of-mixing problem was acute for CAR resulting in very low effective sample size whereas the proposed approach could register between 30 to 60 percent ESS. We tried four different choices for $k_{max} = 20, 40, 60, 80$ and all of them exhibited very comparable predictive performance and similar computation time. To avoid repetition, we include only the results corresponding to $k_{max} = 60$ as that had (very marginally) smaller estimates of absolute bias and uncertainty compared to the rest. The only notable quantity was ESS that showed minor variation between choices for k_{max} due to difference in size of allowed model parameters. For a practical application, the strategy of choosing the value of k_{max} is discussed in detail in Section 4.5.

One interesting and practically useful feature of the proposed approach is that when grid size gets large, its computational efficiency increases sharply unlike CAR. For the smaller grid, it has an ESS of 42% and 30% which increases to 73% and 62%, respectively for the larger grid. This arises from the fact that the total number of parameters in the model is only $(k + p)$ and is not connected to n. As a result, with increase in n, correlation between parameters estimates successive iterations reduces significantly. For the same reason, the computation time increases marginally when the grid size doubles in a major contrast to CAR. These features, coupled with satisfactory predictive performance across grid sizes, makes it an appropriate choice for modeling spatial datasets on very large grids as in the following application.

4.5 Model for Plant Abundance

To illustrate the real-world application of our methodology, we chose a plant abundance dataset from Cape Floristic Region (CFR) of South Africa. This region, globally known as a biodiversity hotspot, encompasses an area of $90,000\ km^2$ and includes about $9,000$ plant species. For our analysis, we use the Protea Atlas Dataset of Rebelo (2002), collected beginning in 1991 as part of a 10-year project to document the distribution of Proteaceae, the flagship family in Southern Africa. Data were collected at "record localities": relatively uniform, geo-referenced areas typically 50 to 100 m in diameter. Across the region, abundance (including absence) has been recorded at around $60,000$ such sites. We use the abundance information in categorized form: category 0: none observed, category 1: 1–10 observed, category 2: 11–100 observed, category 3: >100 observed. Regarding covariate information, Gelfand et al. (2006) has studied data on 16 environmental and soil-type variables, available at a resolution of $1.55 \times 1.85\ km^2$ grid cells and finally selected six most significant variables: (APAN.MEAN), July (winter) minimum temperature (MIN07), January (summer) maximum temperature (MAX01), mean annual precipitation (MEAN.AN.PR), summer soil moisture days (SUMSMD), and soil fertility (FERT1). We choose to work with these six covariates as well. At this resolution, it takes $36,907$ grid cells to cover the entire CFR. It was observed that, the sampled sites fall inside only $10,158$ or 28% of these cells as shown in Figure 4.1. Since a large number of cells have no abundance information from the data, the importance of spatial model will be to create species-wise abundance maps for the entire CFR.

In the following, we develop a hierarchical two-stage spatial model that can be applied to any ordinal abundance dataset. Two-stage spatial models for CFR datasets previously appeared in Gelfand et al. (2006) and Chakraborty et al. (2010). However, the motivation and specification for the two-stage model development in our work is completely different. More specifically, the former dealt with a binary presence-absence model and their motivation for two-stage was to distinguish suitability and availability of a species. The latter worked in an ordinal abundance setting like ours, but their two-stage model was based on a ‘potential’ and an ’observed’ abundance. In our approach, we decompose abundance in two parts - first a model for presence-absence (PA) and then, another model for exact abundance category, conditional on presence (CA). Hence, the first stage model has binary response whereas the stage 2 model has ordinal response taking values $1, 2$ and 3. Since many species in CFR has an excess proportion of absences in the sampled sites, the motivation behind this decomposition is to keep the parameters describing occurrence of zeros (absences) separate from the parameters that distinguish between low and high categories of abundance. A similar idea, in a count data setting, has been used in the hurdle model, described in Ridout et al. (1998). Another important characteristic of CFR data is transformation of available land for species growth due to human intervention and/or alien plant infestation. A map of proportion of available, untransformed land across CFR is shown in Figure 4.1. Modeling-wise this implies, even if a location is environmentally suitable for a species, it may be absent there due to land transformation. In our setting, this is a contributing factor in stage 1 but does not appear in CA model of stage 2.

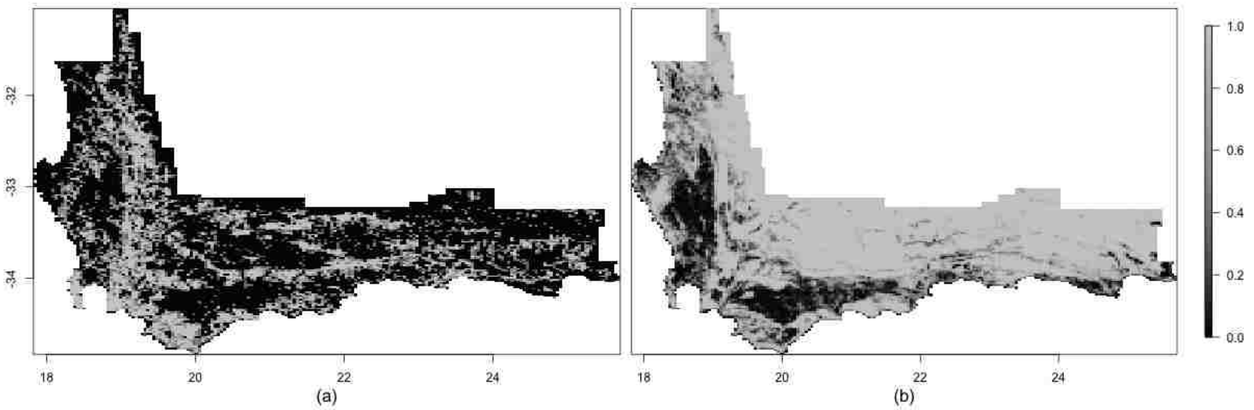


Figure 4.1: (Left) Cells within the CFR that have at least one observation from the Protea Atlas dataset are shown in light grey, while cells with no observations are shown in dark grey. (Right) Proportion of untransformed land inside the CFR. Most of the transformation is due to agriculture, but includes dense stands of alien invasive species.

We develop the hierarchical model below. The dataset includes: (i) the response $z_{ij} \in \{0, 1, 2, 3\}$ denoting the observed category of abundance observed at site j within cell i and (ii) the covariate $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$ representing the environmental and soil-related features (and intercept) from cell i. As mentioned above, the first-stage model is intended to explain whether $z_{ij} = 0$ or > 0. Given $z_{ij} > 0$, the second-stage model will determine the conditional probability distribution of each non-zero abundance category: $P(z_{ij} = h | z_{ij} > 0)$, $h = 1, 2, 3$. Following Albert and Chib (1993), we augment two latent continuous variables y^{PA} and y^{CA} as follows: (i) y^{PA} is the latent presence-absence surface that explains whether the environmental factors at a specific site are suitable for the species to be present there (in any category), (ii) y^{CA} is the latent conditional abundance surface that indicates, given there is a presence, what the specific category of abundance is. A larger value of y^{CA} makes a higher category of abundance more likely. If we assume Gaussian distributions for y^{PA} and y^{CA}, we obtain a binary probit model for stage 1 and an ordinal probit model for stage 2.

We note that z_{ij} can be zero in two ways: (i) if site j is transformed and not available anymore as a plant habitat; or (ii) if site j is not-transformed but y^{PA} is low. If (i) is the case, value of y^{PA} becomes irrelevant. Since u_i the proportion of land transformed within cell i, we can write the PA model as:

$$1\left[z_{ij} > 0\right] \overset{ind}{\sim} u_i\delta_0 + (1-u_i)1\left[y_{ij}^{\mathrm{PA}} > 0\right],\ j = 1, 2, \ldots, n_i,\ i = 1, 2, \ldots, n.$$

The above specification is a two-component mixture with the first component accounting for absence due to land transformation and the second component accounting for presence (absence) at an untransformed site due to high (low) values of latent continuous PA surface. Subsequently, we write the CA model as:

$$z_{ij}|z_{ij} > 0 = \sum_{h=1}^{3} h\, 1\left[\alpha_{h-1} < y_{ij}^{\mathrm{CA}} < \alpha_h\right],\ h = 1, 2, 3,$$

where $-\infty = \alpha_0 < \alpha_1 < \alpha_2 < \alpha_3 = \infty$ are the cut points on the y^{CA} scale that determines the exact abundance category of z_{ij}. We model the distributions of y^{PA} and y^{CA} independently as:

$$\begin{aligned} y_{ij}^{\mathrm{PA}} &= x_i^T\beta^{\mathrm{PA}} + \theta_i^{\mathrm{PA}} + \epsilon_{ij}^{(1)},\ \ \epsilon^{(1)} \overset{\text{iid}}{\sim} N(0,1) \\ y_{ij}^{\mathrm{CA}} &= x_i^T\beta^{\mathrm{CA}} + \theta_i^{\mathrm{CA}} + \epsilon_{ij}^{(2)},\ \ \epsilon^{(2)} \overset{\text{iid}}{\sim} N(0,1) \end{aligned}$$

As usual for Bayesian probit regression, we need to set $\alpha_1 = 0$ and variance of all $\epsilon^{(1)}$ and $\epsilon^{(2)}$ terms to be 1 for identifiability reasons. θ^{PA} and θ^{CA} are spatial random effects for presence-absence and conditional abundance respectively. We model them using truncated eigenvector expansions as:

$$\theta^{\mathrm{PA}} = \sum_{m=1}^{k_1} \eta_m^{PA} v_{s_{m,1}},\ \theta^{\mathrm{CA}} = \sum_{m=1}^{k_2} \eta_m^{CA} v_{s_{m,2}}$$

Note the flexibility in this specification as it allows the spatial effects for presence-absence and conditional abundance to use two different sets (of potentially different sizes k_1 and k_2)

of eigenvectors based on index sets $\{s_{m,1} : m = 1, 2, ...k_1\}$ and $\{s_{m,2} : m = 1, 2, ...k_2\}$, respectively. Hence, they can exhibit different behavior in terms of strength of spatial association as well as sparsity of eigen-expansion. It is also possible to use separate k_{max} values for these two models, as shown in the following analysis.

We turn to prior specifications. There are four sets of regression coefficients $\beta^{\text{PA}}, \beta^{\text{CA}}$, η^{PA} and η^{CA}. The first two have p components each. The dimension of the latter two vectors are fixed conditional on values of k_1 and k_2. To maintain conjugacy, we assign multivariate normal prior distributions with zero mean and a diagonal dispersion matrix with very large variances to all of them. α_2, the only free cutoff parameter for z^{CA}, is assigned an improper uniform prior on all positive real numbers. To control the number of eigenvectors in the expansion for spatial effects, we assign truncated Poisson priors to each of k_1 and k_2. $k_i \overset{ind}{\sim} \text{Poi}(\lambda_i)1[k_i \leq k_{max}]$ for $i = 1, 2$. We summarize the prior specification below:

$$\beta^{\text{PA}},\, \beta^{\text{CA}} \sim \text{MVN}_p\left(0_p, \tau^2 I_p\right),\ \eta^{PA}|k_1 \sim \text{MVN}_{k_1}\left(0_{k_1}, \tau^2 I_{k_1}\right),\ \eta^{CA}|k_2 \sim \text{MVN}_{k_2}\left(0_{k_2}, \tau^2 I_{k_2}\right),$$
$$\pi(\alpha_2) = 1\left[\alpha_2 > 0\right],\ \pi(k_i) = \text{Poi}\left(\lambda_i\right) 1[k_i < k_{max}], i = 1, 2.$$

Given posterior draws of all model parameters, we can construct the empirical posterior distribution for category-specific marginal abundance probabilities (integrating out y^{PA} and y^{CA}) at all cells using following formula:

$$\begin{aligned} P\big[z_{ij} = 0\big] &= P\big[\text{Transformation}\big] + P\big[\text{Absence}\big|\text{Non-transformation}\big]\ P\big[\text{Non-transformation}\big] \\ &= u_i + P\big[\text{Absence}\big|\text{Non-transformation}\big]\ (1 - u_i) \\ &= u_i + \Phi\Big(-x_i^T\beta^{\text{PA}} - \theta_i^{\text{PA}}\Big)(1 - u_i) \end{aligned} \tag{4.2}$$

For category, $h = 1, 2, 3,$

$$\begin{aligned} P\big[z_{ij} = h\big] &= P\big[h\big|\text{Presence}\big]\, P\big[\text{Presence}\big|\text{Non-transformation}\big]\, P\big[\text{Non-transformation}\big] \\ &= P\big[h\big|\text{Presence}\big]\, P\big[\text{Presence}\big|\text{Non-transformation}\big]\, (1 - u_i) \\ &= \Phi\Big(\big[\alpha_{h-1}, \alpha_h\big]\big|(x_i^T\beta^{\text{CA}} + \theta_i^{\text{CA}}), 1\Big)\, \Phi\Big(x_i^T\beta^{\text{PA}} + \theta_i^{\text{PA}}\Big)(1 - u_i) \end{aligned} \tag{4.3}$$

We perform the parameter estimation from the above model using the MCMC scheme described in Section 4.3. Note that, since the model for z is non-Gaussian, the RJMCMC will use the latent continuous pseudo-response y instead of z. The marginal likelihood will be simpler as the variance parameter is fixed at 1. The posterior update of y can be done independently across (i, j) from its prior distribution truncated within the region controlled by corresponding z-category. Posterior simulation of y^{PA} is more non-trivial than simulation of y^{CA} due to presence of land transformation. Whereas the latter is a truncated normal, the former is a mixture of normal and truncated normal distributions. Also note that, since the PA and CA surfaces do not share any common parameters, the MCMC for them can be run in parallel and outputs can be pooled together to construct the marginal probability surface for abundance category as shown in Eq. 4.2 and Eq. 4.3.

For analysis, we chose datasets corresponding to three different species: Protea cynaroides (PRCYNA), Protea repens (PRREPE) and Protea punctata (PRPUNC). The neighborhood matrix W was created using a distance threshold such that most of the cells (except the ones on or near the boundary) have 8 neighbors. Presence of PRCYNA, PRREPE and PRPUNC were observed in 1584, 3831 and 632 cells, respectively. Among the sites where they were present, category 2 was most frequently observed for all three species. Category 3 was least frequent for PRCYNA whereas category 1 was the rarest for the other two species. Working with $k_0 = 100$ eigenvectors, we explored four choices of $k_{max} = 30, 60, 90$ and 100. For each run, we look at the posterior histogram of k in PA and CA models, as shown in

Figures 4.2 – 4.4. Our goal is to choose the smallest k_{max} for which the histogram contains both tails of the posterior distribution because that is indicative of the fact that increasing k_{max} further would not change the behavior significantly. Based on this criterion, we found that, for PA model, we should choose k_{max} to be 90 for PRPUNC and PRCYNA and 100 for PRREPE. The CA models usually require much smaller number of eigenvectors. Setting k_{max} at 60 suffices for PRPUNC and PRCYNA. For PRREPE, the histogram stabilizes at $k_{max} = 30$. We use these values to conduct further spatial analysis for these species.

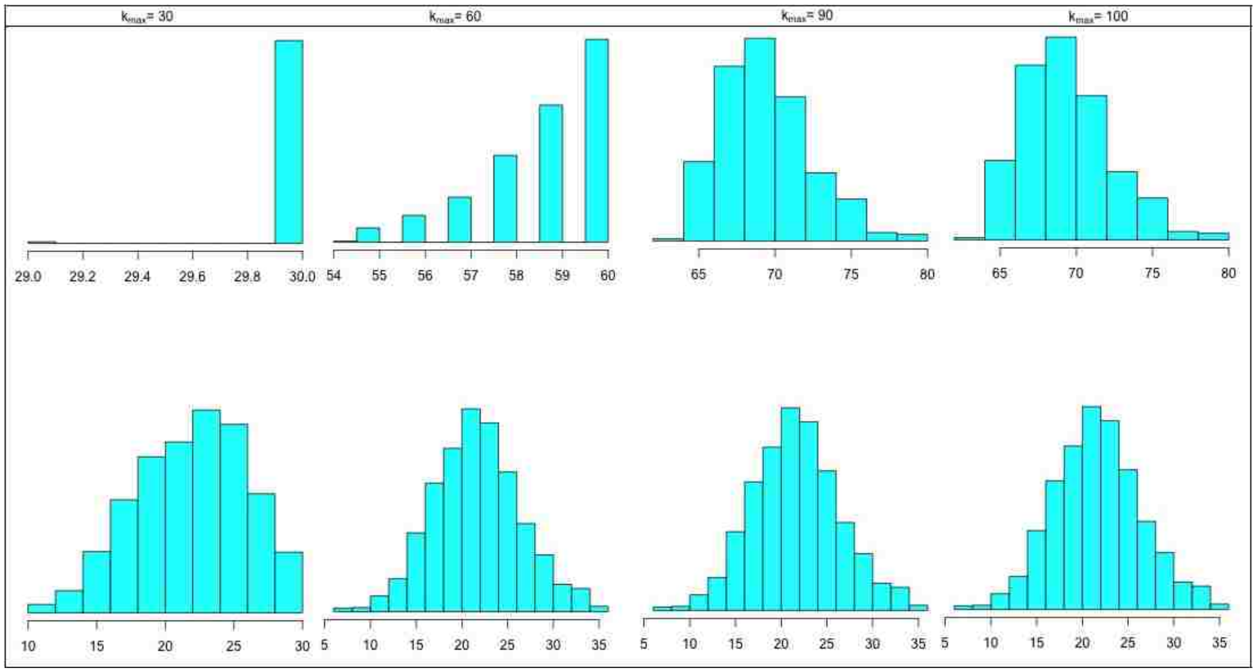


Figure 4.2: PRCYNA: Posterior histogram of number of Laplacian eigenvectors chosen by (top) PA model and (bottom) CA model for different choices of k_{max} parameter

In Table 4.4, we show the significance of covariate effects for each species at 90% level in the PA model. For the CA model, the covariate effects were found to be insignificant except SMDSUM which has a positive effect on increasing abundance of PRPUNC. The most important output of this analysis are the posterior probability maps for four abundance categories for each species based on Eq. 4.2 and Eq. 4.3. They represent the spatial variation in abundance distribution under land transformation. We also show maps without

