Master's Thesis in Biostatistics (STA495)

Multiresolution Decomposition of Incomplete Random Signals

A Statistical Application of Sparse Matrix Calculus

Supervised by Prof. Dr. Reinhard Furrer

by Roman Charles Flury
07-923-659

Master Program in Biostatistics
University of Zurich

University of Zurich[UZH]

Version November 16, 2017

Contents

Preface **iii**

Abstract **v**

1 Motivation **1**
1.1 Scale space multiresolution analysis of random signals 1
1.2 Summary 9

2 Data **11**
2.1 Lägeren mountain 11
2.2 Oral cavity cancer 14

3 Methods **15**
3.1 Multiresolution decomposition of an incomplete signal 15
3.2 Eigenvalue approximation 21

4 Applications **29**
4.1 Multiresolution decomposition of the Lägeren data 29
4.2 Multiresolution decomposition of irregular lattice data 36

5 Conclusion **39**
5.1 Results 39
5.2 Outlook 40

Appendices

Appendix A Additional calculus **41**
A.1 Gaussian Markov random field 41
A.2 Canonical parameterization of a GMRF 41
A.3 Derivation of multivariate t-distribution parameters 41

Appendix B Arnoldi algorithm **43**
B.1 Spam R wrapper function 43
B.2 Spam Fortran routine 50

Appendix C Additional figures **53**

Appendix D Session Info **57**

References **59**

Preface

I would like to thank my supervisors Prof. Dr. Reinhard Furrer as well as Florian Gerber for all their support and inspiring ideas. It has been exiting to work with you, I could profit in every possible aspect. A special thanks goes to Fabian Schneider from the Remote Sensing Laboratories of the Geography Department of the University of Zurich, for providing the intriguing data used in this thesis. In addition, I would like to thank my family and my friends for supporting me during my master degree and my studies in general. In particular I thank Agata Guirard, who gave me the strength to pursue my degree.

Roman Flury
November 2017

Abstract

In scientific applications one often meets eigenvalue problems of matrices with large dimensions, but with most entries equal to zero. Such matrices are accurately called sparse matrices. The non zero entries of a sparse matrix are typically not distributed randomly across the matrix, but they lay on the diagonal and on several higher order diagonals. Instead of calculating the eigenvalues of such sparse matrices with a common approach, algorithms which utilize its sparse structure are more efficient.

For this master thesis, the implicitly restarted Arnoldi method, was implemented in the statistical software `R`, or more specifically in the `R` package `spam`. The algorithm provides the functionality to efficiently determine a set of eigenvalues with the largest or smallest absolute values. The implementation consists basically of wrapper functions of Fortran routines from the ARPACK library, which is well tested and integrated in various scientific procedures.

As an application, a statistical method to decompose random signals into multiple resolutions, was chosen. This method is in particular applicable, since current implementations do rely on eigenvalue calculations, based on fast Fourier transformation. By changing the underneath structure, to a sparse matrix based one, the method could be freed of limitations in its applicability for signals containing missing values and signals distributed on non regular lattices. To account for possible missing values in the input signal, different models were developed to resample the input signal. Furthermore, to sustain efficiently the whole spectrum of eigenvalues of specific precision matrices, an approximation by circulant matrices was developed.

Statistical applications do typically use data to calculate estimates and corresponding confidence or credibility intervals, therefore also suitable data is preluded and discussed proportionately.

All computations, visualizations as well as simulations were done with the statistical programming language `R` by the R Development Core Team (2017) and its additional packages, `fields`, `mrbsizeR` and `spam`.

Chapter 1

Motivation

In this chapter the scale space multiresolution analysis of random signals, proposed by Holmström *et al.* (2011), is introduced and demonstrated on a simple and intuitive example. Along this thesis, this scale space multiresolution analysis is also referred to as multiresolution decomposition. This method poses the statistical application of sparse matrix calculus or more precisely a statistical method for which it can be reasonable to use sparse structure based arithmetic and corresponding eigendecompositions. To motivate an extension and generalization of this, it is emphasized where in current implementations, efficient calculations do limit the applicability of the method, to regular and complete lattice data.

1.1 Scale space multiresolution analysis of random signals

Holmström *et al.* (2011) proposed this method to capture scale dependent features in a random signal, e.g. a spatial field defined on a regular lattice or an image. It smooths the field on multiple scales and considers differences of the smooths to identify eminent features of the data. Credible features are inferred in a Bayesian framework and statistically significant determined for the corresponding smoothing levels. The method can basically be divided in the following three steps.

i Bayesian signal reconstruction.

ii Forming of scale-dependent details using differences of smooths at neighboring scales.

iii Posterior credibility analysis of the different smooths.

This work was originally implemented in Matlab, the language of technical computing, by Holmström and Pasanen (2011) and transferred by Schuster (2017) into the statistical programming language `R`, namely in the package `mrbsizeR`, whereupon the latter is used exclusively throughout this thesis.

1.1.1 Bayesian signal reconstruction

In a realistic scenario, the random signal, i.e. the observed data contains noise due to measurement uncertainties or inaccuracies. To account for those, one assumes that the observed signal

$\boldsymbol{y}$ comprises the true unobserved underlying signal $\boldsymbol{x}$ and the noise $\boldsymbol{\varepsilon}$. In this sense, the model can be stated as

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\varepsilon}. \tag{1.1}$$

Furthermore, it is assumed that the errors $\boldsymbol{\varepsilon}$ are non systematic and normally distributed, with zero mean and an unknown variance $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)^T \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$. The observed data $\boldsymbol{y}$ is assumed to be normal, with $\boldsymbol{y}|\boldsymbol{x}, \sigma^2 \sim \mathcal{N}(\boldsymbol{x}, \mathbf{I}\sigma^2)$. The resulting likelihood function of the model corresponds then to

$$\pi\left(\boldsymbol{y}|\boldsymbol{x}, \sigma^2\right) \propto \sigma^{-n} \exp\left(-\frac{1}{2\sigma^2}||\boldsymbol{y} - \boldsymbol{x}||^2\right).$$

For the positive valued unknown variance σ^2, a Scaled-Inv-$\chi^2(\nu_0, \sigma_0^2)$ prior is used, with the corresponding probability density function (Gelman *et al.*, 1995),

$$\pi(x) = \frac{1}{\Gamma\left(\frac{\nu_0}{2}\right)} \left(\frac{\nu_0 \sigma_0^2}{2}\right)^{\nu_0/2} x^{-(\nu_0/2+1)} \exp\left(-\frac{\nu_0 \sigma_0^2}{2x}\right).$$

The prior of the underlying signal $\boldsymbol{x}$, is an IGMRF (Rue and Held, 2005) of the form

$$\pi(\boldsymbol{x}|\lambda_0, \sigma^2) \propto \left(\frac{\lambda_0}{\sigma^2}\right)^{(n-1)/2} \exp\left(-\frac{\lambda_0}{2\sigma^2} \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x}\right), \tag{1.2}$$

where $\boldsymbol{Q}$ is a precision matrix and defined by

$$\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} = \sum_t \left(\sum_{s \sim t} x_s - 4x_t\right)^2. \tag{1.3}$$

If two lattice locations s and t are neighbors, one writes $s \sim t$.