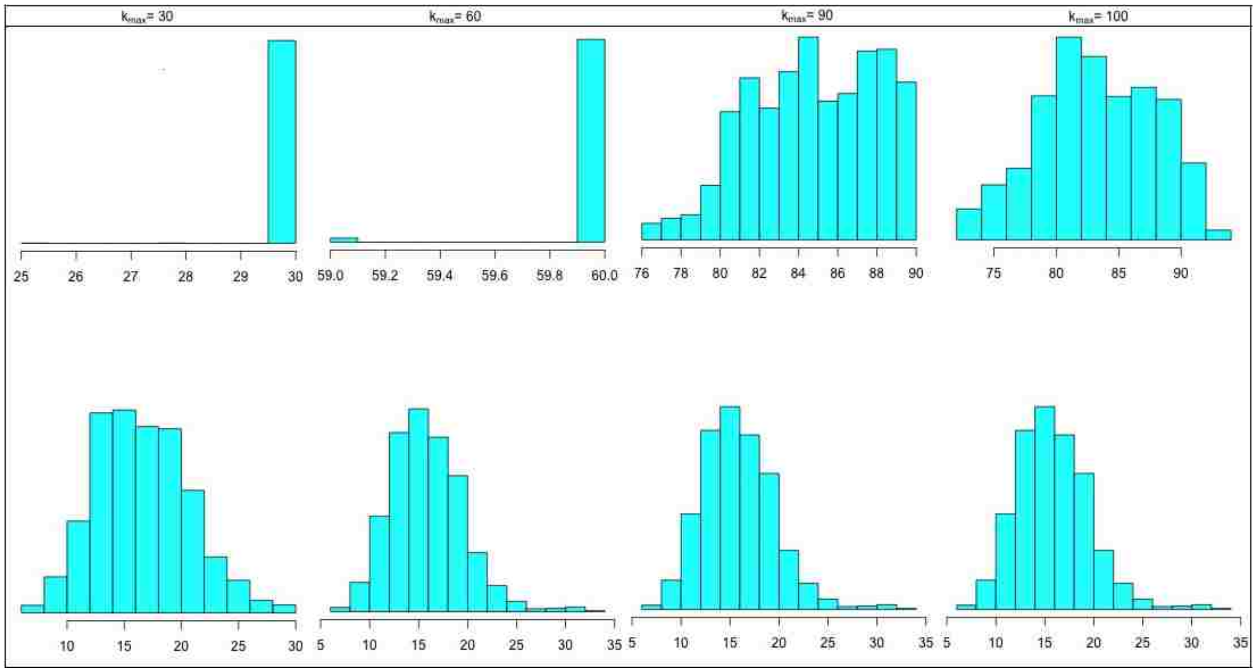


Figure 4.3: PRREPE: Posterior histogram of number of Laplacian eigenvectors chosen by (top) PA model and (bottom) CA model for different choices of k_{max} parameter

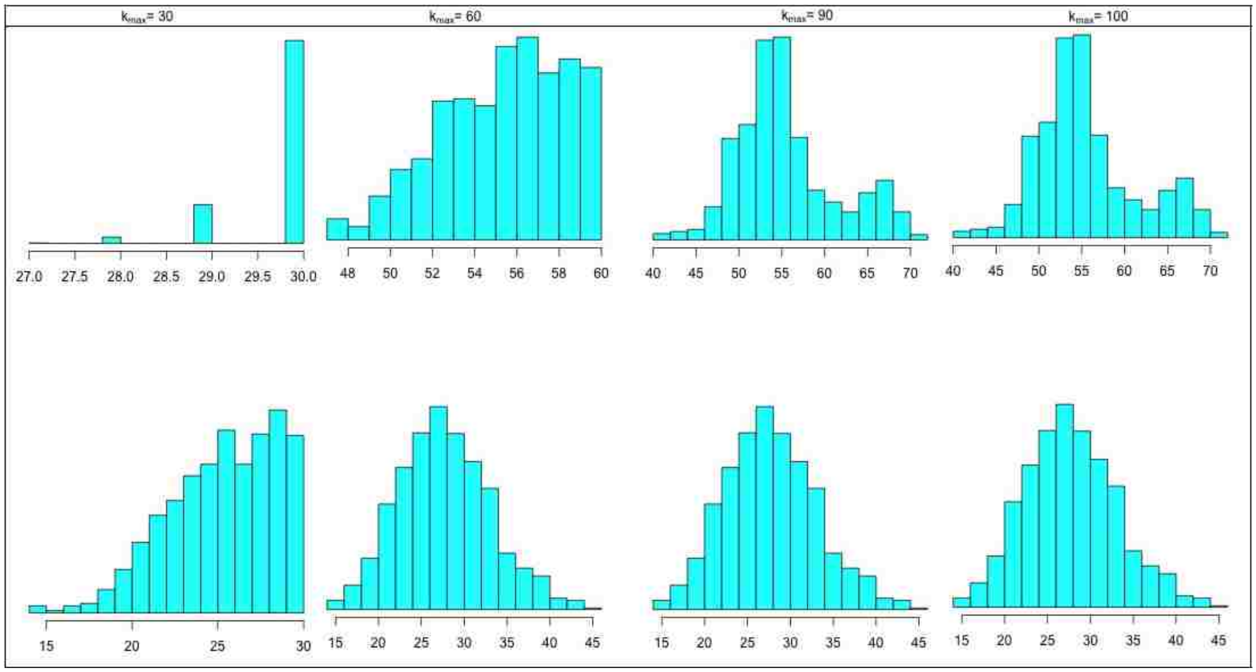


Figure 4.4: PRPUNC: Posterior histogram of number of Laplacian eigenvectors chosen by (top) PA model and (bottom) CA model for different choices of k_{max} parameter

considering any land transformation by setting $u_i = 0$ for all cells in CFR. These represent the abundance distribution under the hypothetical scenario where the entire region would be available for growth of species. Thus, comparing the two abundance maps category-by-category will help us understand how land transformation is influencing the prevalence of a species across the region. The maps are presented in Figures 4.5 – 4.7.

Table 4.4: Significance of covariate effects on presence-absence of a species

Species	APAN.MEAN.	MAX01	MIN07	MEAN.AN.PR	SMDSUM	FERT1
PRCYNA	−	+	−	−	+	+
PRREPE	+	−	o	+	−	+
PRPUNC	+	−	−	+	+	o

Note: +: Positive Significant, −: Negative Significant, o: Insignificant, Level of Significance: 90%

As PRCYNA was observed to present at smaller number of cells than PRREPE, its probability distribution puts very high weight on category 0 (absence) for most of the cells than the latter. This is more evident for PRPUNC which was seen in even fewer cells. However, the regions of positive abundance are mostly separate for PRCYNA and PRPUNC. PRREPE, the most prevalent species, is abundance through out entire CFR except for the Northwestern stretch. Where they were present, the category 2 abundance was observed to be most likely for all three species. Category 3 abundance has the lowest probabilities for PRCYNA whereas Category 1 was least likely for other two species. These findings coincide with the empirical features of the abundance data as discussed above. With respect to overlapping of habitat, PRCYNA and PRPUNC were observed to be prevalent in separate regions- the former is more visible in the Southern part and along the coast whereas the latter is mostly present in the Northeastern part of the region, away from the Ocean.

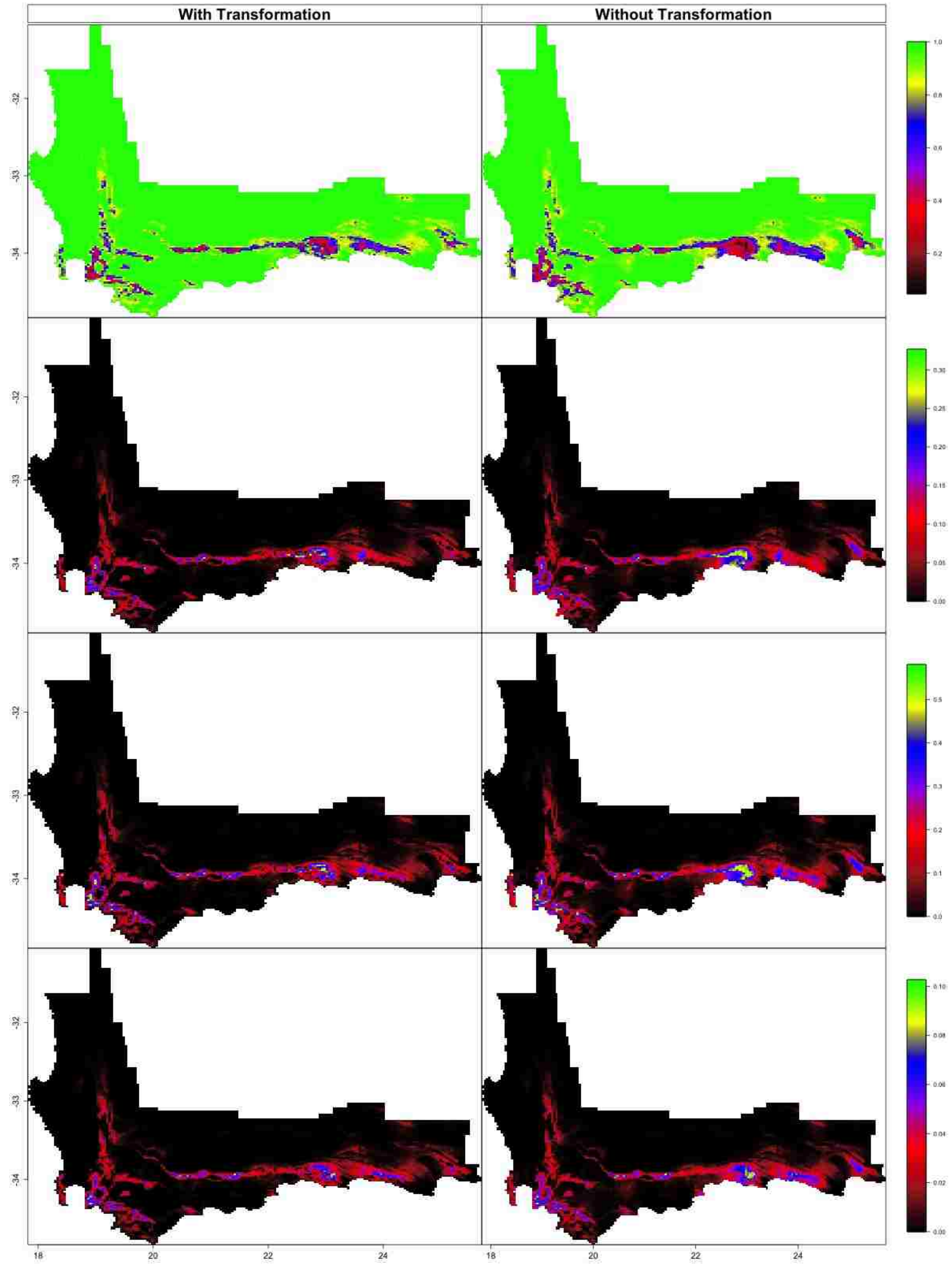


Figure 4.5: Protea cynaroides: Spatial maps of marginal posterior abundance probabilities for category 0 (top) to 3 (bottom) with and without accounting for land transformation

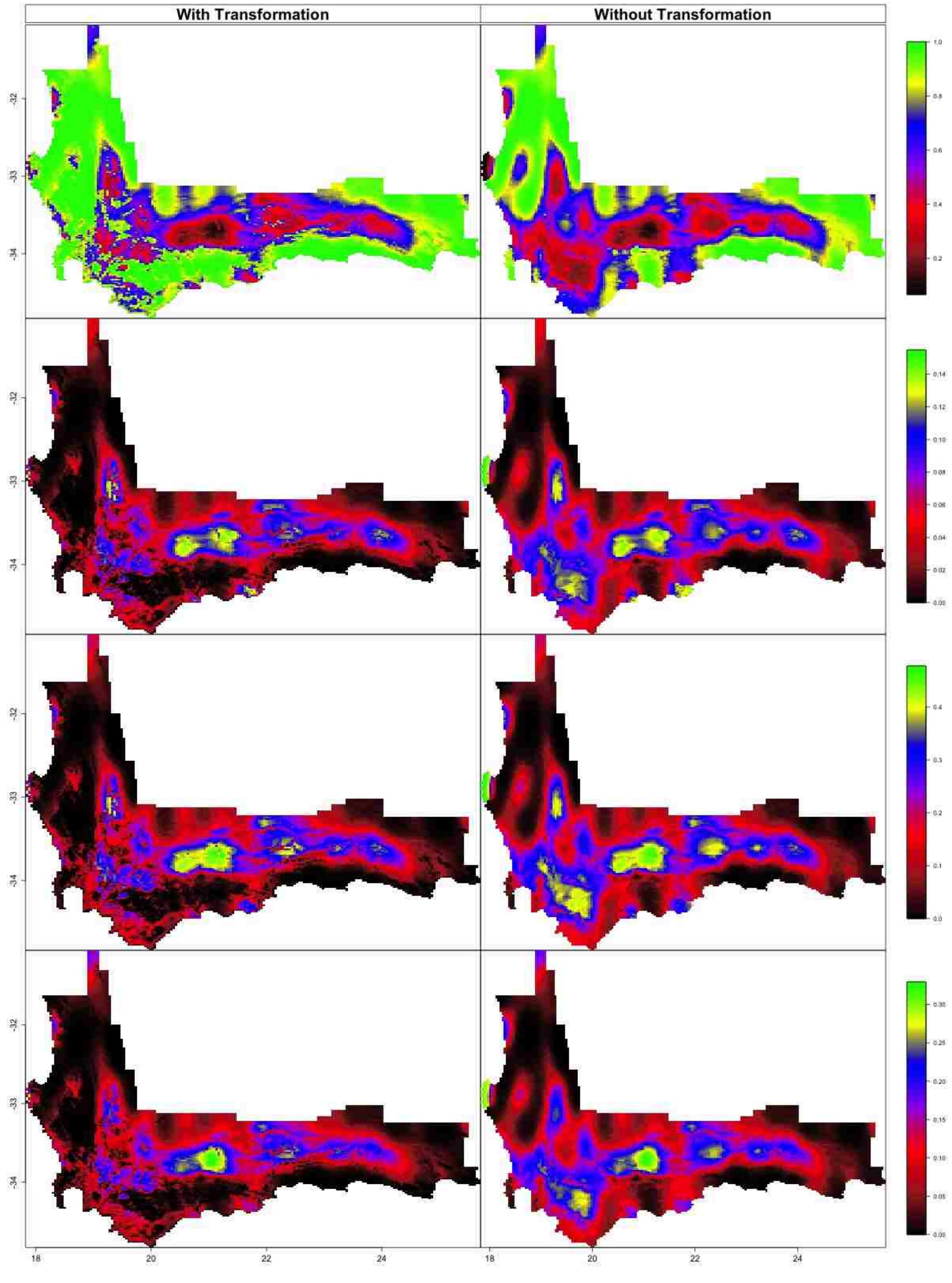


Figure 4.6: Protea repens: Spatial maps of marginal posterior abundance probabilities for category 0 (top) to 3 (bottom) with and without accounting for land transformation

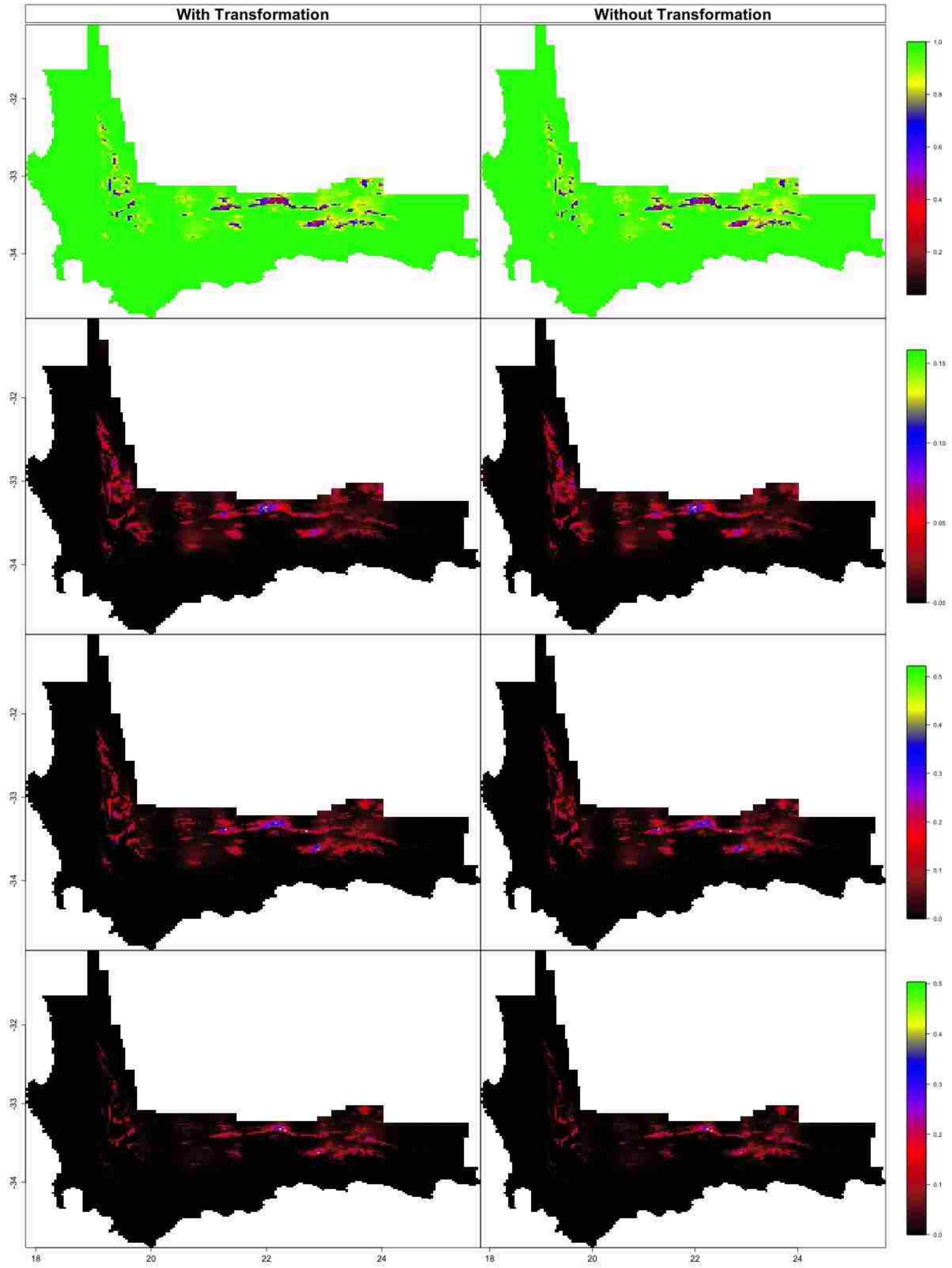


Figure 4.7: Protea punctata: Spatial maps of marginal posterior abundance probabilities for category 0 (top) to 3 (bottom) with and without accounting for land transformation

The influence of land transformation shows notable variation across species. For PRREPE, there is significant difference between the left and right halves of bottom three maps. Some of the regions with high abundance probability in the right part has disappeared in the left part. This is indicative of the fact that significant land transformation has occurred in some of areas environmentally most suitable for PRREPE diminishing its overall presence across the region. For the other two species, the spatial maps with and without transformation do not show any significant visible difference indicating that the their habitat has not witnessed any major land transformation. These findings can be justified from Rebelo (2001). PRPUNC is mostly limited to dry, rocky, or shale slopes which are less suitable for agriculture or development and thus mostly untransformed. PRREPE is much more prevalent across the region and can frequently occur in lowland areas that have been largely transformed by human activities. Hence, the spatial maps help us identify to what extent prevalence of a particular species is influenced by land transformation.