The resulting marginal posterior $\pi(x|y)$ follows then a multivariate t-distribution

$$\pi\left(\boldsymbol{x}|\boldsymbol{y}\right) \propto \left(\lambda_0 \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + ||\boldsymbol{y} - \boldsymbol{x}||^2 + \nu_0 \sigma_0^2\right)^{-((2n+\nu_0-1)/2)}, \tag{1.4}$$

for a more detailed derivation see Erästö and Holmström (2005).

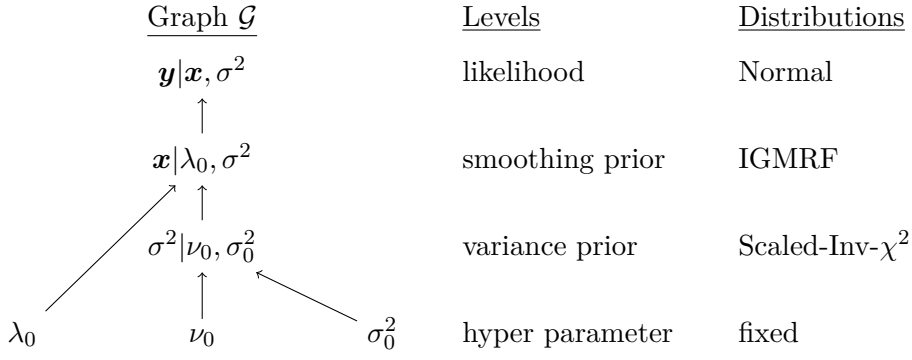

Figure 1.1: Summary of the discussed Bayesian hierarchical model.

Since, the marginal posterior is of closed form $\mathrm{t}_\nu(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, one can easily sample from it. It is only necessary to know the parameters $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ and ν,

$$\boldsymbol{\mu} = (\mathbf{I} + \lambda_0 \boldsymbol{Q})^{-1} \boldsymbol{y}, \tag{1.5}$$

$$\boldsymbol{\Sigma} = (\mathbf{I} + \lambda_0 \boldsymbol{Q})^{-1} \left(\frac{\boldsymbol{y}^T \boldsymbol{y} - \boldsymbol{y}^T \boldsymbol{\mu} + \nu_0 \sigma_0^2}{\nu} \right), \tag{1.6}$$

$$\nu = \nu_0 + n - 1,$$

see Appendix A.3 for a detailed derivation.

Precision matrix $\boldsymbol{Q}$ Through equation (1.3) follows, that for an interior lattice location t, the sum over s in $\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x}$ is the discrete Laplace operator computed at t. The discrete Laplacian is calculated as sum of differences over the nearest neighbors of a lattice location (Reuter *et al.*, 2009). This operator is often used in image procession, for instance in edge detection and motion estimation applications. To assure that there are four neighbors at boundary locations, the boundary values of $\boldsymbol{x}$ are extended beyond the original lattice (Holmström and Pasanen, 2012). The matrix $\boldsymbol{Q}$ is therefore modified by Neumann boundary conditions and thus the rank of the precision matrix is $\operatorname{rank}(\boldsymbol{Q}) = n - 1$, which is also reflected in equation (1.2). This construction can be written in terms of matrix operators $\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} = ||\boldsymbol{C}\boldsymbol{x}||^2$, such that $\boldsymbol{Q} = \boldsymbol{C}^T \boldsymbol{C}$, where the matrix $\boldsymbol{C}$ can be interpreted as the discrete Laplace operator. Under the assumptions that the data is distributed across a regular lattice and moreover, no location on the lattice has missing values, the Laplacian operator can be calculated. According to the construction of $\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x}$.

One notes further, that the parameters of the marginal posterior distribution are determined through the precision matrix $\boldsymbol{Q}$. If the eigendecomposition of $\boldsymbol{Q}$ exists, then one can write

$$\boldsymbol{Q} = \sum_{j=1}^{n} \gamma_j \boldsymbol{v}_j \boldsymbol{v}_j^T. \tag{1.7}$$

Such that $0 \leq \gamma_1 \leq \ldots \leq \gamma_n$ are the eigenvalues of $\boldsymbol{Q}$ and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ are the corresponding orthonormal eigenvectors. The Laplacian operator provides the advantage, that its eigendecomposition can be drastically speed up, with the help of discrete cosine transformations (Strang, 1999). Henceforth, it is for large dimensional problems more efficient, to implement the parameters (1.5) and (1.6) with this approach, instead of calculating matrix vector multiplications naively. Accordingly, it is used in the implementation of Schuster (2017) and as well as in Holmström and Pasanen (2011).

To Illustrate the scale space multiresolution analysis, a simple example is used here. From a standard normal distribution 100 samples are drawn, arranged in a 10×10 matrix and values of some points in the upper left are increased, to create an artificial feature in the sample, as shown in Figure 1.2. This example signal is reconstructed, as discussed above and its resulting mean is displayed in Figure 1.3.

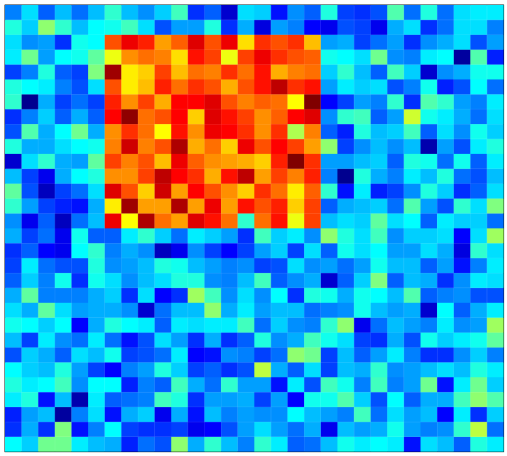

Figure 1.2: Random sample from a standard normal distribution with an artificial feature in the upper left.

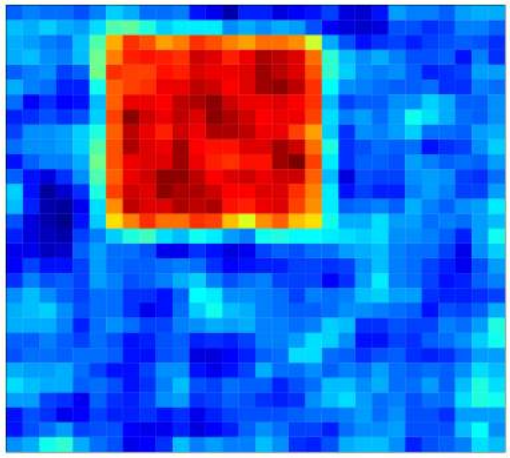

Figure 1.3: Sample mean of the reconstructed signal, in the standard normal signal.