4.6 Discussion

We have developed a Bayesian approach to analyze adjacency-based spatial datasets. Our approach is centered on building a low-rank expansion of the spatial random effect utilizing the spectral properties of graph Laplacian. This approach avoids the sequential sampling usually required for MRF-based spatial models. We enhanced the flexibility of our approach by allowing the dimension of this expansion to be controlled by the data during MCMC. Although our development assumes the entire graph is connected, it can easily be generalized to the case of graphs with multiple connected components. In that case, to avoid singularity in X_k, it suffices to consider all but one of the Laplacian eigenvectors associated with the zero eigenvalue as candidates for inclusion in the model.

We anticipate potential extension of our work in two directions. In Section 4.3, we have motivated this approach based on properties of Laplacian matrix L. It is of theoretical interest to see, if under suitably chosen prior distributions for η_m, the induced prior on θ approaches the CAR prior as $k \to n$. Since Laplacian matrix L is also the precision matrix for the CAR prior (without the scale parameter), it is possible to connect these two prior distributions through same (or related) set of eigenvalues and eigenvectors. Exploring proximity of these distributions under some suitable distance metric would provide theoretical background for our method. Additionally, it can also provide new insight on choice of k by relating it with accuracy of approximation.

From a modeling perspective, our approach can be further enhanced to handle an adaptive neighborhood structure. In this chapter as well as in most of the literature on areal-level spatial analysis, W is set pre-fixed before estimation and often the choice of W is not based on any sound justification. If we think of treating W as a parameter, that would amount to altering the positions of 1 and 0 inside W. In our approach, W influences the estimation through its eigenvalues and eigenvectors. It is not practical to recalculate the k_0 smallest eigenvalues of L and associated eigenvectors, every time we make changes to W. Results from matrix perturbation theory could be useful to check if simple formulas for such recalculations exists for adding or deleting neighbors in W, one at a time.

Chapter 5

Hierarchical Regression model for Outlier Detection

5.1 Introduction

"*An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism*" (Hawkins, 1980). It is almost never true in practice to have datasets without outliers (Pandey et al., 2008). Existence of outliers in the datasets give us inaccurate analysis and wrong predictions. For example, Hatch and Prihoda (1992) show that parameter estimation in regression analysis can be effected by influential outliers. Also, existence outliers in datasets can change the study's results which lead to wrong predictions or are less reliable (Babbar and Chawla, 2010; Indurkhya et al., 2001; Aggarwal, 2013). As a result, these outliers should be discarded. Outlier detection has been studied extensively for the last few decades in many different fields such as Biological Sciences (Yang et al., 2006), Health Sciences (Barghash et al., 2016), Business Management (Kwak and Kim, 2017), Industrial Engineering (Xu et al., 2017), Computer Sciences (Li et al., 2016), etc. Working with large data, detecting outliers become more challenging and complicated. In regression, we can not assume an observation to be an outlier just because it is far from the rest of the data points, but it could be an outlier if the relationship between the response and the covariate is significantly different from most of the data points. Our goal for this chapter is to develop two Bayesian methods to detect outliers in large health datasets. Many methods have used Gaussian error to detect the outliers, in our method we are going to use a novel approach by using t-distributed residuals.

This chapter is organized as follows: In Section 5.2, we describe our two developed outlier detection methods. In Section 5.3, we test our developed methods on 6 simulation datasets that were generated from 3 simulation models. In Section 5.4, we review two existing detection outlier methods in statistical literature and compare their performance against our methods. In Section 5.5, we apply our methods on a real dataset on height of school students.

5.2 Regression with Heavy-tailed Error Distribution

We start with a standard regression setting, where we have n observations where the i^{th} observation has response y_i, the covariate is $x_i = (1, x_{i1}, x_{i2}, \dots, x_{ip})^T$, and the linear regression equation is given by

$$y_i = x_i^T \beta + \epsilon_i, \text{ where } \epsilon_i \sim N(0, \sigma^2)$$

We can not require every observation to have the same variance since outliers can have large positive or large negative uncertainty. We need a different variance for each one, and we can use $\epsilon_i \sim N(0, \sigma_i^2)$. t-distribution has a heavier tail than a Normal distribution, and the heaviness of the tail can be tuned using its degree of freedom (df) as a parameter. Hence, using t-distribution for ϵ can allow for large positive or negative error compared to a normal distribution and the df parameter will control the likelihood of any large error. Therefore, the most flexible way to model this would be to write this as $\epsilon_i \sim t_{\nu_i}(\sigma^2)$. Our new regression model becomes,

$$y_i = x_i^T \beta + \epsilon_i, \text{ where } \epsilon_i \sim t_{\nu_i}(0, \sigma^2) \tag{5.1}$$

However, if we want to do MCMC, the t-distribution is not in standard form. Hence, we are going to use the following result which helps us to express ϵ in t-distribution rather than Normal distribution.

Result: If $x \sim N(0, \frac{\sigma^2}{\lambda})$ and $\lambda \sim \Gamma(\frac{\nu}{2}, \frac{\nu}{2}),$ then $x \sim t_\nu(\sigma^2)$

proof:

$$\begin{aligned}
f(x) &= \int f(x,\lambda) d\lambda = \int f(x|\lambda) f(\lambda) d\lambda \\
&\propto \int (\frac{\lambda}{\sigma^2})^{\frac{1}{2}} \exp\Big(-\frac{1}{2}\lambda \frac{x^2}{\sigma^2}\Big) \lambda^{\frac{\nu}{2}-1} \exp\Big(\frac{-\lambda\nu}{2}\Big) d\lambda \\
&\propto \int (\lambda^{\frac{1}{2}} \lambda^{\frac{\nu}{2}-1}) \exp\Big(\frac{-\lambda x^2}{2\sigma^2}\Big) \exp\Big(-\lambda\nu\Big) d\lambda \\
&\propto \int \lambda^{\frac{\nu+1}{2}-1} \exp\left(-\lambda \left[\frac{x^2}{2\sigma^2} + \frac{\nu}{2}\right]\right) d\lambda \\
&= \frac{\Gamma(\frac{\nu+1}{2})}{\left[\frac{\nu}{2} + \frac{x^2}{2\sigma^2}\right]^{\frac{\nu+1}{2}}} \\
&\propto \frac{1}{\left[\frac{\nu}{2}(1 + \frac{x^2}{\nu\sigma^2})\right]^{\frac{\nu+1}{2}}} \\
&\propto \frac{1}{\left(1 + \frac{x^2}{\nu\sigma^2}\right)^{\frac{\nu+1}{2}}}
\end{aligned}$$

We can rewrite our regression model (Eq. 5.1) in matrix notation as follows,

$$Y = X\beta + \epsilon, \text{ where } \epsilon \sim MVN_n(0_n, \sigma^2\Lambda^{-1}) \tag{5.2}$$

where y is a $n \times 1$ vector, β is $(p+1) \times 1$ vector, X is a $n \times (p+1)$ matrix, σ^2 is a scalar, and $\Lambda = Diag(\lambda_i),\ i = 1, \ldots, n$. The prior distributions for our hierarchical model can be defined as follows,

$$\begin{aligned}
\beta &\sim MVN_n(0, c_0 I_n),\ \sigma^2 \sim IG(a_0, b_0),\ \lambda_i \sim \Gamma(\frac{\nu_i}{2}, \frac{\nu_i}{2}),\ \text{and} \\
\nu_i &\sim f,\ \text{where f is a discrete distribution with probabilities } (q_1, \ldots, q_D) \\
&\quad \text{at values } \nu_{01} < \cdots < \nu_{0D}.
\end{aligned}$$

Very small values for ν_{0j}, where $j = 1, \ldots, D$ and $D > 1$, will indicate a very small df which implies large error is more likely. A large value of df means the error is unlikely to be unusually large. The parameters in the above model are β, σ^2, $\{\lambda_1, \ldots, \lambda_n\}$ and $\{\nu_1, \ldots, \nu_n\}$.

5.2.1 MCMC Algorithms for Outlier Detection Using t-residual

The parameters, β, σ^2, $\{\lambda_1, \ldots, \lambda_n\}$ and $\{\nu_1, \ldots, \nu_n\}$, can be estimated by using an MCMC. To find the conditional posterior distribution for our parameters, we first need to define the Likelihood function. Notice that since $\Sigma = \sigma^2\Lambda^{-1}$, $|\Sigma| = |\sigma^2\Lambda^{-1}| = (\sigma^2)^n(\frac{1}{\lambda_1}\cdots\frac{1}{\lambda_n})$ $= (\sigma^2)^n \prod_{i=1}^{n} \frac{1}{\lambda_i}$, we can define the likelihood function as,

$$\begin{aligned}
L(y|\beta, \sigma^2, \Lambda, \nu) &\propto (\sigma^2)^{-\frac{n}{2}} \prod_{i=1}^{n} (\frac{1}{\lambda_i})^{-\frac{1}{2}} \exp\Big(-\frac{1}{2}\frac{(y - X\beta)^T\,\Lambda\,(y - X\beta)}{\sigma^2}\Big) \\
&\propto (\sigma^2)^{-\frac{n}{2}} \prod_{i=1}^{n} \lambda_i^{\frac{1}{2}} \exp\Big(-\frac{1}{2\sigma^2}\Big[\sum_{i=1}^{n} \lambda_i (y_i - x_i^T\beta)^2\Big]\Big) \\
&= (\sigma^2)^{-\frac{n}{2}} \prod_{i=1}^{n} \lambda_i^{\frac{1}{2}} \prod_{i=1}^{n} \exp\Big(-\frac{1}{2\sigma^2}\lambda_i (y_i - x_i^T\beta)^2\Big) \\
&= (\sigma^2)^{-\frac{n}{2}} \prod_{i=1}^{n} \lambda_i^{\frac{1}{2}} \exp\Big(-\frac{1}{2\sigma^2}\lambda_i (y_i - x_i^T\beta)^2\Big)
\end{aligned}$$

Now, the posterior distribution for β, can be found as follows:

$$\begin{aligned}
\pi(\beta|y, \sigma^2, \Lambda, \nu) &\propto L(y|\beta, \sigma^2, \Lambda, \nu)\,\pi(\beta) \\
&\propto \prod_{i=1}^{n} \exp\Big(-\frac{1}{2\sigma^2}\lambda_i (y_i - x_i^T\beta)^2\Big)\,\exp\Big(-\frac{1}{2}\frac{\beta^T\beta}{c_0}\Big) \\
&\propto \exp\Big(-\frac{1}{2}\Big[(y - X\beta)^T(\frac{\Lambda}{\sigma^2})(y - X\beta) + \frac{\beta^T\beta}{c_0}\Big]\Big) \\
&\propto \exp\Big(-\frac{1}{2}\Big[\frac{(y - X\beta)^T\Lambda(y - X\beta)}{\sigma^2} + \frac{\beta^T\beta}{c_0}\Big]\Big) \\
&\propto \exp\Big(-\frac{1}{2}\Big[\frac{y^T\Lambda y - y^T\Lambda X\beta - \beta^T X^T\Lambda y + \beta^T X^T\Lambda X\beta}{\sigma^2} + \frac{\beta^T\beta}{c_0}\Big]\Big) \\
&\propto \exp\Big(-\frac{1}{2}\Big[\frac{-2\beta^T X^T\Lambda y + \beta^T X^T\Lambda X\beta}{\sigma^2} + \frac{\beta^T\beta}{c_0}\Big]\Big) \\
&\propto \exp\Big(-\frac{1}{2}\Big[-2\beta^T\frac{X^T\Lambda y}{\sigma^2} + \beta^T(\frac{X^T\Lambda X}{\sigma^2})\beta + \frac{\beta^T\beta}{c_0}\Big]\Big)
\end{aligned}$$

$$
\begin{aligned}
&\propto \ \exp\Big(-\frac{1}{2}\Big[-2\beta^T\frac{X^T\Lambda y}{\sigma^2}+\beta^T\Big(\frac{X^T\Lambda X}{\sigma^2}+\frac{I}{c_0}\Big)\beta\Big]\Big) \\
&\propto \ \exp\Big(-\frac{1}{2}\Big[\beta^T A\beta-2\beta^T b\Big]\Big), \text{ where } A=\frac{X^T\Lambda X}{\sigma^2}+\frac{I_n}{c_0} \text{ and } b=\frac{X^T\Lambda y}{\sigma^2}
\end{aligned}
$$

Hence,

$$\beta|- \sim MVN_n(A^{-1}b, A^{-1}) \tag{5.3}$$

The posterior distribution for σ^2 can be written as follows,

$$
\begin{aligned}
\pi(\sigma^2|y,\beta,\Lambda,\nu) &\propto L(y|-)\times\pi(\sigma^2) \\
&\propto (\sigma^2)^{-\frac{n}{2}}\prod_{i=1}^{n}\exp\Big(-\frac{1}{2\sigma^2}\lambda_i(y_i-x_i^T\beta)^2\Big)\exp\Big(-\frac{b_0}{\sigma^2}\Big) \\
&\propto (\sigma^2)^{-\frac{n}{2}} \ \exp\Big(-\frac{1}{2}\frac{(y-X\beta)^T\Lambda(y-X\beta)}{\sigma^2}\Big)(\sigma^2)^{-(a_0+1)} \ \exp\Big(-\frac{b_0}{\sigma^2}\Big) \\
&\propto (\sigma^2)^{-(\frac{n}{2}+a_0+1)} \ \exp\Big(-\frac{1}{\sigma^2}(\frac{G}{2}+b_0)\Big), \text{ where } G=(y-X\beta)^T\Lambda(y-X\beta)
\end{aligned}
$$

Hence,

$$\sigma^2|- \sim IG(\frac{n}{2}+a_0, \frac{G}{2}+b_0) \tag{5.4}$$

Before we show the posterior distribution for Λ, we can prove that $\lambda_1,\ldots,\lambda_n$ have independent posterior distribution given β,σ^2,ν and D as follows,

$$
\begin{aligned}
\pi(\Lambda|\beta,\sigma^2,\nu,y) &\propto L(y|-)\times\pi(\Lambda) \\
&\propto \prod_{i=1}^{n}\lambda_i^{\frac{1}{2}}\exp\Big(-\lambda_i\frac{(y_i-x_i^T\beta)^2}{2\sigma^2}\Big)\times\prod_{i=1}^{n}\pi(\lambda_i) \\
&= \prod_{i=1}^{n}\lambda_i^{\frac{1}{2}}\exp\Big(-\lambda_i\frac{(y_i-x_i^T\beta)^2}{2\sigma^2}\Big)\times\prod_{i=1}^{n}\lambda_i^{\frac{\nu_i}{2}-1}\exp\Big(-\lambda_i\frac{\nu_i}{2}\Big) \\
&= \prod_{i=1}^{n}\lambda_i^{\frac{1}{2}}\exp\Big(-\lambda_i\frac{(y_i-x_i^T\beta)^2}{2\sigma^2}\Big)\lambda_i^{\frac{\nu_i}{2}-1}\exp\Big(-\lambda_i\frac{\nu_i}{2}\Big)
\end{aligned}
$$

Hence,

$$\pi(\Lambda|\beta, \sigma^2, y) \propto \prod_{i=1}^{n} \pi(\lambda_i|\beta, \sigma^2, y)$$

In other words, $\lambda_1, \ldots, \lambda_n$ have an independent posterior distribution given β, σ^2, ν and y. Now, the posterior distribution for Λ, can be found as follows,

$$\begin{aligned}
\pi(\lambda_i|\beta, \sigma^2, \nu, y) &\propto L(y|-) \times \pi(\Lambda) \\
&\propto \lambda_i^{\frac{1}{2}} \exp\left(-\lambda_i \frac{(y_i - x_i^T\beta)^2}{2\sigma^2}\right) \lambda_i^{\frac{\nu_i}{2}-1} \exp\Big(-\lambda_i \frac{\nu_i}{2}\Big) \\
&\propto \lambda_i^{\frac{\nu_i+1}{2}-1} \exp\left(-\lambda_i \left[\frac{(y_i - x_i^T\beta)^2}{2\sigma^2} + \frac{\nu_i}{2}\right]\right)
\end{aligned}$$

Hence,

$$\lambda_i|- \sim \Gamma\left(\frac{\nu_i+1}{2}, \frac{(y_i - x_i^T\beta)^2}{2\sigma^2} + \frac{\nu_i}{2}\right),\ i = 1, 2, \ldots, n. \tag{5.5}$$

Next, we can write the posterior distribution of ν_i as follows,

$$\begin{aligned}
\pi(\nu_i|\lambda_i) &\propto f(\lambda_i|\nu_i)\pi(\nu_i) \\
&\propto \frac{(\frac{\nu_i}{2})^{\frac{\nu_i}{2}}}{\Gamma(\frac{\nu_i}{2})} (\lambda_i)^{\frac{\nu_i}{2}-1} \exp\Big(-\frac{\lambda_i \nu_i}{2}\Big)\pi(\nu_i)
\end{aligned}$$

Notice that since $\pi(\nu_i) = 0$ if $\nu_i \notin \{\nu_{01}, \ldots, \nu_{0D}\}$ and $\pi(\nu_i|\lambda_i) \propto f(\lambda_i|\nu_i) \times \pi(\nu_i)$, then $\pi(\nu_i|\lambda_i) = 0$ if $\nu_i \notin \{\nu_{01}, \ldots, \nu_{0D}\}$. First, define c_{0i} for $i = 1, 2, \ldots, D$,

$$c_{0i} = \frac{(\frac{\nu_i}{2})^{\frac{\nu_i}{2}}}{\Gamma(\frac{\nu_i}{2})} (\lambda_i)^{\frac{\nu_i}{2}-1} e^{-\frac{\lambda_i \nu_i}{2}}, \text{ and } p_{ij} \propto c_{0i} q_j$$

Hence, $\pi(\nu_i = \nu_{0j}|\lambda_i) = p_{ij}$ and since $\sum_{j=1}^{D} p_{ij} = 1$, we obtain

$$p_{ij} = \frac{c_{0i} q_j}{\sum_{i=1}^{D} c_{0i} q_j} \tag{5.6}$$

The posterior distribution of $\nu_i|\lambda_i$ is a multinomial distribution with D possible values and probabilities $p_{i1}, p_{i2}, \ldots, p_{iD}$.

Table 5.1: Prior and posterior probabilities distribution for ν_i

values	ν_{01}	ν_{02}	$\ldots$	ν_{0D}
prior probabilities	q_1	q_2	$\ldots$	q_D
posterior probabilities	p_{i1}	p_{i2}	$\ldots$	p_{iD}

From the above discussion, we can summarize Method-I by the following algorithm,

Algorithm 12 Method-I

INPUT: Initial values a_0, b_0 and prior parameters c_0, σ_0.

OUTPUT: Samples for $\beta, \sigma^2, \nu_i, u_i$.

```
for $k = 1, 2, \ldots, N$ do
    draw $\beta|-$ from $MVN(A^{-1}b, A^{-1})$
    draw $\sigma^2|-$ from $IG(\frac{n}{2} + a_0, \frac{G}{2} + b_0)$
    draw $\lambda_i|-$ from $Ga(\frac{\nu_i+1}{2}, \frac{(y_i - x_i^T\beta)^2}{2\sigma^2} + \frac{\nu_i}{2})$
    draw $\nu_i|-$ from $f$, where $f$ is a discrete distribution such that $p[\nu_i = \nu_{0j}] = p_{ij}$
end for
```

In the above method, Method-I, we assumed that the variance of the error, σ_i^2, and the covariance are independent. We improve the method by allowing the variance to change depending on the X values for each observation. We assume that σ_i^2 depends on X in the form function f, and we define $X^* = f(X) = [f(x_1) \cdots f(x_p)]$. We chose two different forms of the function f, which are $f(X) = |X|$ and $f(X) = X^2$. In reality, we do not know the correct function form, so we used both functions, $f(X) = |X|$ and $f(X) = X^2$, and we denote them by Method-IIa and Method-IIb, respectively.

In general, our model is:

$$Y = X\beta + \epsilon, \text{ where } \epsilon \sim NVN_n(0_n, \sigma^2\Lambda^{-1}G^*) \tag{5.7}$$

where $G^* = Diag(g_i)$, $g_i = \exp(x_i^{*^T}\alpha)$, and $x_i^* = (x_{i1}^*, x_{i2}^*, \dots x_{ip}^*)^T$. The prior distributions for the hierarchical model are:

$$\begin{aligned}
\beta &\sim MVN_n(0_n, c_0 I_n),\ \alpha \sim MVN_n(0_n, d_0 I_n),\ \sigma^2 \sim IG(a_0, b_0),\ \lambda_i \sim \Gamma(\frac{\nu_i}{2}, \frac{\nu_i}{2}), and \\
\nu_i &\sim f,\ \text{where } f \text{ is a discrete distribution with probabilities } (q_1, \dots, q_D) \\
&\quad \text{at values } \nu_{01} < \dots < \nu_{0D}.
\end{aligned}$$

Method-I is a special case from Method-II when $\alpha = 0$. The parameters that we need to estimate in Method-II are $\beta, \alpha, \sigma^2, \{\lambda_i, \dots, \lambda_n\}$ and $\{\nu_i, \dots, \nu_n\}$.

Before we derive the conditional posterior distribution for our parameters, we need to define the likelihood function. From Eq. 5.7, $y_i \sim N(x_i^T\beta, g_i\frac{\sigma^2}{\lambda_i})$, and hence the likelihood function can be defined as follows

$$\begin{aligned}
L(y|-) &\propto \sigma^{-\frac{n}{2}} \prod_{i=1}^{n} \exp\Big(-\frac{1}{2}\frac{(y_i - x_i^T\beta)^T \lambda_i \exp(x_i^{*^T}\alpha)(y_i - x_i^T\beta)}{\sigma^2}\Big) \\
&\propto \sigma^{-\frac{n}{2}} \prod_{i=1}^{n} \exp\Big(-\frac{1}{2} h_i \exp(x_i^{*^T}\alpha)\Big),\ \text{where } h_i = \frac{(y_i - x_i^T\beta)^2\lambda_i}{\sigma^2}
\end{aligned}$$

In Method-II, the posterior distribution for ν_i is exactly the same in Method-I. As we show in Method-I, we can easily show that the posterior distribution for β is:
$\beta|- \sim MVN(A^{-1}b, A^{-1})$, where $A = \frac{1}{\sigma^2}X^T\Lambda G^* X + \frac{I_n}{c_0} + \frac{I_n}{d_0}$, $b = \frac{1}{\sigma^2}X^T\Lambda G^* y$, $G^* = Diag(g_i)$, $i = 1, \dots, n$, and $g_i = \exp(x_i^{*T}\alpha)$

Similarly, we can show that the posterior distribution for λ_i is:

$\lambda_i|- \sim \Gamma(\frac{\nu_i+1}{2}, \frac{(y_i - x_i^T\alpha)^2}{2g_i\sigma^2} + \frac{\nu_i}{2})$

And the posterior distribution for σ^2 is:

$\sigma^2|- \sim IG(\frac{n}{2} + a_0, \frac{G^{new}}{2} + b_0)$, where $G^{new} = (y - X\beta)^T\Lambda G^*(y - X\beta)$

The new parameter in Method-II is α, and the posterior distribution for α is:

$$\begin{aligned}
\pi(\alpha|-) &\propto L(y|-) \times \pi(\alpha) \\
&\propto \prod_{i=1}^{n} \exp\Big(-\frac{1}{2} h_i \exp(x_i^{*^T}\alpha)\Big) \exp(x_i^{*^T}\alpha)^{-\frac{1}{2}}, \text{ where } h_i = \frac{(y_i - x_i^T\beta)^2\lambda_i}{2\sigma^2} \\
&\propto \exp\Big(-\frac{1}{2}\sum_{i=1}^{n} x_i^{*^T}\alpha\Big) \exp\Big(-\sum_{i=1}^{n} h_i \exp(x_i^{*^T}\alpha)\Big)
\end{aligned}$$

The above posterior is not a standard distribution, so we use MH algorithm to sample from α. We can generate the proposed value for α as:

$$\alpha_j^{proposed} \sim N(\alpha_j^{old}, \sigma^{2^{proposed}})$$

In MH, we accept the $\alpha^{proposed}$ as a new value if:

$$u < \frac{\pi(\alpha^{proposed}|-)}{\pi(\alpha^{old}|-)}, \text{ where } u \sim \text{unif}(0,1)$$

We can rewrite the above expression as:

$$\log(u) < \log(\pi(\alpha^{proposed}|-)) - \log(\pi(\alpha^{old}|-))$$

We can summarize Method-II by the following algorithm.

Algorithm 13 Method-II

INPUT: Initial values a_0, b_0 and prior parameters c_0, σ_0.

OUTPUT: Samples for $\beta, \alpha, \sigma^2, \nu_i$.

```
for $k = 1, 2, \ldots, N$ do
    draw $\beta|-$ from $MVN(A^{-1}b, A^{-1})$
    draw $\sigma^2|-$ from $IG(\frac{n}{2} + a_0, \frac{G^*}{2} + b_0)$
    draw $\lambda_i|-$ from $\Gamma(\frac{\nu_i+1}{2}, \frac{(y-x_i^{*T}\alpha)^2}{2g_i\sigma^2} + \frac{\nu_i}{2})$
    draw $\nu_i|-$ from $f$, where $f$ is a discrete distribution such that $p[\nu_i = \nu_{0j}] = p_{ij}$
    draw $\alpha$ from MH algorithm as follows:
    draw $\alpha_j^{proposed} \sim N(\alpha_j^{old}, \sigma^{2^{proposed}})$
    draw $u_i$ from unif$(0, 1)$
    if $\log(u_i) < \log(\pi(\alpha^{proposed})) - \log(\pi(\alpha^{old}))$ then
        $\alpha = \alpha^{proposed}$
    else
        $\alpha = \alpha^{old}$
    end if
end for
```

Rule for Detecting Outliers: If we want to assign a rule for detecting outliers, we need to set a threshold for value of ν_i, *e.g.* v_0. Since smaller values of ν_i indicate higher likelihood of unusually large error, for all observations with posterior mean $E(\nu_i|Data) < \nu_0$, we identify them as outliers. If we increase (decrease) ν_0, we will detect more (fewer) outliers.

5.3 Simulation Studies

In this section, we applied our methods on simulation datasets. First, we described the simulation models that we used to generate the datasets. Then, we checked the accuracy of β estimation in the presence of the outliers. In the end of the section, we applied the methods in different datasets and compared among them.

5.3.1 Model for Simulation

Three simulation models have been used to simulate 6 different datasets with 2000 observations for each. In the first simulation model, denoted by SI, we assumed that the variance does not depend on X, so SI can be defined as follows,

$$Y = X\beta + \epsilon, \text{ where } \epsilon \sim MVN_n(0_n, Diag((\sigma^2))_{i=1}^n)$$

We want to work with more challenging datasets. So in the following two simulation models, we assumed the variance depends on the observations through a function $f(X)$. We are going to use two different function forms.

The second simulation model was denoted by SIIa, and defined as follows,

$$Y = X\beta + \epsilon, \text{ where } \epsilon \sim MVN_n(0_n, \sigma^2 Diag((e^{x_i^{*T}\alpha}))_{i=1}^n), \text{ where } x_i^* = (x_{i1}^{*2}, x_{i2}^{*2}, \ldots, x_{ip}^{*2})^T$$

The third simulation model was denoted by SIIb, and defined as follows,

$$Y = X\beta + \epsilon, \text{ where } \epsilon \sim MVN_n(0_n, \sigma^2 Diag((e^{x_i^T\alpha}))_{i=1}^n), \text{ where } x_i^* = (|x_{i1}^*|, |x_{i2}^*|, \ldots, |x_{ip}^*|)^T$$

In SI, we used $\beta = (0.1, -1.5, 2.7)$ and $\sigma = 2$. In SIIa and SIIb, we used the same β and σ values that we used in Model-SI. In addition, we used $\alpha = (3.5, -0.5)$. We randomly chose 20 and 100 positions from our data points, and for each position we chose a magnitude of error randomly between 6 and 9, and randomly added and subtracted from the error the observed value.

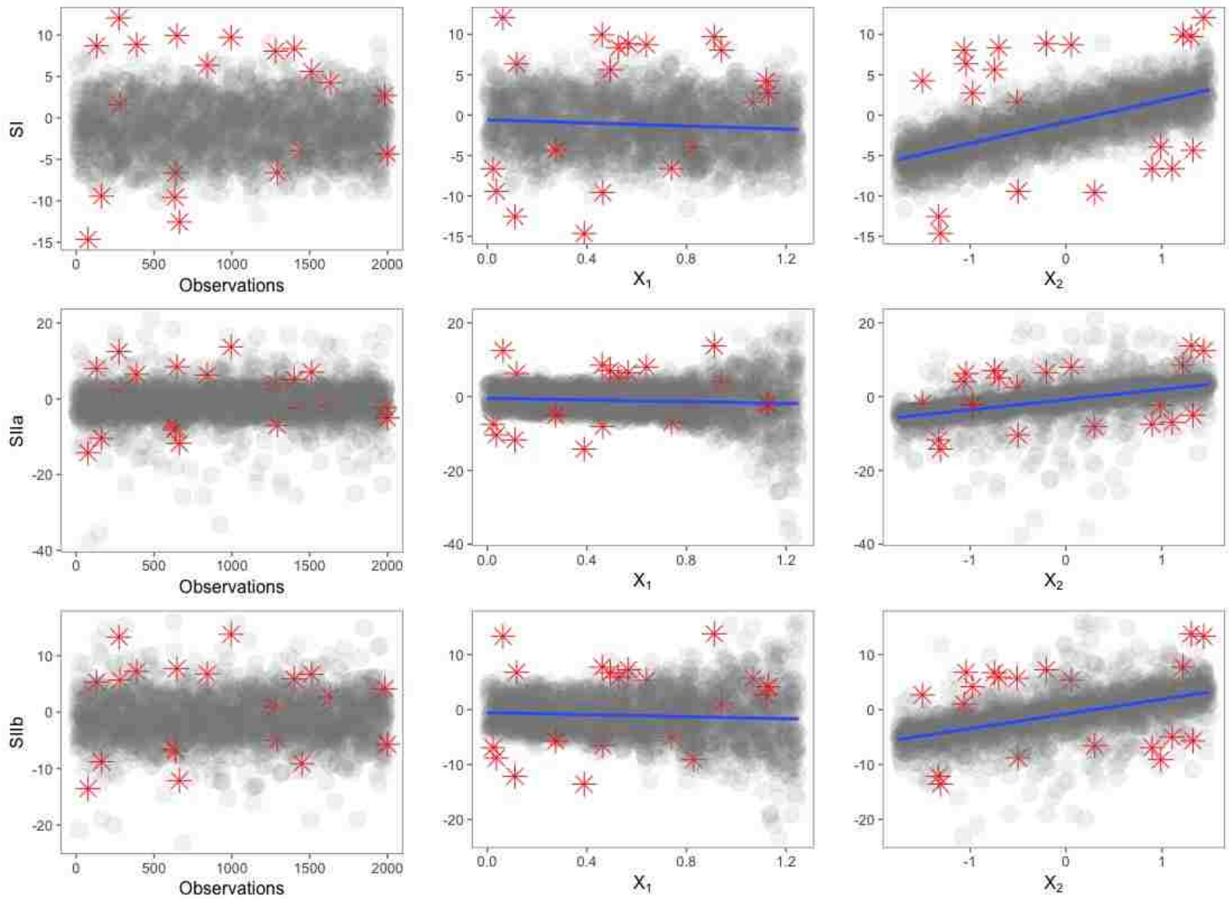


Figure 5.1: Datasets with 20 outliers: the true outliers are marked with red stars, and the blue line represents the trend of the data points.

In Figures 5.1 and 5.2, we plotted the datasets with 20 and 100 outliers, respectively. The plots show us the observations and the observations versus X_1 and X_2. It is easy to tell from the plot that β_1 and β_2 have negative and positive trend, respectively. It is easy to identify some of the outliers just by visual inspection in SI datasets. In contrast, in SIIa and SIIb, it is difficult to identify the outliers because in many cases the outliers are nested among the observations, and they lie in the same region with the other observations. Also, some data points look more extreme even though they are not outliers.

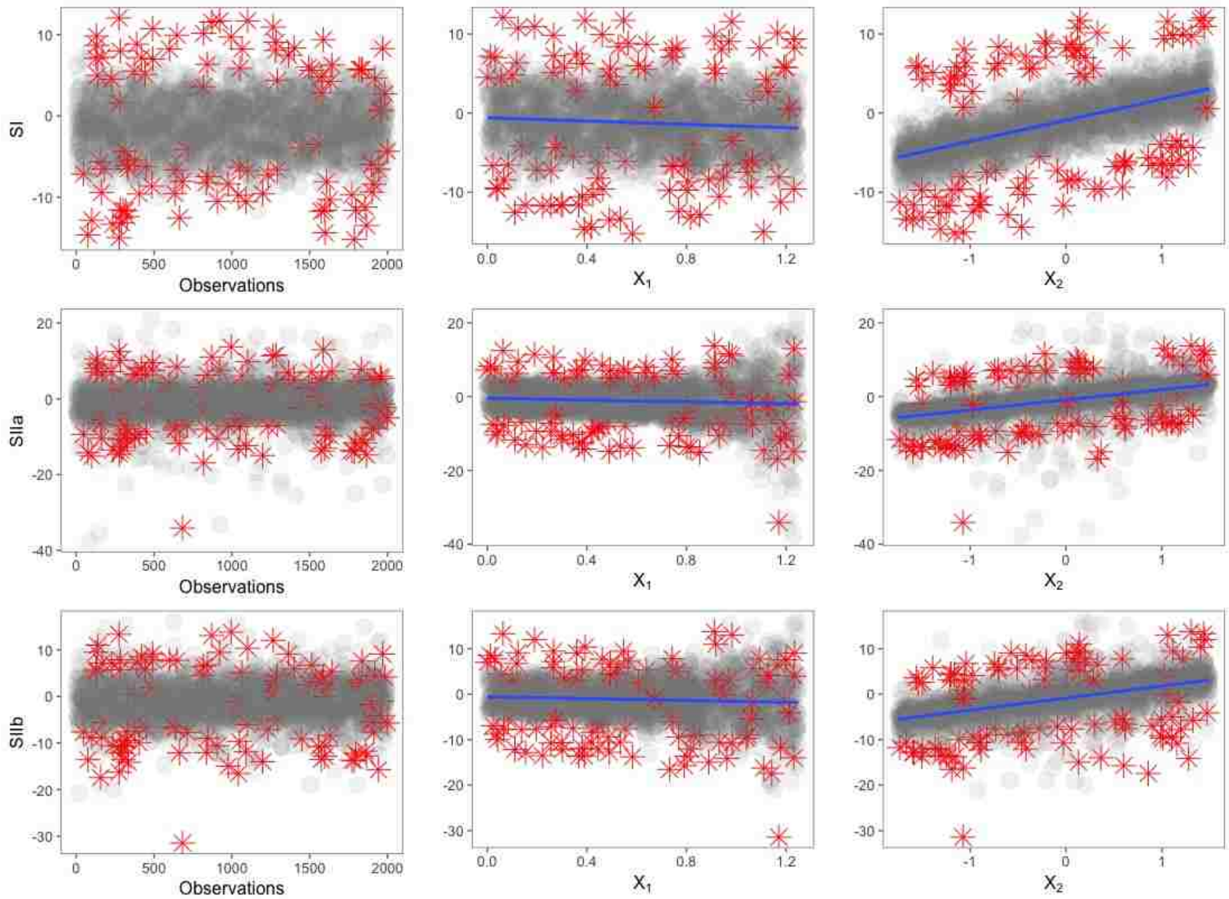


Figure 5.2: Datasets with 100 outliers: the true outliers are marked with red stars, and the blue line represents the trend of the data points.