1.1.2 Forming of scale-dependent details

The next step in the multiresolution decomposition is the decomposition itself. Therefore, $\boldsymbol{S}_\lambda$ shall be, for now, an arbitrary smoother, which can be represented by an $n \times n$ matrix. It is supposed that $\lambda > 0$ is the smoothing parameter and the random signal $\boldsymbol{x}$ is interpreted as an n dimensional random vector. Moreover, let

$$0 = \lambda_1 < \lambda_2 < \ldots < \lambda_{L-1} < \lambda_L = \infty,$$

such that $\boldsymbol{S}_{\lambda_1}\boldsymbol{x} = \boldsymbol{x}$ is the identity mapping and $\boldsymbol{S}_{\lambda_L}\boldsymbol{x} = \boldsymbol{S}_\infty\boldsymbol{x}$ the mean of the random vector $\boldsymbol{x}$. With the help of those preliminaries, the random signal $\boldsymbol{x}$ can be written as differences of consecutive smooths,

$$\boldsymbol{x} = \sum_{i=1}^{L-1} \left(\boldsymbol{S}_{\lambda_i} - \boldsymbol{S}_{\lambda_{i+1}}\right) \boldsymbol{x} + \boldsymbol{S}_\infty \boldsymbol{x} = \sum_{i=1}^{L-1} \boldsymbol{z}_i + \boldsymbol{z}_L.$$

Then, the scale-dependent details are defined as

$$\boldsymbol{z}_i = \left(\boldsymbol{S}_{\lambda_i} - \boldsymbol{S}_{\lambda_{i+1}}\right) \boldsymbol{x}, \qquad \text{for } i = 1, \ldots, L-1, \tag{1.8}$$

$$\boldsymbol{z}_L = \boldsymbol{S}_\infty \boldsymbol{x}. \tag{1.9}$$

The smoother in use, is a roughness penalty smoother

$$\boldsymbol{S}_\lambda = (\mathbf{I} + \lambda \boldsymbol{Q})^{-1}, \tag{1.10}$$

which minimizes the penalized loss defined by $\boldsymbol{Q}$, which can be expressed as

$$\boldsymbol{S}_\lambda \boldsymbol{x} = \underset{\boldsymbol{u}}{\operatorname{argmin}} \{||\boldsymbol{x} - \boldsymbol{u}||^2 + \lambda \boldsymbol{u}^T \boldsymbol{Q} \boldsymbol{u}\}.$$

As the attentive reader already recognized in the definition (1.10) of the smoother, one can replace the precision $\boldsymbol{Q}$ by its eigendecompostion (1.7). Furthermore, one supposes that $\boldsymbol{Q}$ has a null space of dimension n_0, such that $1 \leq n_0 < n$, hence

$$\operatorname{rank}(\boldsymbol{Q}) = n - n_0 \quad \text{and then} \qquad 0 = \gamma_1 = \ldots = \gamma_{n_0} < \gamma_{n_0+1} \leq \ldots \leq \gamma_n.$$

The composition of the smoothing matrix $\boldsymbol{S}_\lambda$ is, as suggested above, depicted as

$$\boldsymbol{S}_\lambda \boldsymbol{x} = \sum_{j=1}^{n_0} \left(\boldsymbol{v}_j^T \boldsymbol{x}\right) \boldsymbol{v}_j + \sum_{i=n_0+1}^{n} (1+\lambda\gamma_j)^{-1} \left(\boldsymbol{v}_j^T \boldsymbol{x}\right) \boldsymbol{v}_i. \tag{1.11}$$

The $L-2$ first details $\boldsymbol{z}_i$ can be rewritten also as

$$\boldsymbol{z}_i = \left(\boldsymbol{S}_{\lambda_i} - \boldsymbol{S}_{\lambda_{i+1}}\right) \boldsymbol{x} = \sum_{j=n_0+1}^{n} \left[(1+\lambda_i\gamma_j)^{-1} - (1+\lambda_{i+1}\gamma_j)^{-1}\right] \left(\boldsymbol{v}_j^T \boldsymbol{x} \boldsymbol{v}_j\right). \tag{1.12}$$

The $L-1$th detail is

$$\boldsymbol{z}_{L-1} = \left(\boldsymbol{S}_{\lambda_{L-1}} - \boldsymbol{S}_{\lambda_L}\right) \boldsymbol{x} = \sum_{j=n_0+1}^{n} (1+\lambda_{L-1}\gamma_j)^{-1} \left(\boldsymbol{v}_j^T \boldsymbol{x}\right) \boldsymbol{v}_j \tag{1.13}$$

and the Lth detail $\boldsymbol{z}_L$ writes it as

$$\boldsymbol{z}_L = \boldsymbol{S}_{\lambda_L} \boldsymbol{x} = \sum_{j=1}^{n_0} \left(\boldsymbol{v}_j^T \boldsymbol{x}\right) \boldsymbol{v}_j. \tag{1.14}$$

The smoothing effect of $\boldsymbol{S}_\lambda$ on $\boldsymbol{x}$ can be interpreted, such that, if λ is increasing, it suppresses most of the projections of $\boldsymbol{x}$ onto the $\boldsymbol{v}_j$'s with the respective largest eigenvalues.

Selection of smooths The question, which in this context remains, is how should one choose the smoothing levels, such that the decomposition will capture the scale-dependent features of $\boldsymbol{x}$ as precise as possible? Avoiding experimenting brute force with different choices of the smoothing levels λ_i's, whereupon $i \in \mathbb{N}^+$ is the number of choices. The answer to this question are so-called tapering functions. To construct those, one rewrites first the equations (1.12), (1.13) and (1.14) as

$$\boldsymbol{z}_i = \sum_{j=1}^{n} \alpha_j^{(i)} \left(\boldsymbol{v}_j^T \boldsymbol{x}\right) \boldsymbol{v}_j,$$

where for $i = 1, \ldots, L-2$

$$\alpha_j^{(i)} = \begin{cases} 0, & 1 \leq j \leq n_0 \\ (1+\lambda_i\gamma_j)^{-1} - (1+\lambda_{i+1}\gamma_j)^{-1}, & n_0 < j \leq n \end{cases}, \tag{1.15}$$

for $i = L-1$

$$\alpha_j^{(L-1)} = \begin{cases} 0, & 1 \leq j \leq n_0 \\ (1+\lambda_{L-1}\gamma_j)^{-1}, & n_0 < j \leq n \end{cases} \tag{1.16}$$

and for $i = L$

$$\alpha_j^{(i)} = \begin{cases} 1, & 1 \leq j \leq n_0 \\ 0, & n_0 < j \leq n \end{cases}. \tag{1.17}$$

The L sequences $\alpha_i = \left[\alpha_j^{(i)}\right]_{j=1}^n$ are henceforth referred to as tapering functions. The set $0 = \lambda_1 < \lambda_2 < \ldots < \lambda_{L-1} < \lambda_L = \infty$ is now chosen, such that the supports of the tapering functions are approximately disjoint, i.e. consisting of roughly non-overlapping segments of the integers $1, \ldots, n$. This will produce details that are in an extended sense orthogonal and therefore it can be assumed that the corresponding features consists of non similar characteristics.

One notes, that for the tapering functions, defined like in equations (1.15), (1.16) and (1.17), the eigenvalues of the precision matrix $\boldsymbol{Q}$ are absolutely essential.