5.3.2 Accuracy of coefficient estimation in presence of outliers

We run the methods that we have developed in the previous sections with the same setting for all datasets. For the prior parameters, we chose $a_0 = 2.1$, $b_0 = 1$, $c_0 = 10000$ and $d_0 = 10000$. The least-square estimators were chosen to be our initial values. We chose $\nu_0 = (2, 4, 5, 10, 30, 40, 50, 80, 90, 100)$, and $\sigma^{2^{proposed}} = 0.15$ to get an acceptance ratio between 35% and 45%. We run the codes for 10000 iterations, we burned-in the first 25%, and we thinned the rest by five.

Before analyzing the performance on detection of outliers, we first want to check if our method produces a more reliable coefficient estimator than the ones produced by usual

Gaussian regression in the presence of outliers. In the following tables, we showed the true parameters, their point estimate, distance from the point estimate, 95% credible intervals, and the width of these intervals.

In the Gaussian case, we assume the correct simulation model but the error is Gaussian. However, in our methods, we want to account for outliers, so we assumed every observation has a t-error as mentioned in Section 5.2. We marked the 95% credible intervals that contain the true parameters with green, and the 95% CI that do not contain the true parameters with red.

In Table 5.2, we compared the coefficient estimation between the two methods for Simulation models SI, SIIa and SIIb, respectively, for both 20 and 100 outliers datasets.

We see that all the credible intervals for the estimation of regression coefficient cover the true simulation parameters, but with wider intervals in the Gaussian case compared to our methods with much smaller intervals. In fact, when there are outliers in datasets, and we do not account for these outliers, we can still correctly cover the regression coefficients, but we can not cover the variance.

Also, in the effect of parameters in the variance term, there are no cases where the credible intervals can cover the true parameters, and we see that the estimated values are very far from the true values. This is because there are outliers in the data, and we are not adjusting for outliers. That means the model has to have high variance to reach those far observations, and hence the estimated variance will be more effected. That is why we perform much worse with estimation of α than the estimation of β. Moreover, in the above three tables, we see the estimated parameters in the t-distribution case is not sensitive to the number of outliers. Also, the t-distribution case produced close estimated values and less uncertainty compared to the Gaussian case when we have 1% and 5% of outliers.

5.3.3 Model Application on the Simulated Datasets

In the following figures, we plotted the posterior mean for the ν for all observations. Clearly, we can see the posterior mean for most of the true outliers is far below the posterior mean for the majority of non-outlier values, as expected. In Figure 5.3, in both 20 and 100 outliers datasets, it was easy to identify most of the outliers from the other data points using all the three methods. In Figures 5.4 and 5.5, Method-I did not work well for both models because it did not assume covariate-dependence variance for each observation. We saw that for Method-IIa and Method-IIb the performance on SIIa and SIIb are comparable, which essentially means for the current simulation the detection of outliers is not very sensitive with respect to choice of function of covariate-dependence. It was easy to identify up to 70% of the outliers, but since there were a few outliers which were hard to detect, we have to drop most of the data points to exclude them.

Table 5.2: Parameter comparison among all methods across the three simulation models

Simulation Parameters	**Method-I on SI**			
	20 outliers datasets		100 outliers datasets	
	with Gaussian error	with t-dist error	with Gaussian error	with t-dist error
$\beta_0 = 0.10$	-0.027(+0.127) **(-0.228, 0.169)**	0.003(+0.103) **(-0.192, 0.179)**	-0.023(+0.123) **(-0.270, 0.218)**	0.010(+0.090) **(-0.191, 0.209)**
$\beta_1 = -1.50$	-1.351(-0.185) **(-1.589, -1.038)**	-1.411(-0.089) **(-1.659, -1.149)**	-1.498(-0.097) **(-1.741, -1.063)**	-1.511(+0.011) **(-1.788, -1.216)**
$\beta_2 = 2.70$	2.682(+0.018) **(2.581, 2.790)**	2.690(+0.010) **(2.593, 2.788)**	2.693(+0.007) **(2.569, 2.826)**	2.699(+0.001) **(2.592, 2.805)**
Simulation Parameters	**Method-IIa on SIIa**			
	20 outliers datasets		100 outliers datasets	
	with Gaussian error	with t-dist error	with Gaussian error	with t-dist error
$\beta_0 = 0.10$	0.107(-0.007) **(-0.025, 0.235)**	0.148(+0.048) **(0.052, 0.241)**	0.124(-0.024) **(-0.085, 0.323)**	0.163(-0.063) **(0.062, 0.266)**
$\beta_1 = -1.50$	-1.503(+0.003) **(-1.814, -1.189)**	-1.606(+0.106) **(-1.840, -1.367)**	-1.617(+0.117) **(-2.044, -1.179)**	-1.655(+0.155) **(-1.914, -1.398)**
$\beta_2 = 2.70$	2.682(+0.018) **(2.645, 2.722)**	2.700(0.000) **(2.695, 2.705)**	2.694(+0.006) **(2.610, 2.783)**	2.7(0.00) **(2.694, 2.705)**
$\alpha_1 = 3.50$	2.819(+0.681) **(2.690, 2.940)**	3.535(-0.035) **(3.367, 3.688)**	2.051(+1.449) **(1.926, 2.171)**	3.415(+0.085) **(3.259, 3.563)**
$\alpha_2 = -0.50$	0.339(-0.161) **(-.418, -0.256)**	-0.514(-0.014) **(-0.609, -0.423)**	-0.197(-0.303) **(-0.274, -0.119)**	-0.524(+0.024) **(-0.617, -0.437)**
Simulation Parameters	**Method-IIb on SIIb**			
	20 outliers datasets		100 outliers datasets	
	with Gaussian error	with t-dist error	with Gaussian error	with t-dist error
$\beta_0 = 0.10$	-0.045(+0.145) **(-0.195, 0.100)**	0.119(-0.019) **(0.025, 0.215)**	-0.017(+0.117) **(0.232, 0.190)**	0.140(-0.040) **(0.025, 0.266)**
$\beta_1 = -1.50$	-1.211(-0.289) **(-1.557, -0.865)**	-1.28(-0.22) **(-1.565, -0.995)**	-1.356(-0.144) **(-1.778, -0.933)**	-1.316(+0.184) **(-1.614, -1.012)**
$\beta_2 = 2.70$	2.669(+0.031) **(2.572, 2.763)**	2.64(+0.06) **(2.570, 2.710)**	2.692(+0.008) **(2.559, 2.821)**	2.646(+0.054) **(2.572, 2.722)**
$\alpha_1 = 3.50$	2.681(+0.819) **(2.510, 2.845)**	3.472(+0.028) **(3.278, 3.686)**	1.948(+1.552) **(1.778, 2.106)**	3.372(+0.128) **(3.178, 3.582)**
$\alpha_2 = -0.50$	-0.350(-0.150) **(-.482, -0.222)**	-0.562(-0.062) **(-0.708, -0.422)**	-0.302(-0.198) **(-0.430, -0.178)**	-0.556(+0.056) **(-0.714, -0.408)**

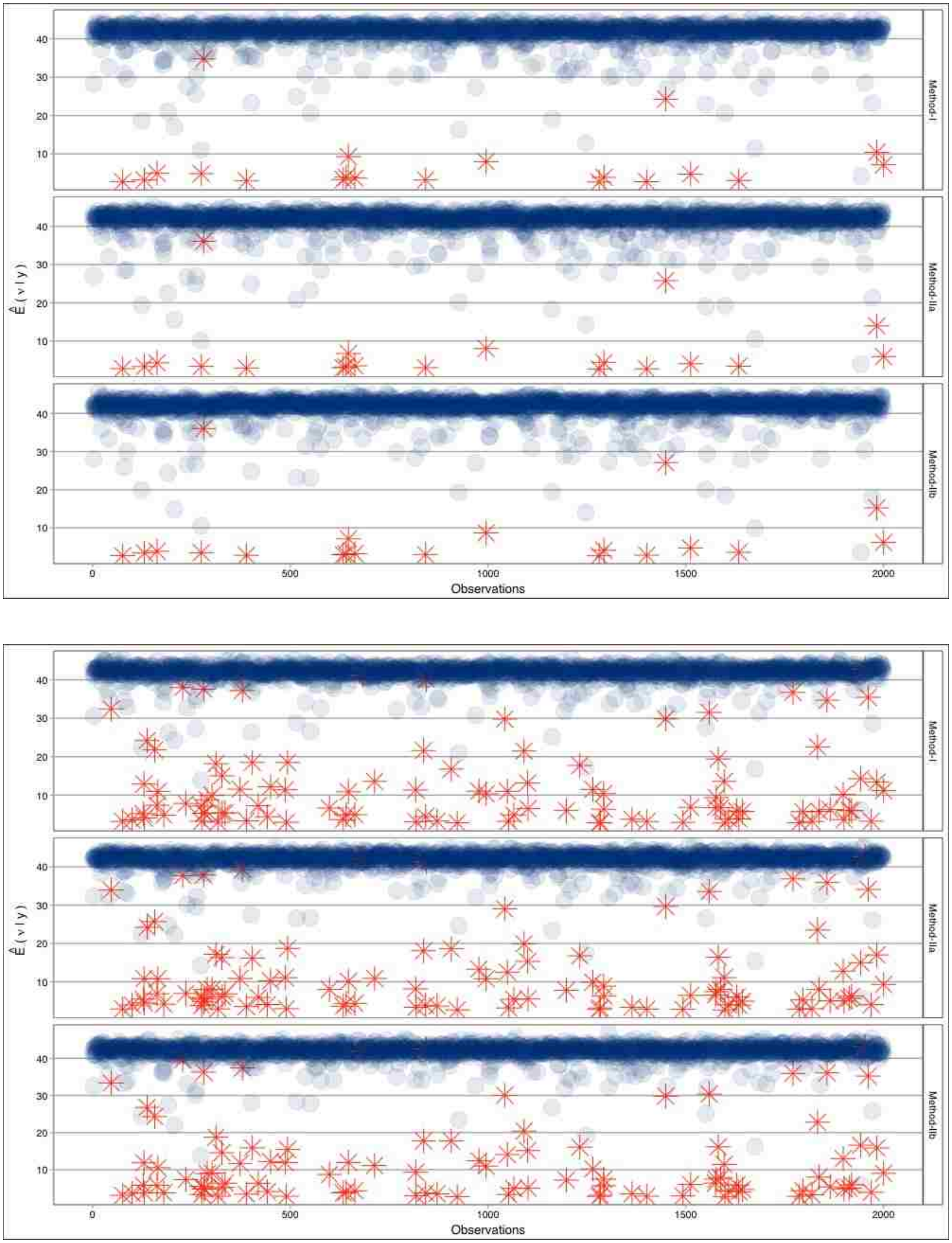
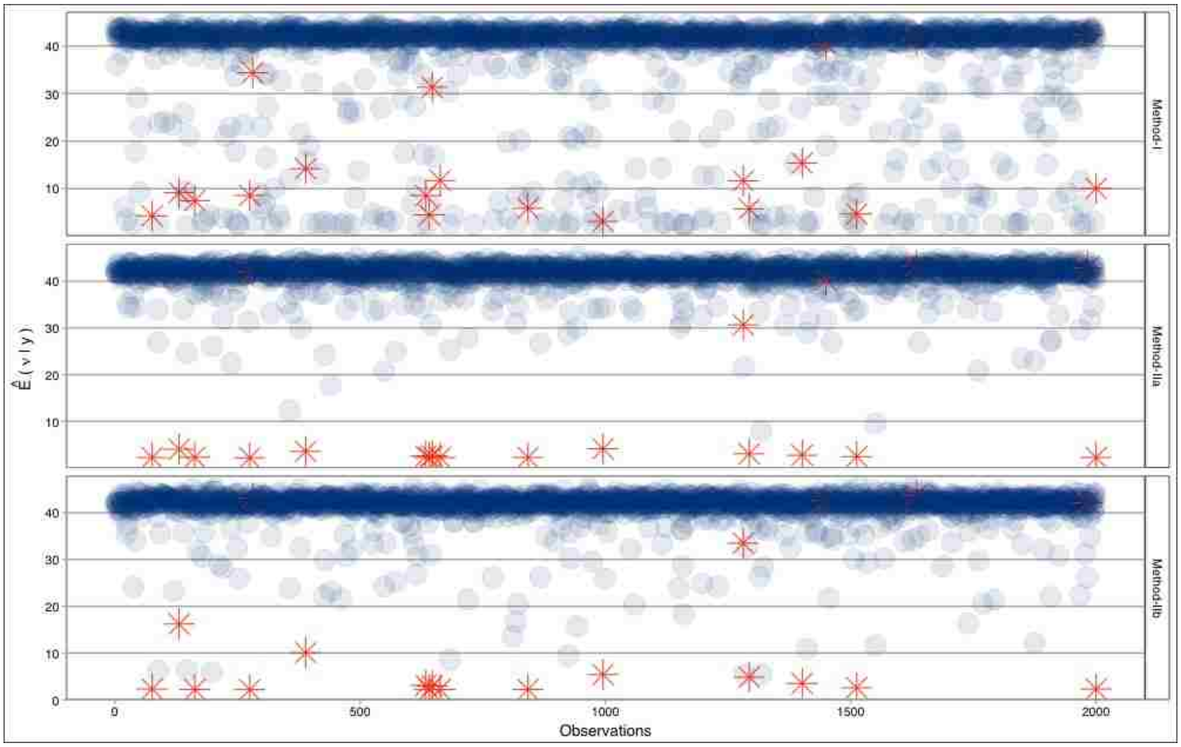


Figure 5.3: Outlier detection using posterior sample mean for ν for Method-I, IIa and IIb on SI (Top) with 20 outliers and (Bottom) with 100 outliers: The actual outliers are marked as red stars.



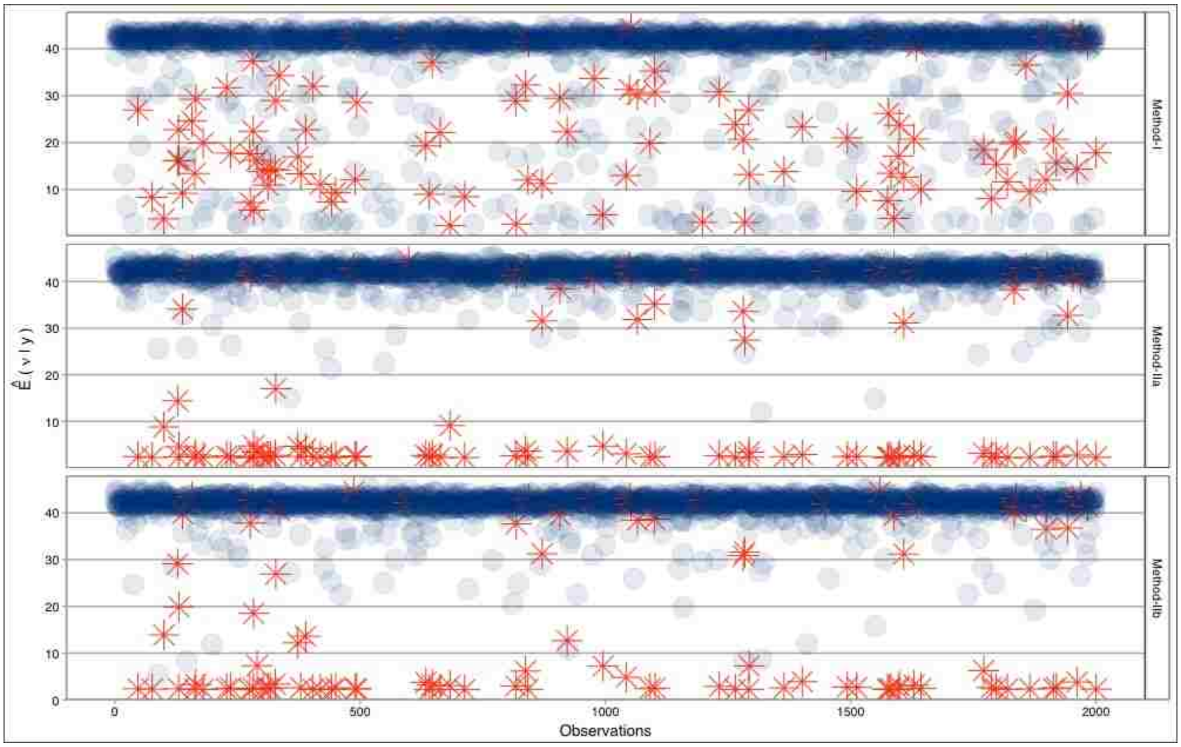
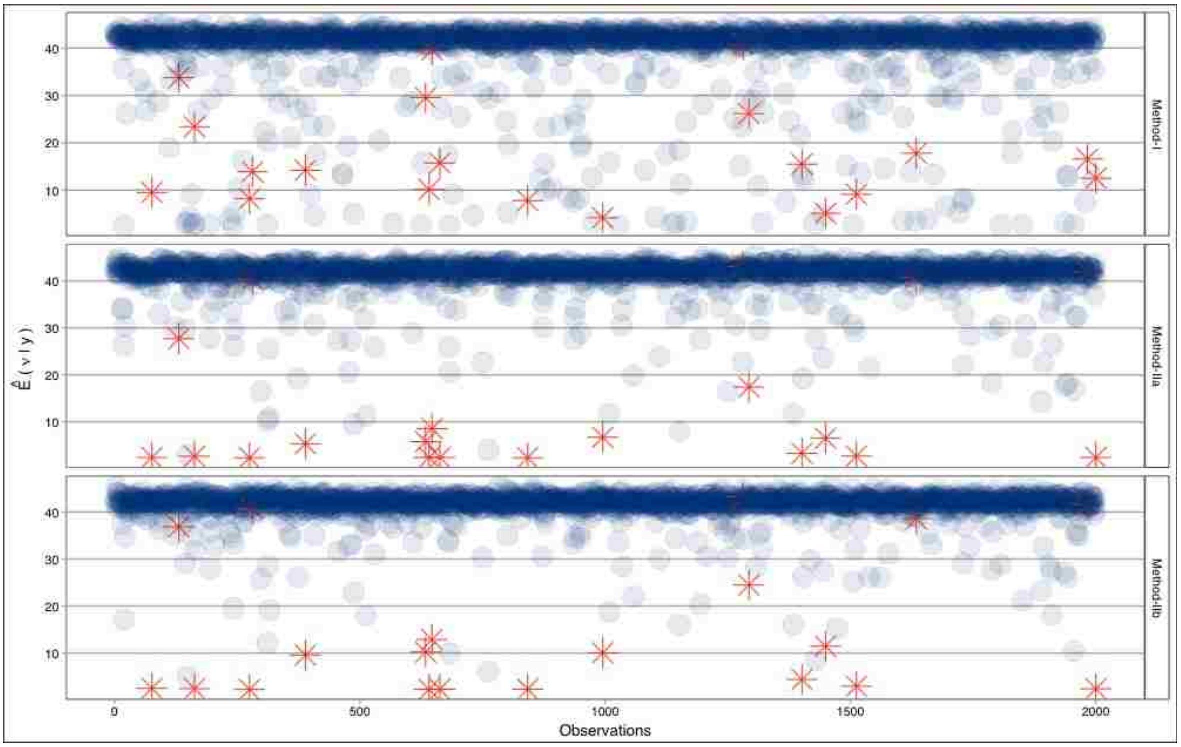


Figure 5.4: Outlier detection using posterior sample mean for ν for Method-I, IIa and IIb on SIIa (Top) with 20 outliers and (Bottom) with 100 outliers: The actual outliers are marked as red stars.