Signal-dependent tapering functions The smoothing level selection method with tapering functions depends, so far, only on the eigenvalues of the matrix $\boldsymbol{Q}$ and therefore on the particular smoother $\boldsymbol{S}_\lambda$ in use, as well as the dimensions of the analyzed signal. It seems reasonable that better results could be obtained, by taking into account also the structure of the underlying signal $\boldsymbol{x}$. This can be accomplished by considering signal-dependent tapering functions, to derive them one starts with

$$\left[\alpha_j^{(i)}\left(\boldsymbol{v}_j\boldsymbol{x}\right)\right]_{j=1}^n.$$

However, as the underlying signal is unknown, it is replaced by the posterior mean $\mathbb{E}(\boldsymbol{x}|\boldsymbol{y})$ and therefore the signal-dependent tapering functions are defined as

$$\tilde{\alpha}_i = \left[\tilde{\alpha}_j^{(i)}\right]_{j=1}^n, \qquad \tilde{\alpha}_j^{(i)} = \alpha_j^{(i)}\left(\boldsymbol{v}_j^T\mathbb{E}(\boldsymbol{x}|\boldsymbol{y})\right), \qquad i = 1, \ldots, L.$$

The tapering functions of the standard normal example are shown in the Figures 1.4 and 1.5. One notes, that the x-axis of the tapering functions are on a logarithmic scale. Where Figure 1.4 shows the tapering function, which is only dependent on the dimension of the precision $\boldsymbol{Q}$ and the corresponding eigenvalues. On the other hand shows Figure 1.5 the signal-dependent tapering functions for the same set of smoothing levels. As expected, do the signal-dependent tapering functions involve more scatter and are more irregular than the other ones, but show also the desired disjoint supports.

Optimization The optimization procedure is mentioned here for completeness. Since the following work does not consider changes or improvements for this step, it is omitted in the following chapters. The signal-dependent tapering functions are irregular and a visual selection of fitting smoothing levels is not appropriate. Therefore an analytically selection is needed. The proposed method uses optimization of a suitable objective function, with respect to the λ_i's to achieve rough orthogonality of the tapering functions. To optimize a signal multiresolution decomposition consisting of just four terms corresponding to a smoothing parameter sequence $[0, \lambda_2, \lambda_3, \infty]$, minimize the objective function

$$\mathrm{G}(\lambda_2, \lambda_3) = \sum_{\substack{i,j=1,2,3 \\ \text{s.t. } |i<j|}} \frac{|\tilde{\boldsymbol{\alpha}}_i^T\tilde{\boldsymbol{\alpha}}_j|}{||\tilde{\boldsymbol{\alpha}}_i||||\tilde{\boldsymbol{\alpha}}_j||},$$

with respect to λ_2 and λ_3. It is possible to carry out the optimization, by simply evaluating $\mathrm{G}(\lambda_2, \lambda_3)$ on a grid.

The expected, scale dependent details of the data above, are shown in Figure 1.6. The smoothing levels are chosen with the aforementioned optimization procedure and the corresponding details are briefly discussed. Detail $\boldsymbol{z}_1$ is the most narrow one, which can be compared

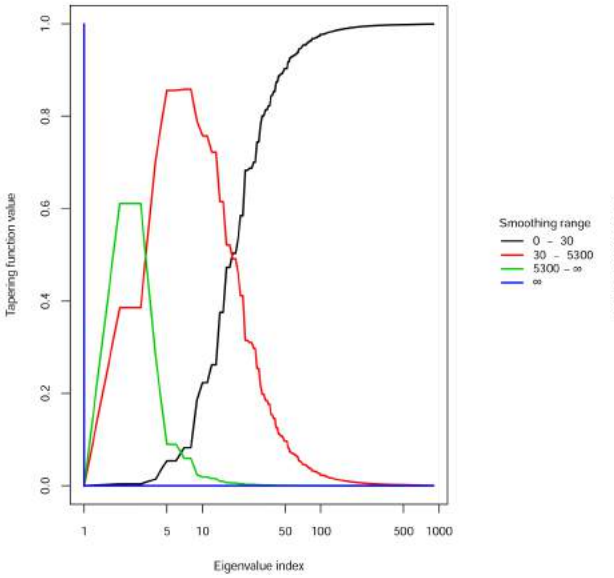

Figure 1.4: Signal-independent tapering functions of the standard normal signal.

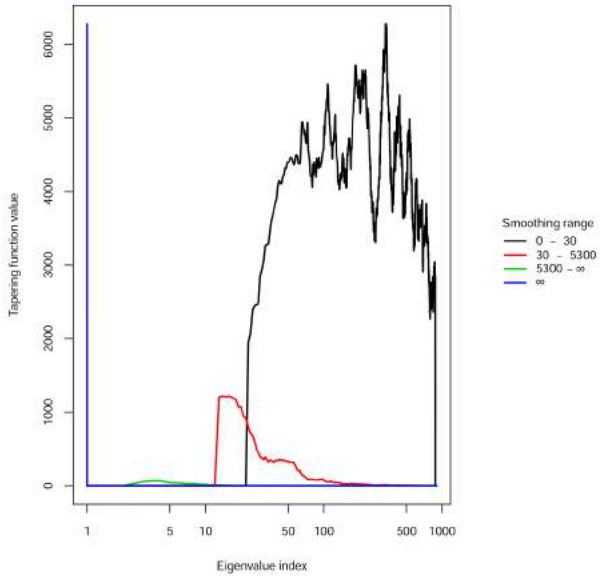

Figure 1.5: Signal-dependent tapering functions of the standard normal signal.

to a very close zoom level, accordingly very specific and small features of the data are recognizable. Otherwise in the details $\boldsymbol{z}_2$ and $\boldsymbol{z}_3$, the artificial feature in the upper left corner is salient and outshines the small features. The mean is illustrated in detail $\boldsymbol{z}_4$, respectively corresponds to $\boldsymbol{z}_\infty$, with only one single color.

1.1.3 Posterior credibility analysis

During the Bayesian signal reconstruction step a specific amount of posterior samples was generated, those samples are not only used to calculate the expected scale dependent details, but also to infer the credible features in the detail components. Inference on credible features is proposed in three different ways by Holmström *et al.* (2011).

Pointwise maps In the pointwise maps (PW) credibility analysis, a detail $\boldsymbol{z}_i$ is considered as a vectorization of an array $[\boldsymbol{z}_s]_{s\in I}$. For every location s, $\boldsymbol{z}_s$ is divided in three disjoint subsets of I in which the components $\boldsymbol{z}_s$ differ jointly credibly from zero.

$$
\begin{aligned}
I^b &= \{s|\mathrm{P}(\boldsymbol{z}_s > 0|\text{data}) \geq \alpha\},\\
I^r &= \{s|\mathrm{P}(\boldsymbol{z}_s < 0|\text{data}) \geq \alpha\},\\
I^g &= I \setminus \left(I^b \cup I^r\right).
\end{aligned}
$$

The common choice for the credibility level α is 95%. In pointwise maps, the subset allocation is done for each location independently over all samples. A point is colored blue if $s \in I^b$, red if $s \in I^r$ or gray if $s \in I^g$. Since pointwise maps treat every location independently, those can exhibit only very small islands of credibility. Simultaneous approaches, on the other hand,

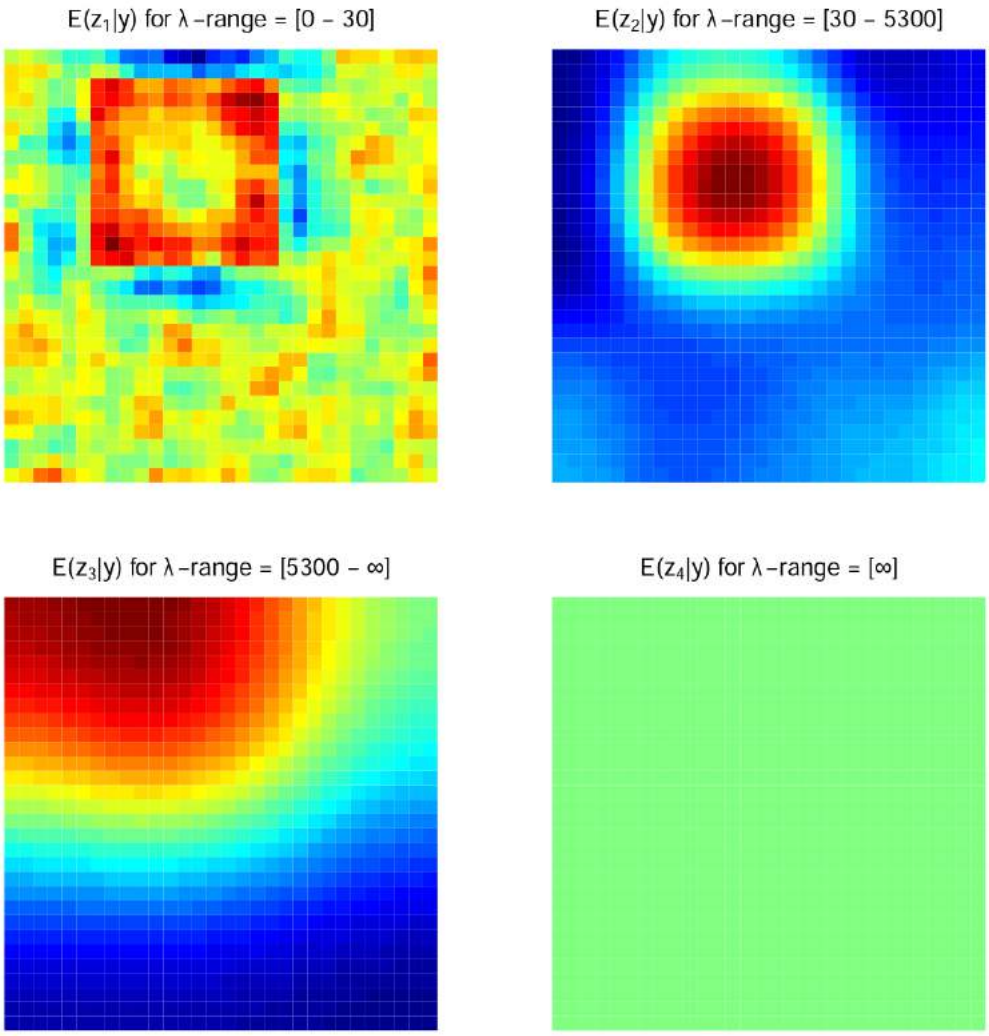

Figure 1.6: Scale dependent details of the standard normal signal, summarized by their posterior means.

assign credibility more conservatively to locations, but maximize the connectedness of credible locations.

Highest pointwise probability maps In simultaneous maps, the index subsets J^b, J^r and $J^g = I \setminus (J^b \cup J^r)$ are used. For highest pointwise probability maps (HPW), the division into subsets is according to the

$$\mathrm{P}(\boldsymbol{z}_s > 0 \text{ for } s \in J^b \text{ and } \boldsymbol{z}_s < 0 \text{ for } s \in J^r | y) \geq \alpha.$$

Not all of the locations s in sets I^b and I^r are also marked as credible in simultaneous maps, $J^b \subset I^b$, $J^r \subset I^r$ and $J^g \supset I^g$. This leads to more non-credible locations than with pointwise maps, but on the other hand, a stronger connectedness of credible locations can be observed

(Erästö and Holmström, 2005). If S is a set and $|S|$ the number of elements in this set, then $N = |I^b| + |I^r|$ is the number of credible locations using PW maps. The events $(\boldsymbol{z}_s > 0|y)|s \in I^b$ and $(\boldsymbol{z}_s < 0|y)|s \in I^r$ are denoted with E_s. For the permutations $s_1, \ldots, s_N$ of the locations $s \in I^b \cup I^r$ for which

$$\mathrm{P}(E_{s_1}) \geq \mathrm{P}(E_{s_2}) \geq \ldots \geq \mathrm{P}(E_{s_N}) \geq \alpha,$$

is valid and let $k = \max(l|\mathrm{P}(E_{s_1} \cup \ldots \cup E_{s_l}) \geq \alpha)$. If then, $l \in \{1, \ldots k\}$ s_l is marked blue or red, depending whether $s_l \in I^b$ or $s_l \in I^r$ and the rest is colored in gray.

Simultaneous credible intervals The third way to infer credible features proposed in the same paper, are simultaneous credible intervals. Therefore, let $\Delta > 0$ such that

$$\mathrm{P}\left(\max_{s \in l} \left| \frac{\boldsymbol{z}_s - \mathrm{E}(\boldsymbol{z}_s|\text{ data})}{\mathrm{Std}(\boldsymbol{z}_s|\text{ data})} \right| \leq \Delta|\text{ data}\right) = \alpha$$

and define

$$J^b = \{s|\,\mathrm{E}(\boldsymbol{z}_s|\text{ data}) - \Delta\mathrm{SD}(\boldsymbol{z}_s|\text{data}) > 0\},$$
$$J^r = \{s|\,\mathrm{E}(\boldsymbol{z}_s|\text{ data}) + \Delta\mathrm{SD}(\boldsymbol{z}_s|\text{data}) < 0\}.$$

In Figure 1.7 the pointwise probability maps for the standard normal example are separately shown for the different details. Blue points denote low draws from the standard normal example and red ones high draws, which are mostly dominated by the artificial feature in the upper left. The probability maps for the scale-dependent detail $\boldsymbol{z}_1$ includes features, which are more specific, for instance are the borders of the artifact in the upper left clearly credible. On larger scales almost all locations become credible.

1.2 Summary

The discussed method poses a exiting statistical method to decompose a spatial field into scale dependent features. Since the method is embedded in a Bayesian framework, it is possible to put probabilities on the different locations of the computed details. This is in particular useful and revolutionary, for making probabilistic statements about the appearance of specific features in spatial fields.

This method does rely on several assumptions, most importantly it is applicable for complete and regular lattice data only. To deal with this matter, imputation approaches are typically used, but current research covers only few information about its applicability. Imputation procedures replace missing values, with exact values, i.e in the signal reconstruction step the range of possible samples is close around the imputed value. If one thinks two steps ahead and considers also the credibility analysis, it would be of advantage if missing values are replaced, in the signal reconstruction step, such that a certain uncertainty is propagated for the respective locations. Especially, if the data to analyze consists of large areas with missing values and an imputation method is used, then is in addition the normal assumption violated. Those circumstances motivate for an enhancement of the outlined multiresolution decomposition method.