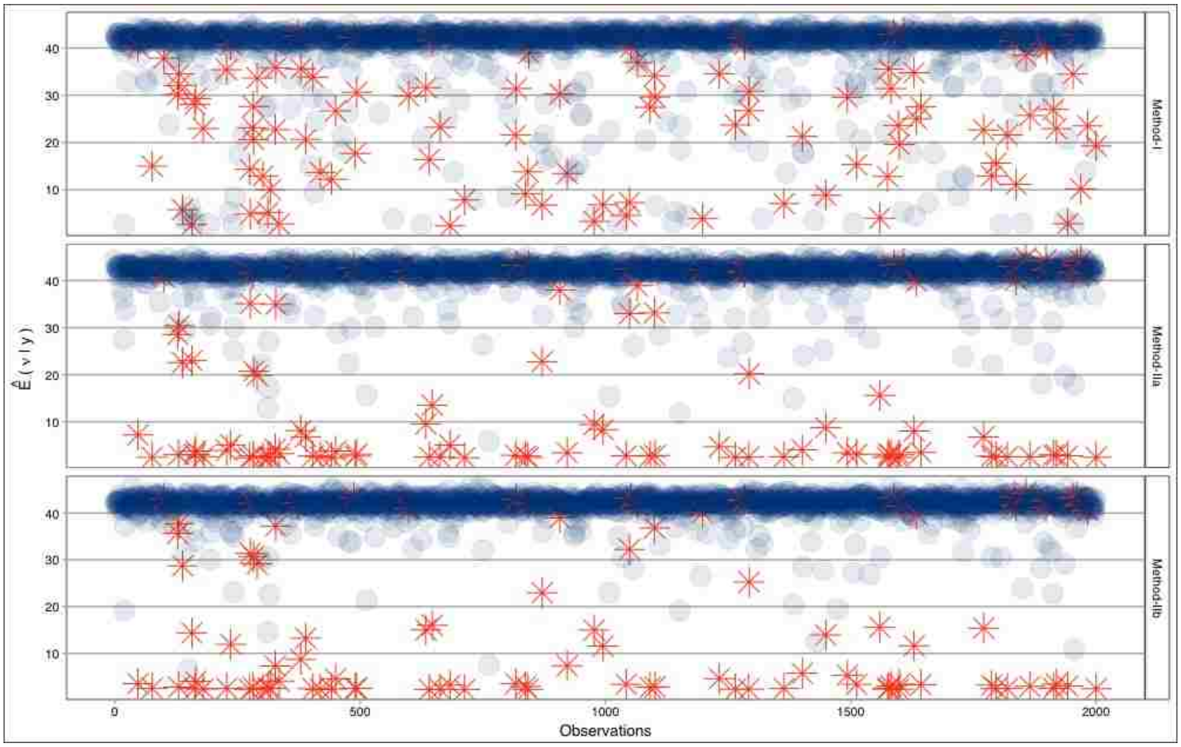


Figure 5.5: Outlier detection using posterior sample mean for ν for Method-I, IIa and IIb on SIIb (Top) with 20 outliers and (Bottom) with 100 outliers: The actual outliers are marked as red stars.

5.3.4 Comparison Among Methods for Detection of Outliers

We compared all the methods across three simulation models. One statistic for comparison could be to check how many observations we need to discard if we want to throw away a certain percentage of true outliers. The perfect case will delete only those data points that are true outliers. A method will be more efficient if it needs to discard a smaller number of observations compared to another method for deleting the same proportion of outliers. We compared the performance to detect outliers ranging from 10% to 90% (essentially detection of outliers that are either "too easy" or " too hard" to detect).

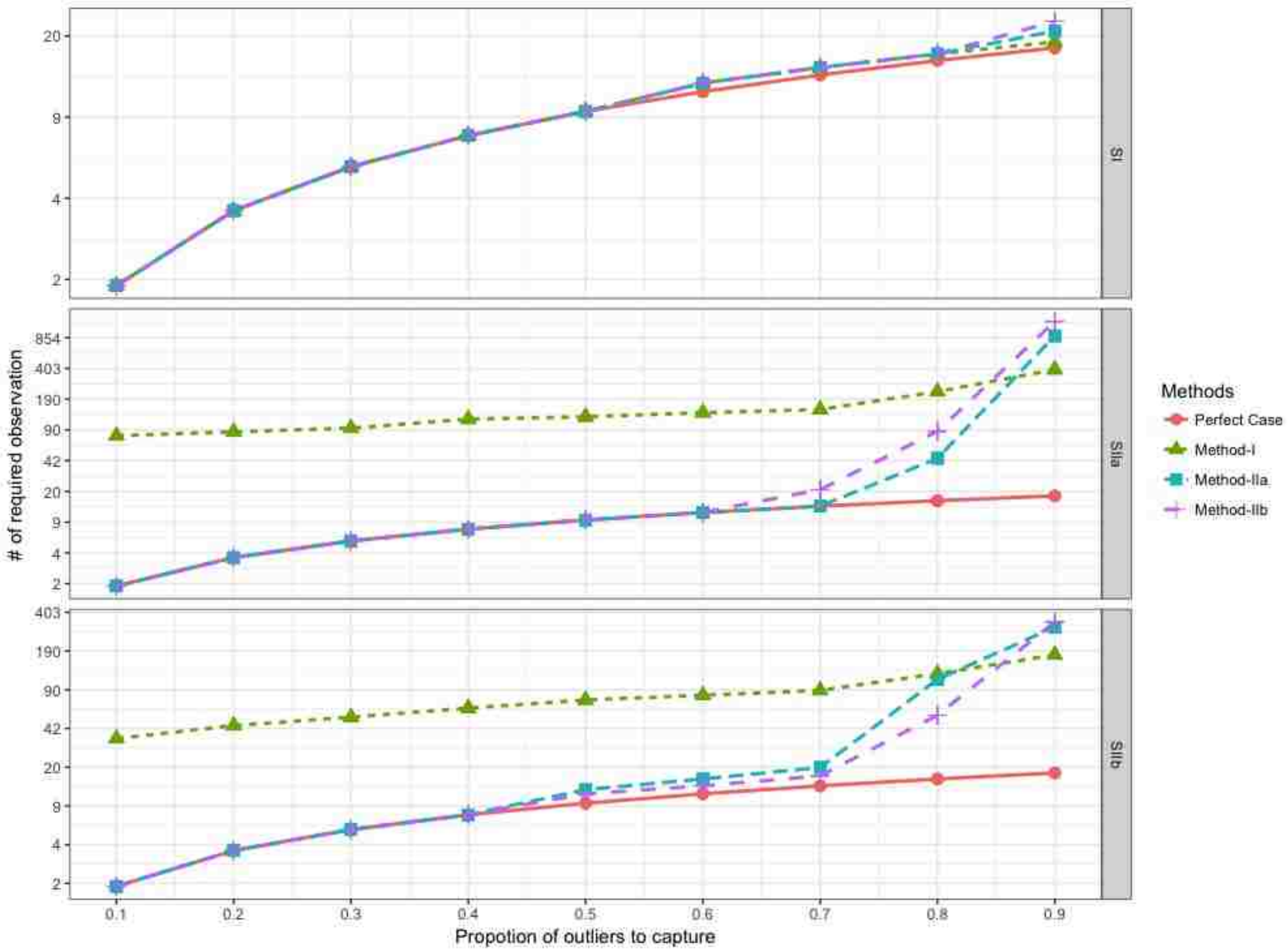


Figure 5.6: The number of observations that should be considered to detect different proportions of the actual outliers on datasets with 20 outliers

In Figures 5.6 and 5.7, we can see that all methods perform almost perfectly in SI. In SIIa and SIIb, Method-I could not detect the outliers easily. In contrast, Method-IIa and Method-IIb could detect up to 70% of the outliers with the same efficiency as the perfect case, and their performance becomes worse than the perfect case as we move to detecting the last 30% of outliers. As we can see in the plots, after 80% of the outliers, methods have much reduced efficiency, but this is not important (and it is something we expected) because in practical cases there are a few outliers which are very hard to detect.

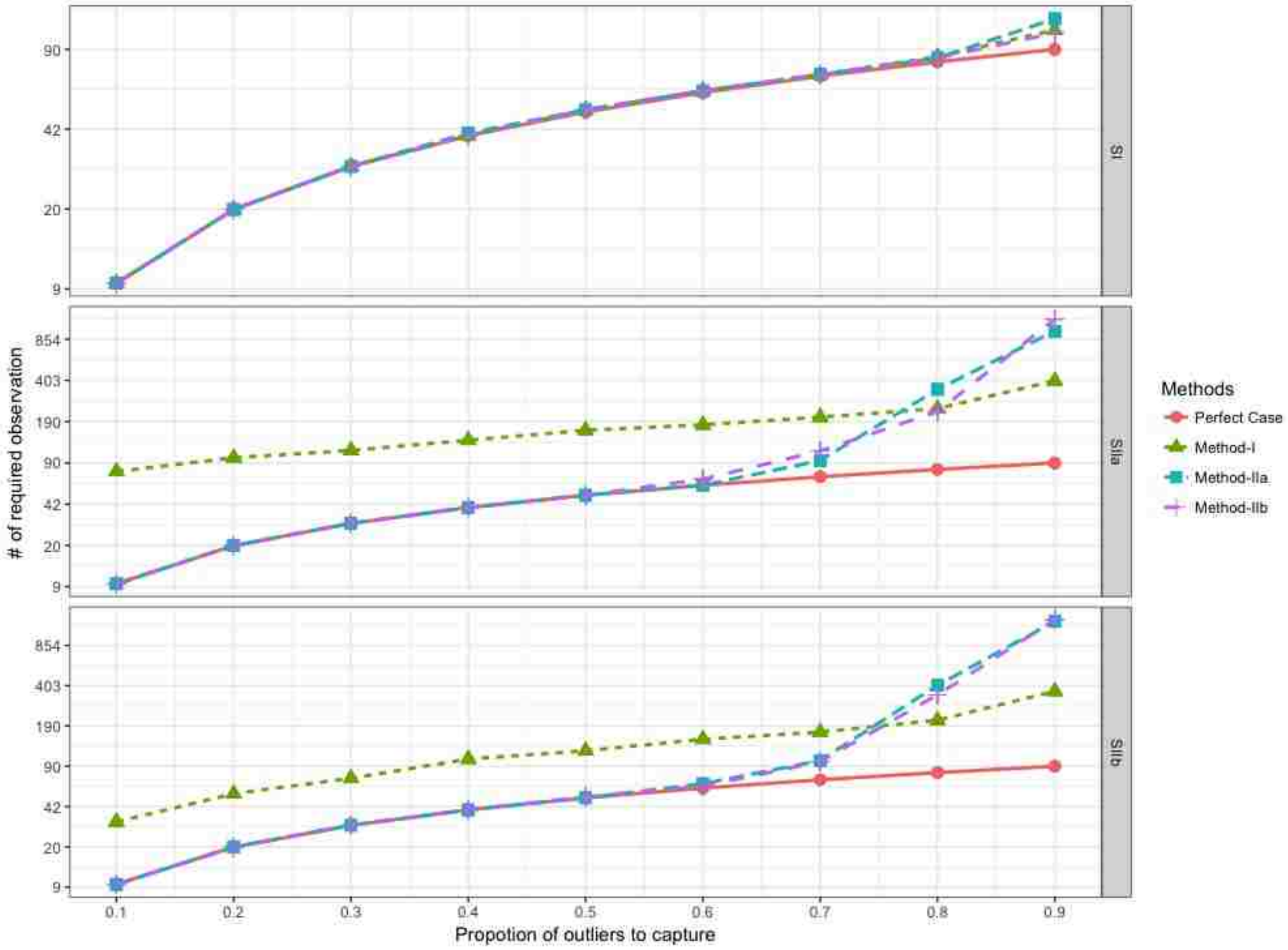


Figure 5.7: The number of observations that should be considered to detect different proportions of the actual outliers on datasets with 100 outliers

We used another measure of comparison, in Table 5.3, where we arrange the posterior mean of ν values for different observations from smallest to largest, and we select only the top 20 and 100 most extreme observations from the selected set as true outliers. As we can see all methods work well with SI. However, in SIIa and SIIb, Method-I did not work well. The performances of Method-IIa and Method-IIb are very comparable, and they could detect between 70–75% of the true outliers. This indicates that the actual choice of the method is not sensitive to the (type of) dataset.

Table 5.3: Number of outliers in first 20 and 100 ranked ν values

	Simulation Models	Method-I	Method-IIa	Method-IIb
20 outliers	SI	**18**	17	17
	SIIa	1	**15**	14
	SIIb	1	14	**15**
100 outliers	SI	**88**	87	87
	SIIa	27	**72**	70
	SIIb	38	71	**73**

5.3.5 Criterion for Outlier Determination

Now, we talk about ways to identify outliers. There are two ways to do this. First, we can choose to throw away a certain percentage, *e.g.* p, of observations as potential outliers. In that case, we will rank $E(\nu|Data)$ for all the observations from smallest to largest and throw away the top p fraction of the observations based on the ranking. Second, we can set a threshold ν_0 on the posterior mean of ν_i, as discussed before. For a specific choice of ν_0, we can compute its sensitivity and specificity for the simulated datasets. The sensitivity will be computed as a fraction of true outliers that were detected by the thresholding. Specificity will be determined as a fraction of non-outliers that remain in the data after thresholding. If we increase ν_0, we expect sensitivity to increase and specificity to decrease. In our simulation models, we investigate three different ν_0 values which are 5, 10 and 20. In Table 5.4, a comparison has been made for all methods across all the simulation datasets. We see that if we increase the value of ν_0 the sensitivity increases, but the specificity decreases. In SI, all methods are comparably perfect across the three methods. In SIIa and SIIb, Method-I performs very poorly, but Method-IIa and Method-IIb perform well. Also from the table, we see that the performance is reduced when we have a larger number of outliers.

Table 5.4: Sensitivity and specificity comparison for different choices of ν_0

	Simulation Models	ν_0	Number of Outliers	Method-I	Method-IIa	Method-IIb
20 outliers datasets	SI	5	14	(0.70 , 0.999)	(0.70 , 0.999)	(0.70 , 0.999)
		10	17	(0.85 , 0.999)	(0.85 , 0.999)	(0.85 , 0.999)
		20	18	(0.90 , 0.996)	(0.90 , 0.995)	(0.90 , 0.996)
	SIIa	5	15	(0.20 , 0.958)	(0.75 , 1)	(0.60 , 1)
		10	15	(0.50 , 0.941)	(0.75 , 0.001)	(0.65 , 0.996)
		20	15	(0.75 , 0.933)	(0.75 , 0.999)	(0.75 , 0.992)
	SIIb	5	9	(0.05 , 0.984)	(0.45 , 0.999)	(0.45 , 0.999)
		10	14	(0.30 , 0.976)	(0.70 , 0.998)	(0.50 , 0.998)
		20	15	(0.75 , 0.923)	(0.75 , 0.990)	(0.70 , 0.992)
100 outliers datasets	SI	5	33	(0.30 , 1)	(0.31 , 1)	(0.34 , 1)
		10	62	(0.56 , 0.999)	(0.60 , 0.999)	(0.59 , 0.999)
		20	84	(0.82 , 0.998)	(0.84 , 0.998)	(0.83 , 0.998)
	SIIa	5	63	(0.07 , 0.971)	(0.63 , 1)	(0.54 , 1)
		10	65	(0.20 , 0.958)	(0.65 , 1)	(0.59 , 0.998)
		20	67	(0.53 , 0.939)	(0.67 , 0.998)	(0.65 , 0.995)
	SIIb	5	55	(0.09 , 0.988)	(0.50 , 0.999)	(0.48 , 1)
		10	61	(0.19 , 0.982)	(0.61 , 0.999)	(0.53 , 0.999)
		20	65	(0.36 , 0.970)	(0.64 , 0.994)	(0.64 , 0.996)

5.4 Comparison Against Existing Methods

In previous literature in the field, differing methods for detection of outliers in linear regression are available. We first review the two most commonly used methods, and then we compare them with the methods that we have developed in Section 5.2.

5.4.1 Bonferroni Outlier Test

We are going to refer to the Bonferroni Outlier Test as the BO-Test. The idea of this method is to use Studentized residuals for each observation, which are defined as $t_i = \frac{e_i}{MSE_{(i)}(1-h_{ii})}$, where $MSE_{(i)}$ is the mean-square error from the regression model fitted with the i^{th} observation deleted. In this case, e_i and $MSE_{(i)}$ are independent, and can be shown as $t_i \sim t_{n-p-2}$. The loss of the extra one df is due to the deletion of observation i.

BO-Test reports the Bonferroni p-values for Studentized residuals in linear models based on the t-test (Fox and Weisberg, 2011). A very small p-value means the observation is highly likely to be an outlier, and a large p-value indicates the observation is more fitting with the rest of the data. Hence, to rank how extreme the observation is, we should arrange the p-values from smallest to largest, or equivalently, we can arrange the Studentized residuals in the order of their absolute values from large to small.

If we want to give a rule for detecting outliers, we set a threshold value, *e.g.* p_0, and for all observations with p-values less than p_0, we call them outliers. The usual choices of the thresholds are 0.01 and 0.05. This method was implemented using outlierTest in R-package CAR (https://www.rdocumentation.org/packages/car).