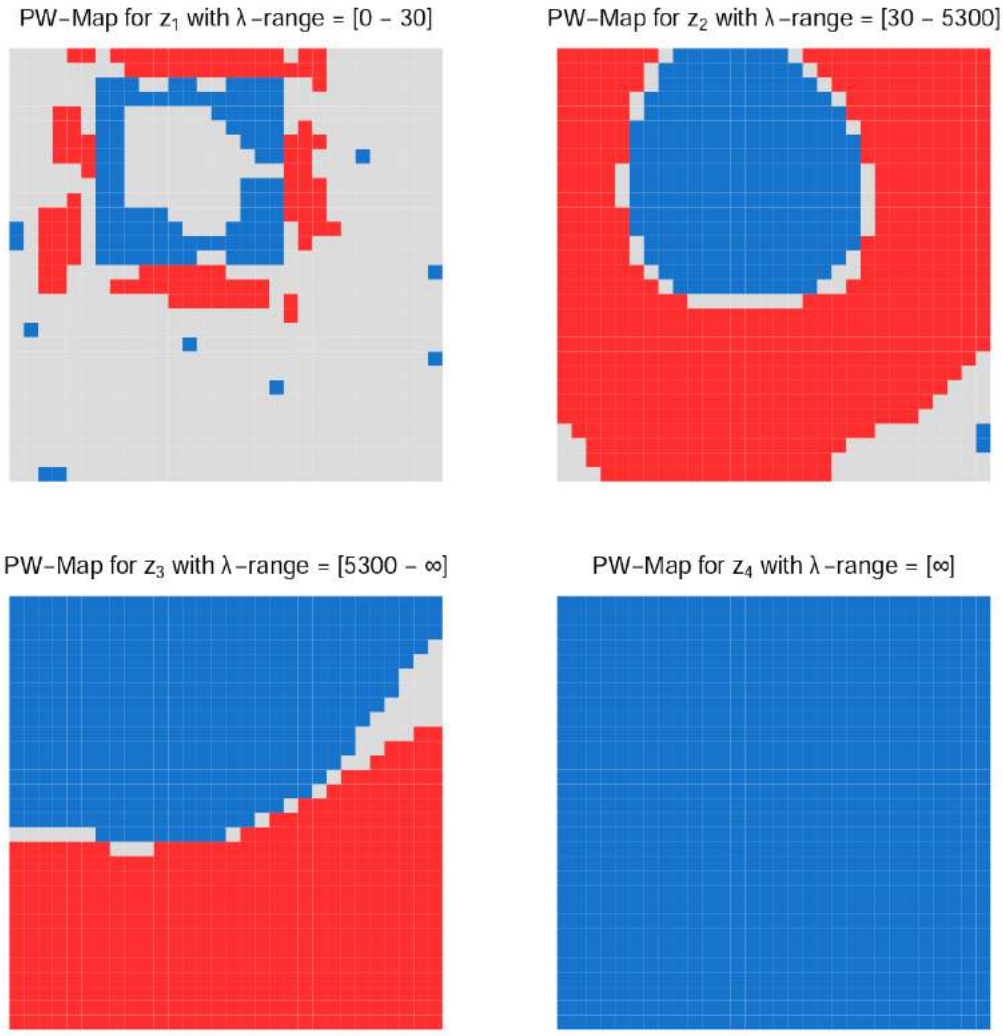

Figure 1.7: Pointwise probability maps of the standard normal signal.

Chapter 2

Data

In data collection, one often deals with missing measurements or with obvious errors thereof. Both cases lead to missing values in the data or in terms of multiresolution decomposition, to an incomplete signal. The introduced multiresolution decomposition method in Chapter 1 does rely on the assumption, that the observed signal is complete. Therefore, this chapter introduces second, the Lägeren mountain data set, which is then used to illustrate how the model in the introduced method does not give fully satisfying results. This motivates for new sampling models to deal with missing values, which are henceforth detailed derived in Chapter 3. Moreover, data is presented which is distributed on an irregular lattice. This data consists of standardized mortality ratios of oral cavity cancer counts in German districts.

2.1 Lägeren mountain

The example data originates from testing highly sensitive measuring devices in aircraft, to measure the diversity of vegetation. A team from the University Zurich, led by the head of the Remote Sensoring Laboratories Prof. Dr. Michael Schaepman, is using for instance a laser scanner (lidar) to determine the height, density and form of all the trees in a scanned area. The purpose of this measurements, is developing highly precise methods for remote sensing data, to systematically monitor changes in biodiversity. For more details about those projects and lidar, see Schneider *et al.* (2014) and Morsdorf *et al.* (2009).

The data in use was obtained from the Lägeren mountain, which is situated approximately 15 km northwest of Zurich, Switzerland. It stretches from Baden to Dielsdorf and its south slope marks the north boundary of the Swiss Plateau, which is bordered by the Jura and the alps. The natural vegetation is a productive and managed beech forest. According to Eugster *et al.* (2007), the forest stand has a relatively high diversity concerning species, age, and diameter distribution. In the eastern part, the common beech (Fagus sylvatica L.) and norway spruce (Picea abies L.) are prevalent. Where in the western part, broad-leaved trees such as european ash (Fraxinus excelsior L.), sycamore maple (Acer pseudoplatanus L.) and common beech are dominating. The size of the covered area is about 1.5×3 kilometers. The specific measurement for the further use, are "Canopy Heights" of trees in meters (m), see Figure 2.1 for an illustration.

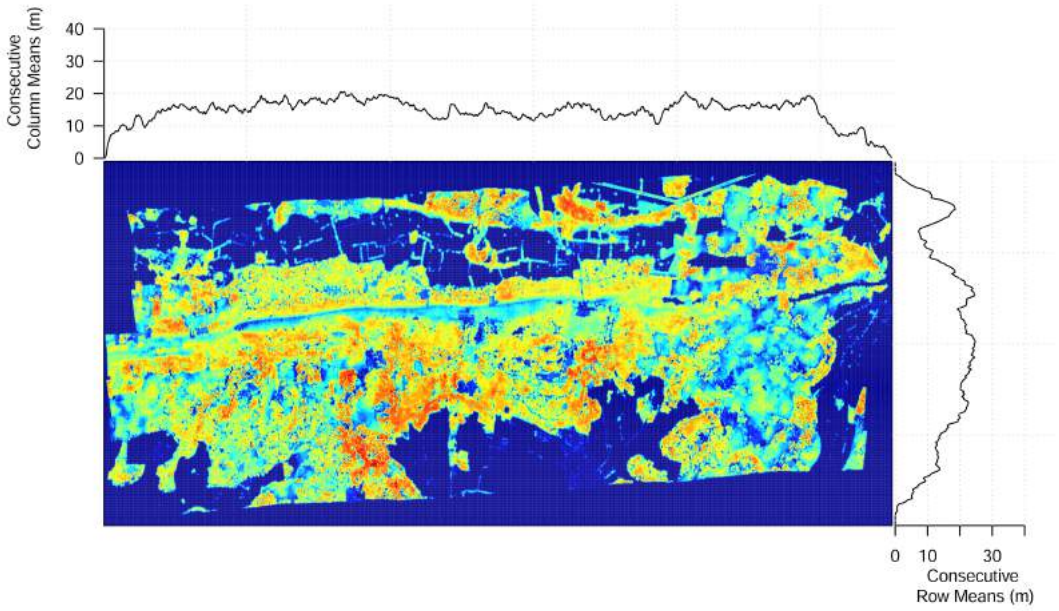

Figure 2.1: "Canopy Heights" measurements in meters, across the observed area of the Lägeren mountain. On the axis are the respective consecutive marginal means displayed.