5.4.2 Bayesian Test for Outliers Detection

Another method for detecting outliers in linear regression is based on the Bayesian approach (Chaloner and Brant, 1988). We are going to refer to this method as Ch-Br. In Ch-Br, the outlyingness of an observation is based on values of $\left|\frac{\epsilon_i}{\sqrt{var(\epsilon_i)}}\right|$, where $\epsilon_i = y_i - x_i^T\beta$. Ch-Br has additional flexibility because we do not need to use the same variance for all data points. This means each observation $var(\epsilon_i)$ can be different. We actually use different quantities for variances based on which simulation model we think is correct. For example, if we think Method-I is reasonable to be compared with Ch-Br, we use the $var(\epsilon_i)$ to be σ^2, and we use $var(\epsilon_i)$ to be $\sigma^2 f(x_i)^T \alpha$ if we think Method-II is a reasonable comparison. Since X and y are known, at each iteration, we are going to look at samples of β and α, and use them to compute the value $\left|\frac{y_i - x_i^T \beta}{\sqrt{\sigma^2 f(x_i)^T \alpha}}\right|$. We compute the posterior mean of this quantity for each observation, and by arranging them from largest to smallest, we get a ranking of outliers. The largest value is the most extreme, and the second largest is the second most extreme, and so on.

To specify a rule for detecting outliers, we follow the recommendation of Chaloner and Brant (1988). They suggested setting a threshold k_0 and computing the posterior probability $q_i = P\Big(\Big|\frac{\epsilon_i}{\sqrt{var(\epsilon_i)}}\Big| > k_0 \Big| Data\Big)$ for each i. The corresponding prior probability would be $2\Phi(-k_0)$; hence any observations with $q_i < 2\Phi(-k_0)$ is going to be considered an outlier.

5.4.3 Comparison of Simulation Datasets

In all the plots and tables, as mentioned before, when we compare Ch-Br against one of our methods, we use the corresponding specification of $var(\epsilon_i)$. In Table 5.5, we compare our methods with BO-Test and Ch-Br, and we reported the number of detected outliers on the simulation datasets that we discussed in Section 5.3 in the first 20 and 100 observations. All methods perform well for SI datasets, because for that particular model the error variance does not depend on X variables. But, for SIIa and SIIb models, BO-Test performs significantly poorer compared with the others. For our methods and Ch-Br, the performance is comparable in both, and our methods perform slightly better than Ch-Br in some cases. Similar results can be seen in Figures 5.8 and 5.9. As we can see, the curve of our methods are slightly below the curve of Ch-Br. Now, if we want to detect the most extreme outlier, 0.9 of outliers, the performance of the methods becomes comparable, and that is expected, because these are outliers which are most heavily mixed with other regular data points, so they are more difficult to detect. However, for most outliers, our method performs as well as, and in some cases better, than Ch-Br method.

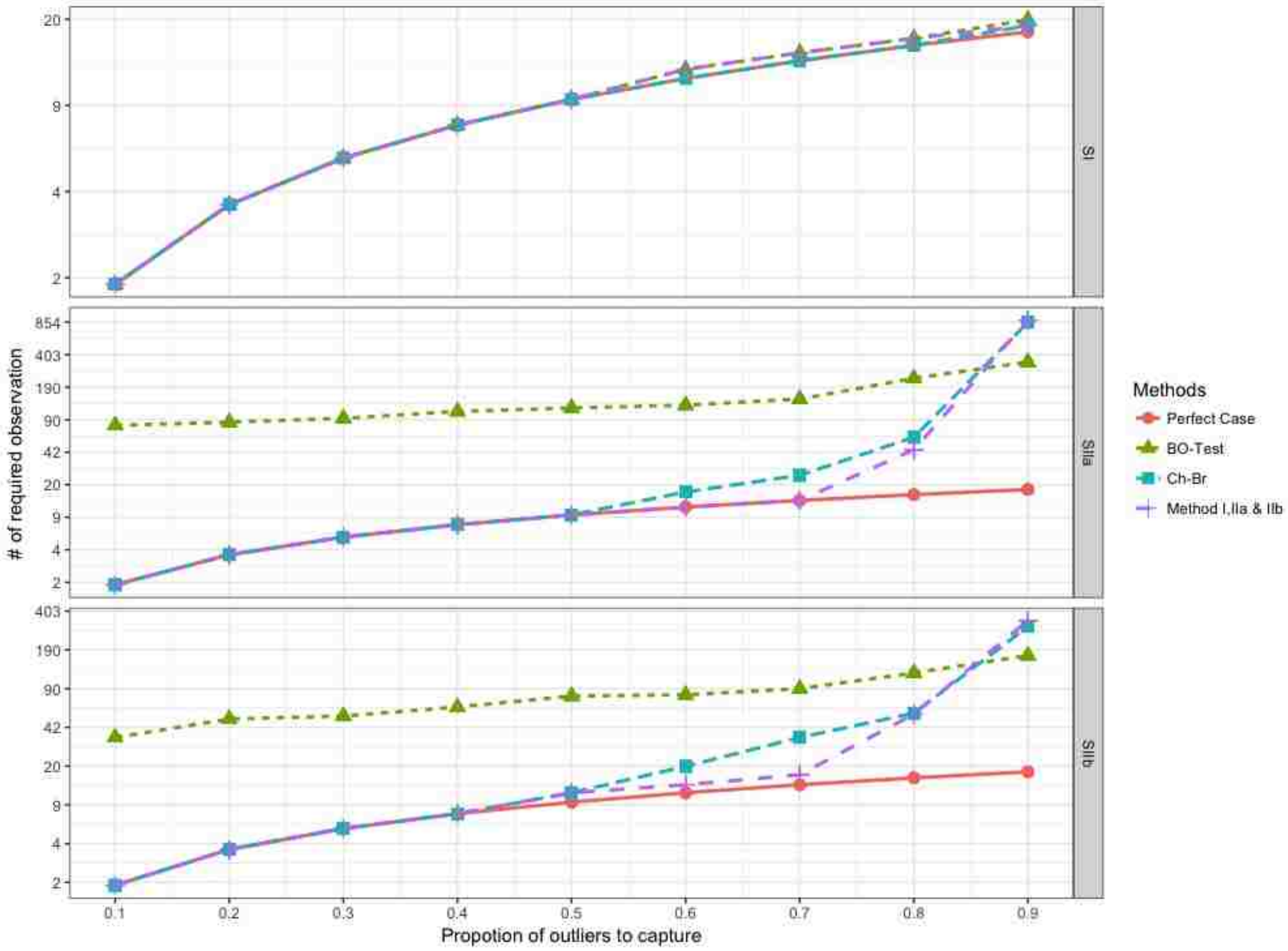


Figure 5.8: The number of observations that should be considered to detect different proportions of the actual outliers on datasets with 20 outliers

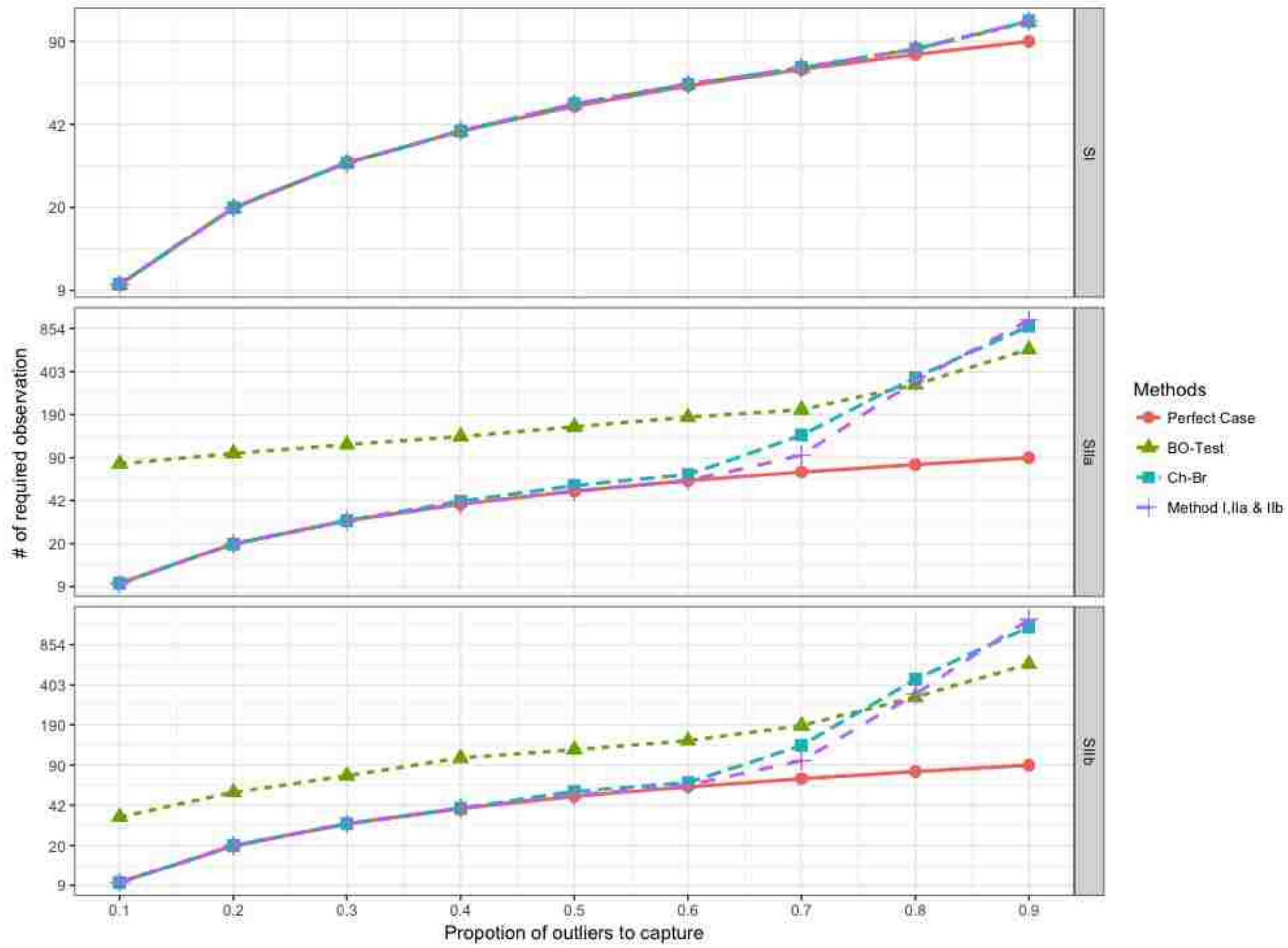


Figure 5.9: The number of observations that should be considered to detect different proportion of the actual outliers on datasets with 100 outliers

Table 5.5: Number of outliers in first 20 and 100 observations

Number of outliers	Models	Method-I, IIa & IIb	BO-Test	Ch-Br
20 outliers	SI	18	18	18
	SIIa	15	1	15
	SIIb	15	2	13
100 outliers	SI	88	91	87
	SIIa	72	21	68
	SIIb	73	38	69

5.5 Application to a Real Dataset

5.5.1 Data Description

The goal of this section is to identify potential outliers in the longitudinal datasets of student heights. The dataset includes recorded height of $53,206$ Arkansas students each measured at four different grades – KG, GR2, GR4 and GR6. These students were admitted in the KG between 2005–2008. We considered data for only those students for whom we have all four measurements up to GR6. Here, the height represents the response, and it has only one covariate which is gender. Due to inappropriate measurements, incorrect data entry, and bias from data recorded by different individuals, the dataset could have outliers. Before doing any analysis these outliers should be removed from the dataset to avoid misleading inference. We are going to apply the methods that we have developed in the previous sections on the real dataset, but first we need to formulate the response appropriately as in Section 5.5.2.

5.5.2 Formulation of the Problem

We denoted the height of the student i at grade t by $H_i(t)$. The outliers exist whenever we have a large change between two data points, and hence our goal is to detect the large change between two consecutive measurements for an individual. For a particular student, we denoted the difference of heights in two consecutive grades by $\Delta_i(t)$, i.e., $\Delta_i(t) = H_i(t) - H_i(t-1)$. Any negative or large positive value of $\Delta_i(t)$ is considered to be an outlier. We see that $\Delta_i(t)$ does not follow a symmetric distribution. We want to use an error distribution property that is symmetric around 0. Instead of working with actual heights, we want to work with some kind of relative height that can increase or decrease over time. So, we used Z-scores that are computed using LMS method. The LMS parameters are the power in the Box-Cox transformation (L), the median (M), and the generalized coefficient of variation

(S). The LMS method involves the following formula:

$$Z_i(t) = \begin{cases} \frac{\left((H_i(t)/M)^L\right)-1}{LS}, & \text{where } L \neq 0 \\ \frac{\ln\left(H_i(t)/M\right)}{S}, & \text{where } L = 0 \end{cases}$$

(L, M and S are the values from the appropiate table corresponding to the age in months of the child.) The values of the LMS parameters change based on age and gender as specified in Centers of Disease Control and Prevention (CDC) 2000 growth chart (Kuczmarski RJ, 2002). $Z_i(t)$ can take any real value between $-\infty$ and ∞. So, we do not have any constraint in the model for $Z_i(t)$. With this new setting, we work with $y_i(t) = Z_i(t) - Z_i(t-1)$, and now our goal is to see if $y_i(t)$ has a large positive or negative change.

$$y_i(t) = \beta_0 + \beta_1 x_{it}^{(1)} + \cdots + \beta_p x_{it}^{(p)} + \epsilon_{it}, \text{ where } \epsilon_{it} \sim t_{\nu_i}(0, \sigma_{it}^2)$$

where $x_{it}^{(1)}, \ldots, x_{it}^{(p)}$ are a set of covariates related to the i^{th} student at t^{th} grade. For our dataset we have only one covariate gender which is 1 for male and 0 for female.

5.5.3 Applying the Methods on the Dataset

We applied Method-I and Method-II on the real dataset. In the two methods, we used the same number of iterations, initial values and prior parameters that we used in our simulation studies, Section 5.3. In Method-II, in the MH part, we used different $\sigma^{proposed}$ values to get an acceptance ratio that lies between 35–45%. We burned-in the first 25% of the posterior samples, and the rest was thinned by 5.

In terms of the number of matches between the two methods, we saw that if we consider 100 and 1000 of the most extreme observations, we found close to 90% matches in both of them. In real data, gender is the only covariate. If Method-I and Method-II are showing strong matches, it implies that gender may not have strong effect on the variability of y.

Furthermore, we applied BO-Test and Ch-Br that we discussed in Section 5.4 on the real dataset and compared them with our methods. As we can see in Table 5.6, our methods match between 85–90% with the other methods. This match indicates there is a significant amount of agreement between our methods and existing methods.

Table 5.6: Number of matches in 100 and 1000 most outlying observations

	BO-Test	Ch-Br
Method-I	(87 , 923)	(92 , 983)
Method-II	(85 , 916)	(86 , 945)

5.5.4 Comparison of Thresholds for Determining Outliers

In each method, we have a different way of specifying the threshold. As we discussed in Sections 5.2 and 5.4, we used ν_0, p_0 and k_0 as threshold parameters for our methods, BO-Test and Ch-Br, respectively. In the following table, we show the values of the threshold parameter that we need to use in different methods for obtaining comparable numbers of outliers.

Table 5.7: Values of threshold parameter for obtaining comparable numbers of outliers

Method	Threshold parameter	Criterion	Number of outliers				
			~ 500	$\sim$ 750	~ 1000	~ 1500	~ 2000
Method-I & II	ν_0	$E(\nu \mid Y) < \nu_0$	2.4	2.6	2.8	3	10
BO-Test	p_0	p-value $< p_0$	10^{-8}	10^{-6}	10^{-5}	10^{-4}	10^{-3}
Ch-Br	k_0	$P\Big(\Big\lvert\frac{\epsilon_i}{\sqrt{var(\epsilon_i)}}\Big\rvert > k_0\Big)$ $< 2\Phi(-k_0)$	6	5	4	3.5	2.5

Since the real dataset is significantly larger than our simulation datasets, we are able to detect a massive amount of outliers compared with what we could detect in our simulation.

5.5.5 Exploring Temporal Pattern

Since our real dataset represents the height of students in 4 consecutive grades, which are KG, GR2, GR4 and GR6, another approach can be to introduce temporal dependence in y. This is equivalent to adding one more covariate to the model:

$$\begin{aligned} Y^{(1)} &= \beta_0^* + \beta_1^* X + \epsilon \\ Y^{(t+1)} &= \beta_0 + \beta_1 X + \beta_2 Y^{(t)} + \epsilon, \text{ for } t = 1, \dots, T-1. \end{aligned}$$

In our real dataset, $T = 3$. In Table 5.8, we present posterior means and 95% credible intervals for β values with and without temporal dependence effect. As we can see, β for temporal effect is insignificant, which is why we expect a strong match between the temporal and non-temporal methods for identifying outliers.

Table 5.8: Posterior mean and 95% credible intervals for β

Temporal dependence	Covariate effect	Method-I	Method-II
with	β_1^*	-0.052 $(-0.058 , -0.046)$	-0.052 $(-0.058 , -0.046)$
	β_1	0.061 (0.057 , 0.066)	0.006 (0.058 , 0.067)
	β_2	0 $(-0.001 , 0.001)$	0 $(-0.001 , 0.001)$
without	β_1	0.020 (0.016 , 0.024)	0.021 (0.017 , 0.025)

We applied the time series approach for both Method-I and Method-II, and we denoted them by Method-I-T and Method-II-T, respectively. We made a comparison among our times series and regular method for the first 100 and 1000 observations. As we can see in Table 5.9, the number of matches is around 90%.

Table 5.9: Number of matches outliers in 100 and 1000 most outlying observations

	Method-I-T	Method-II-T
Method-I	(89 , 956)	(94 , 950)
Method-II	(87 , 924)	(88 , 941)
Method-I-T	–	(87 , 960)

When we ran time series techniques on BO-Test and Ch-Br methods, we did not get significantly different results, as we can see in Tables 5.10 and 5.6.

Table 5.10: Number of matches in 100 and 1000 most outlying observations

	BO-Test-T	Ch-Br-T
Method-I-T	(88 , 961)	(93 , 987)
Method-II-T	(86 , 953)	(89 , 964)

5.5.6 Analysis of Outlying Observations

Since our methods gave similar results, we chose one of the methods (Method-I) to explore properties of outliers. It has been observed that for some students two data points were identified as outliers. This can happen if the erroneous measurement is at one of the intermediate grades like GR2 or GR4. In that case, the change of Z-score between the grades immediately before and after the erroneous measurements will have unusually large values indicating both of them as outliers. On the other hand, if the erroneous measurements were recorded in the KG or GR6, then only one change in Z-score will be large, so one outlier will be detected. When we considered the top 1000 most extreme observations, we found around 80% of them come from unique students which implies about 20% of the students have more than one unusually large change, which is potentially caused by erroneous measurement at an intermediate grade.