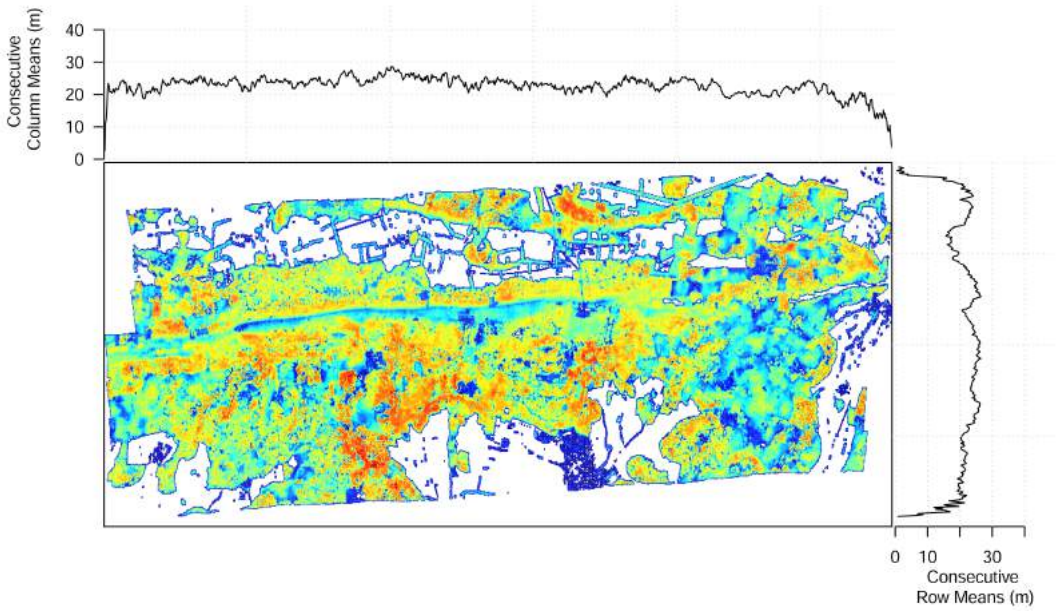

Figure 2.2: "Canopy Heights" measurements in meters, across the observed area of the Lägeren mountain. The white areas represent measurements treated as missing values.

Figure 2.3: Google maps satellite image of the Lägeren mountain.

2.1.1 Data distribution

The data is observed on a regular lattice, consisting of 400 × 1'100 points. Figure 2.1 shows canopy heights measurements between 0 and 50 meters. Red locations signify the highest values and blue ones the smallest. Eminent features, which are already recognizable through the raw data, is the mountain ridge, located in the middle of the forest. The ridge contains only few trees and is therefore depicted as a blue line. On the south slope are pattern of the highest trees visible, and accordingly colored in red. Also other smaller areas of high measurements as well as of low ones are salient.

Of the corresponding locations are 160'820 equal to zero and are interpreted as missing values. Figure 2.2 visualizes those, with respective white areas. One notices in this context, the differences of the corresponding row and column means in Figure 2.1 and 2.2, which become more uniform in the latter case.

Moreover, one can recognize relatively straight borders in the upper left as in the lower right area, between missing and non missing data. This implies, that in the areas above and respectively below, the measurements are truly missing, since there no measurements have been taken. This can be easily verified by comparing Figure 2.3, which shows a Google Maps satellite image of the Lägeren mountain and its environment, with the former figures. In this sense there are two types of missing values, locations with measurements close or equal to zero and locations, for which no measurements were taken. Since there is no clear distinction from the data possible, both cases are treated equally. Nevertheless, to assure that the data is normally distributed, the data is transformed with the third radical.

2.2 Oral cavity cancer

A second considered data,is distributed on a irregular lattice. Oral cavity cancer counts y_i were collected for each of the 544 districts of Germany over a five year period, from 1986-1990 (Knorr-Held and Raßer, 2000). To explore the spatial distribution of the relative risk, the expected number of cases e_i was derived using demographical data, that allows to display the standardized mortality ratios (SMRs) y_i/e_i, see also Figure 2.4. Those SMRs vary between between 0.15 and 2.40 and moreover, the data is assumed to be normally distributed, as Figure 2.6 shows. One recognizes a pattern of high SMRs in the south-west of Germany and an area of smaller values in the north-east. Otherwise, there are individual districts salient, with relative high values compared to their neighbor districts.

To model the neighboring structure of the German districts, one considers for every district its neighbors, which ultimately border on the respective district, i.e. its nearest neighbors. The adjacency matrix (Biggs, 1993) for the districts in Figure 2.5 determines for every district its nearest neighbors. This matrix and the data as well, are available in the `spam` package.

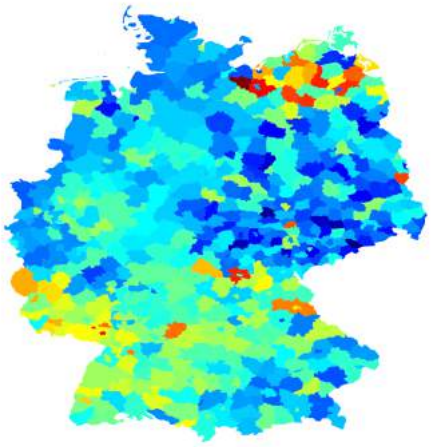

Figure 2.4: Standardized oral cavity cancer mortality ratios across Germany.

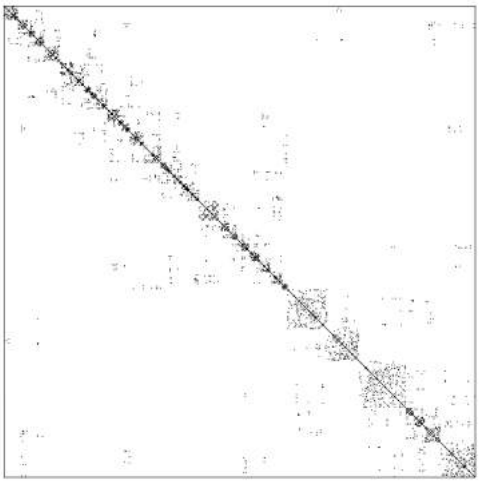

Figure 2.5: Adjacency matrix of all 544 districts in Germany, according to their neighboring relation.

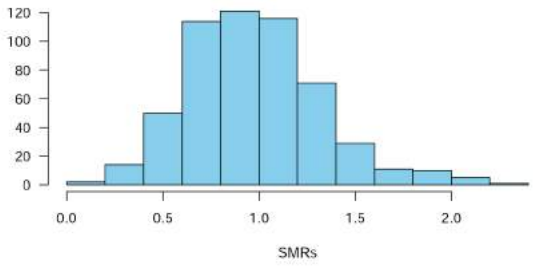

Figure 2.6: Histogram of standardized oral cavity cancer mortality ratios across Germany.