Below, we plot Z-score for three students depending on the time of erroneous measurement. For the student with the most extreme outlier (denoted by Student-A), the erroneous measurement occurred at grade KG. For the student with the 3^{rd} and 5^{th} most extreme outliers (denoted by Student-B), the error occurred at GR2. For the student with the 6^{th} most extreme outlier (denoted by Student-C) the error occurred at GR6. The line plots of the students' Z-scores are shown in Figure 5.10. We can also see the change in Z-score for the student with most extreme outlier is significantly large compared to the change for the other two students' outlying observations.

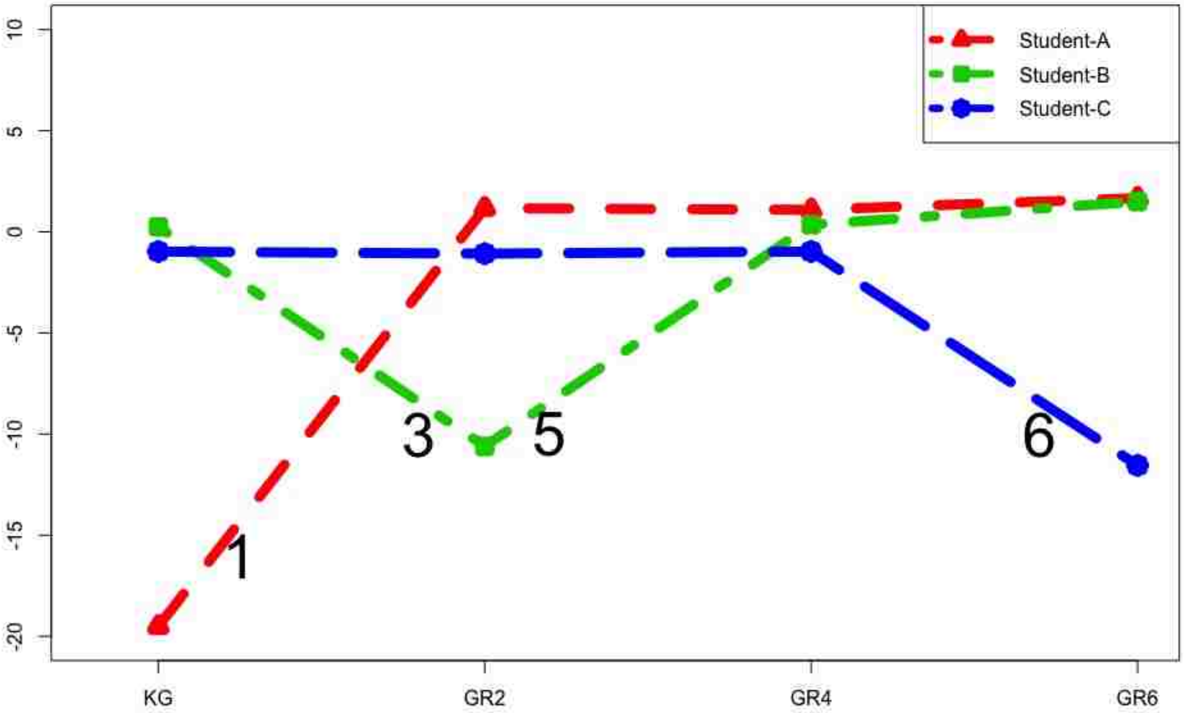


Figure 5.10: The time series of Z-scores for 3 students (The numbers indicate the rank of changes w.r.t. all outliers)

Figure 5.11 shows an interesting case where all three changes are unusually large, and are ranked within the top 600. We found the rank of 1^{st} change is 44, the rank of 2^{nd} change is 540, the rank of 3^{rd} change is 317.

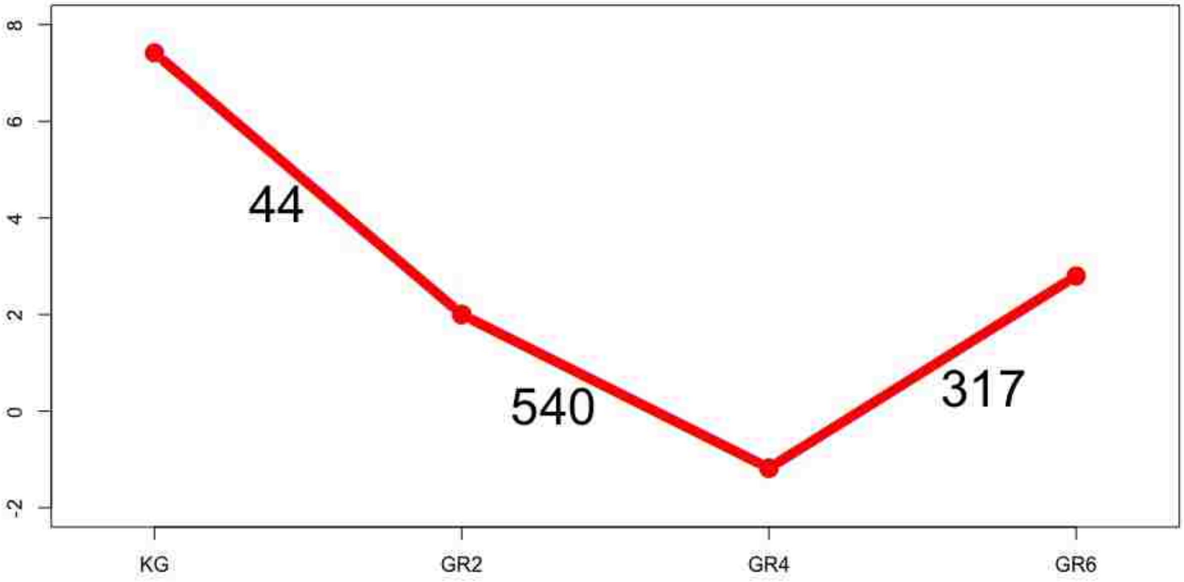


Figure 5.11: Z-scores from one student with unusually large values between all grades (The numbers indicate the rank of changes w.r.t. all outliers)

Moreover, we checked the distribution of most outlying 1000 and 10000 for gender and grades. The values were reported in Table 5.11. We saw that for both 1000 and 10000 most outlying observations the number of such outliers is relatively low in Female between GR2 and GR4 and relatively high between GR4 and GR6, and in the other groups the error is relatively uniform.

Table 5.11: The 1000 and 10000 most outlying observations across different gender and grades

	Detection of unusual change between		
Gender	KG → GR2	GR2 → GR4	GR4 → GR6
Female	(166, 1500)	(109, 856)	(209, 2060)
Male	(181, 1686)	(152, 1932)	(183, 1965)

5.6 Conclusion

In this chapter, we have developed hierarchical models for outlier detection using heavy tailed residuals. Instead of a hard determination of outliers, our methods provide an ordering of outlyingness of all observations. If one wants to specifically identify outliers to eliminate, we discussed ways of doing that in Section 5.3.5.

For real data on student heights, currently the CDC uses threshold on values of Z-score to identify outliers based on Biologically Implausible Values (BIV) as explained in "Modified Z-Scores in the CDC Growth Charts" available at (https://www.cdc.gov/nccdphp/dnpao/growthcharts/resources/biv-cutoffs.pdf). However, there maybe observations that have Z-scores with the usual range at all time points, but between two successive time points are unusual. Our approach can identify those measurements which may not be detected using BIV criterion. As we have seen, the position of outlying changes between two measurements may give a different picture about the grade of outlying measurement. Hence, it is always advisable to biologically correlate the model output for a complete understanding of the outliers present in the data.

Bibliography

Aggarwal, C. C. (2013), "An introduction to outlier analysis," in *Outlier analysis*, pp. 1–40, Springer.

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

Andrieu, C., De Freitas, N., Doucet, A., and Jordan, M. I. (2003), "An introduction to MCMC for machine learning," *Machine learning*, 50, 5–43.

Babbar, S. and Chawla, S. (2010), "On Bayesian Network and Outlier Detection." in *COMAD*, p. 125.

Banerjee, S., Gelfand, A. E., Finley, A. O., and Sang, H. (2008), "Gaussian predictive process models for large spatial data sets," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70, 825–848.

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2014), *Hierarchical modeling and analysis for spatial data*, Crc Press.

Barghash, A., Arslan, T., and Helms, V. (2016), "Robust Detection of Outlier Samples and Genes in Expression Datasets," *Journal of Proteomics and Bioinformatics*, 9, 38–48.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970), "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *The annals of mathematical statistics*, 41, 164–171.

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

Bhark, E. W. (2011), *Multiscale spectral-domain parameterization for history matching in structured and unstructured grid geometries*, Texas A&M University.

Bouckaert, R., Heled, J., Kühnert, D., Vaughan, T., Wu, C.-H., Xie, D., Suchard, M. A., Rambaut, A., and Drummond, A. J. (2014), "BEAST 2: a software platform for Bayesian evolutionary analysis," *PLoS computational biology*, 10, e1003537.

Brubaker, M., Salzmann, M., and Urtasun, R. (2012), "A family of MCMC methods on implicitly defined manifolds," in *Artificial Intelligence and Statistics*, pp. 161–172.

Bujanović, Z. (2011), "Krylov type methods for large scale eigenvalue computations," Ph.D. thesis, Prirodoslovno matematički fakultet-Matematički odsjek, Sveučilište u Zagrebu.

Calvetti, D., Reichel, L., and Sorensen, D. C. (1994), "An implicitly restarted Lanczos method for large symmetric eigenvalue problems," *Electronic Transactions on Numerical Analysis*, 2, 21.

Chakraborty, A., Gelfand, A. E., Wilson, A. M., Latimer, A. M., and Silander Jr, J. A. (2010), "Modeling large scale species abundance with latent spatial processes," *The Annals of Applied Statistics*, pp. 1403–1429.

Chakraborty, A., Gelfand, A. E., Wilson, A. M., Latimer, A. M., and Silander, J. A. (2011), "Point pattern modelling for degraded presence-only data over large regions," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 60, 757–776.

Chaloner, K. and Brant, R. (1988), "A Bayesian approach to outlier detection and residual analysis," *Biometrika*, 75, 651–659.

Chung, F. R. (1997), *Spectral graph theory*, no. 92, American Mathematical Soc.

Courant, R. and Hilbert, D. (1965), *Methods of mathematical physics [Methoden der mathematischen Physik, engl.] 1*, CUP Archive.

Cullum, J. and Willoughby, R. A. (1981), "Computing eigenvalues of very large symmetric matrices?an implementation of a Lanczos algorithm with no reorthogonalization," *Journal of Computational Physics*, 44, 329–358.

Cuppen, J. (1980), "A divide and conquer method for the symmetric tridiagonal eigenproblem," *Numerische Mathematik*, 36, 177–195.

de Melo, S. N., Matias, L. F., and Andresen, M. A. (2015), "Crime concentrations and similarities in spatial crime patterns in a Brazilian context," *Applied Geography*, 62, 314–324.

Del Moral, P. (2013), "Mean Field Simulation for Monte Carlo Integration Chapman & Hall: London," .

Embree, M. (2009), "The Arnoldi eigenvalue iteration with exact shifts can fail," *SIAM Journal on Matrix Analysis and Applications*, 31, 1–10.

Fox, J. and Weisberg, S. (2011), *An R companion to applied regression*, Sage Publications.

Francis, J. G. (1962), "The QR transformation?part 2," *The Computer Journal*, 4, 332–345.

Gelfand, A. E., Silander, J. A., Wu, S., Latimer, A., Lewis, P. O., Rebelo, A. G., Holder, M., et al. (2006), "Explaining species distribution patterns through hierarchical modeling," *Bayesian Analysis*, 1, 41–92.

Geman, S. and Geman, D. (1984), "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on pattern analysis and machine intelligence*, pp. 721–741.

Gilks, W. R. (2005), "Markov chain monte carlo," *Encyclopedia of Biostatistics.*

Hastings, W. K. (1970), "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, 57, 97–109.

Hatch, J. P. and Prihoda, T. J. (1992), “The effect of influential outliers on parameter estimation in regression analysis,” *Biofeedback and self-regulation*, 17, 153–156.

Hawkins, D. M. (1980), *Identification of outliers*, vol. 11, Springer.

Higdon, D. (1998), “A process-convolution approach to modelling temperatures in the North Atlantic Ocean,” *Environmental and Ecological Statistics*, 5, 173–190.

Indurkhya, A., Gardiner, J. C., and Luo, Z. (2001), “The effect of outliers on confidence interval procedures for cost-effectiveness ratios,” *Statistics in medicine*, 20, 1469–1477.

Kuczmarski RJ, Ogden CL, G. S. e. a. (2002), “2000 CDC Growth Charts for the United States: Methods and Development,” *National Center for Health Statistics. Vital Health Stat*, 11(246).

Kwak, S. K. and Kim, J. H. (2017), “Statistical data preparation: management of missing values and outliers,” *Korean journal of anesthesiology*, 70, 407–411.

Lawson, A. B. and Denison, D. G. (2002), *Spatial cluster modelling*, CRC press.

Lee, D. (2013), “CARBayes: An R package for Bayesian spatial modeling with conditional autoregressive priors,” *Journal of Statistical Software*, 55, 1–24.

Lehoucq, R. B., Sorensen, D. C., and Yang, C. (1998), *ARPACK users’ guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*, vol. 6, Siam.

Li, A., Feng, M., Li, Y., and Liu, Z. (2016), “Application of outlier mining in insider identification based on Boxplot method,” *Procedia Computer Science*, 91, 245–251.

Lin, S.-J. and Huang, M.-T. (2002), “Estimating Jump-Diffusion Models Using the MCMC Simulation, National Tsing Hua University Department of Economics NTHU Working Paper Series,” Tech. rep., Working paper.

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, 1087–1092.

Moller, J. and Waagepetersen, R. P. (2003), *Statistical inference and simulation for spatial point processes*, CRC Press.

Neal, R. M. (2003), “Slice sampling,” *Annals of statistics*, pp. 705–741.

Nucci, L. B., Souccar, P. T., and Castilho, S. D. (2016), “Spatial data analysis and the use of maps in scientific health articles,” *Revista da Associação Médica Brasileira*, 62, 336–341.

Pandey, S., Billor, N., and Turkmen, A. (2008), “The effect of outliers in independent component analysis,” *American Journal of Mathematical and Management Sciences*, 28, 399–418.

Parlett, B. N. (1980), "A new look at the Lanczos algorithm for solving symmetric systems of linear equations," *Linear algebra and its applications*, 29, 323–346.

Parlett, B. N. and Scott, D. S. (1979), "The Lanczos algorithm with selective orthogonalization," *Mathematics of computation*, 33, 217–238.

Plummer, M., Best, N., Cowles, K., and Vines, K. (2006), "CODA: convergence diagnosis and output analysis for MCMC," *R news*, 6, 7–11.

Qiu, Y. and Mei, J. (2016), "RSpectra: Solvers for Large Scale Eigenvalue and SVD Problems," *URL https://CRAN. R-project. org/package= RSpectra. R package version 0.12-0.*

Rampaso, R. C., de Souza, A. D. P., and Flores, E. F. (2016), "Bayesian analysis of spatial data using different variance and neighbourhood structures," *Journal of Statistical Computation and Simulation*, 86, 535–552.

Rebelo, A. (2002), "The state of plants in the Cape Flora," in *Proceedings of a Conference Held at the Rosebank Hotel in Johannesburg (GH Verdoorn and J. Le Roux, eds.)*, vol. 18.

Rebelo, A. G. (2001), *Proteas: a field guide to the Proteas of Southern Africa*, Fernwood Press, Vlaeberg, SA.

Richardson, S. and Green, P. J. (1997), "On Bayesian analysis of mixtures with an unknown number of components (with discussion)," *Journal of the Royal Statistical Society: series B (statistical methodology)*, 59, 731–792.

Ridout, M., Demétrio, C. G., and Hinde, J. (1998), "Models for count data with many zeros," in *Proceedings of the XIXth international biometric conference*, vol. 19, pp. 179–192.

Robert, C. P. (2004), *Monte carlo methods*, Wiley Online Library.

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

Sleijpen, G. L. and Van der Vorst, H. A. (2000), "A Jacobi–Davidson iteration method for linear eigenvalue problems," *SIAM review*, 42, 267–293.

Sorensen, D. C. (1992), "Implicit application of polynomial filters in ak-step Arnoldi method," *Siam journal on matrix analysis and applications*, 13, 357–385.

Stein, M. L., Chi, Z., and Welty, L. J. (2004), "Approximating likelihoods for large spatial data sets," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66, 275–296.

Thomas, K. (2013), "Poverty and Obesity in America: How They Map," *blogorrhea*, http://asserttrue.blogspot.com/2013/02/poverty-and-obesity-in-america-how-they.html.

Ver Hoef, J. M., Cressie, N., Fisher, R. N., and Case, T. J. (2001), "Uncertainty and spatial linear models for ecological data," in *Spatial Uncertainty in Ecology*, pp. 214–237, Springer.

Walker, H. F. (1988), “Implementation of the GMRES method using Householder transformations,” *SIAM Journal on Scientific and Statistical Computing*, 9, 152–163.

Weston, D. J., Adams, N. M., Russell, R. A., Stephens, D. A., and Freemont, P. S. (2012), “Analysis of spatial point patterns in nuclear biology,” *PLoS One*, 7, e36841.

Xu, S., Lu, B., Bell, N., and Nixon, M. (2017), “Outlier Detection in Dynamic Systems with Multiple Operating Points and Application to Improve Industrial Flare Monitoring,” *Processes*, 5, 28.

Yang, S., Guo, X., Yang, Y.-C., Papcunik, D., Heckman, C., Hooke, J., Shriver, C. D., Liebman, M. N., and Hu, H. (2006), “Detecting outlier microarray arrays by correlation and percentage of outliers spots,” *Cancer informatics*, 2, 117693510600200017.

Zhu, L., Carlin, B. P., and Gelfand, A. E. (2003), “Hierarchical regression with misaligned spatial data: relating ambient ozone and pediatric asthma ER visits in Atlanta,” *Environmetrics*, 14, 537–557.