Chapter 3

Methods

In this chapter, first the scale space multiresolution analysis by Holmström *et al.* (2011) is put in a more general framework, to loose the dependency of the model specific precision matrix. Of main interest are then, two approaches to improve the aforementioned method for missing values, which are derived in this chapter. Due to this generalization, the fast eigendecompostion procedure through the discrete cosine transformation is lost. Hence, in possible steps, sparse structure based arithmetic is used to assure that large dimensional problems can be dealt with. For the implementation in the statistical Software `R`, the sparse matrices are of compressed sparse row format (Buluç *et al.*, 2009), which is available through the `spam` package (Furrer and Sain, 2010). Since tapering functions rely solely on the eigenvalues of the precision matrix, a method to approximate eigenvalues for precision matrices in use, is derived as well.

3.1 Multiresolution decomposition of an incomplete signal

3.1.1 Signal reconstruction

The first step to lose the requirement of a complete signal, is to make the sampling model more flexible, and independent of the specific precision matrix from equation (1.3). Therefore again a normal response model is chosen, but with a Gamma prior for the precision $\boldsymbol{\kappa}$ instead of a Scaled-Inv-χ^2 for σ^2. Those changes in the model lead to full conditional distributions of closed form, i.e. one can use a Gibbs sampling approach (Casella and George, 1992) to reconstruct the underlying signal. Since the Bayesian sample does not originate from a closed distribution anymore, but from a sampling procedure, one has to consider the diagnostics of it to assure the convergence of the sampling process.

Assumptions The observed signal $\boldsymbol{y}$ is again understood as composition of the true underlying signal $\boldsymbol{x}$ and an error $\boldsymbol{\varepsilon}$, as in equation (1.1). The error is assumed to be normal, with zero mean and constant unknown variance, i.e. $\varepsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$.

The likelihood function If one sets σ^2 to $1/\kappa_y$, then the likelihood function is proportional to

$$\pi(\boldsymbol{y}|\boldsymbol{x}, \kappa_y) \propto \kappa_y^{n/2} \times \exp\left(-\frac{\kappa_y}{2}\left(\boldsymbol{y}^\mathsf{T} - \boldsymbol{x}^\mathsf{T}\right)(\boldsymbol{y} - \boldsymbol{x})\right).$$

The process It is assumed, that $x_r - x_s$ for all $r \sim s$, are independent normals with zero mean, precision κ_x and spatial weight matrix $\boldsymbol{Q}_x$, i.e. an intrinsic random Walk model of order one, then

$$\pi(\boldsymbol{x}|\kappa_x) \propto \kappa_x^{(n-2)/2} \times \exp\left(-\frac{\kappa_x}{2}\boldsymbol{x}^\mathsf{T}\boldsymbol{Q}_x\boldsymbol{x}\right).$$

The priors For the normal response model, with known mean and unknown precision, the conjugate prior distribution is an independent Gamma prior distribution (Held and Sabanés Bové, 2013), which is used here,

$$\begin{aligned}\pi(\kappa_y) &\propto \kappa_y^{\alpha_y - 1} \times \exp\left(-\kappa_y\beta_y\right),\\ \pi(\kappa_x) &\propto \kappa_x^{\alpha_x - 1} \times \exp\left(-\kappa_x\beta_x\right).\end{aligned}$$

Whereupon, the according shape and rate parameters $\alpha_x, \alpha_y, \beta_x, \beta_y$ are strictly positive.

Joint density Let $\boldsymbol{\kappa}$ denote the vector of the precisions κ_y and κ_x. Then the corresponding joint density is

$$\pi\left(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\kappa}\right) = \pi\left(\boldsymbol{y}|\boldsymbol{x}, \kappa_y\right)\pi\left(\boldsymbol{x}|\kappa_x\right)\pi(\kappa_x)\pi(\kappa_y),$$

an intrinsic GMRF with precision matrix $\kappa_x\boldsymbol{Q}_x$ (see Appendix A.1). Which is proportional to

$$\begin{aligned}\kappa_x^{\alpha_x + (n-2)/2 - 1}\kappa_y^{\alpha_y + n/2 - 1} &\times \exp\left(-\kappa_x\beta_x - \kappa_y\beta_y\right)\\ &\times \exp\left(-\frac{\kappa_y}{2}\left(\boldsymbol{y}^\mathsf{T} - \boldsymbol{x}^\mathsf{T}\right)\left(\boldsymbol{y} - \boldsymbol{x}\right)\right)\\ &\times \exp\left(-\frac{\kappa_x}{2}\boldsymbol{x}^\mathsf{T}\boldsymbol{Q}_x\boldsymbol{x}\right).\end{aligned}$$

Full conditional distributions The Gibbs sampler is based on the following full conditional distributions. Thereby $\mathcal{N}_\mathcal{C}$ denotes the canonical parametrization of a normal distribution (see Appendix A.2).

$$\begin{aligned}
\pi(\boldsymbol{x}|\boldsymbol{\kappa},\boldsymbol{y}) &\propto \exp\left(-\frac{\kappa_y}{2}\left(\boldsymbol{y}^\mathsf{T}-\boldsymbol{x}^\mathsf{T}\right)(\boldsymbol{y}-\boldsymbol{x})-\frac{\kappa_x}{2}\boldsymbol{x}^\mathsf{T}\boldsymbol{Q}_x\boldsymbol{x}\right)\\
&\propto \exp\left(-\frac{1}{2}\boldsymbol{x}^\mathsf{T}\left(\kappa_x\boldsymbol{Q}_x+\kappa_y\mathbf{I}_n\right)\boldsymbol{x}+\kappa_y\boldsymbol{y}^\mathsf{T}\boldsymbol{x}\right),\\
\boldsymbol{x}|\boldsymbol{\kappa},\boldsymbol{y} &\sim \mathcal{N}_\mathcal{C}\left(\kappa_y\boldsymbol{y},\kappa_x\boldsymbol{Q}_x+\kappa_y\mathbf{I}_n\right),
\end{aligned}$$

$$\begin{aligned}
\pi(\kappa_x|\boldsymbol{x},\boldsymbol{y}) &\propto \kappa_x^{\alpha_x-1}\exp\left(-\kappa_x\beta_x\right)\kappa_x^{(n-2)/2}\exp\left(-\frac{\kappa_x}{2}\boldsymbol{x}^\mathsf{T}\boldsymbol{Q}_x\boldsymbol{x}\right)\\
&\propto \kappa_x^{\alpha_x+(n-2)/2-1}\exp\left(-\kappa_x\left(\beta_x+\frac{1}{2}\boldsymbol{x}^\mathsf{T}\boldsymbol{Q}_x\boldsymbol{x}\right)\right),\\
\kappa_x|\boldsymbol{x},\boldsymbol{y} &\sim \text{Gamma}\left(\alpha_x+\frac{n-2}{2},\beta_x+\frac{1}{2}\boldsymbol{x}^\mathsf{T}\boldsymbol{Q}_x\boldsymbol{x}\right),
\end{aligned}$$

$$\begin{aligned}
\pi(\kappa_y|\boldsymbol{x},\boldsymbol{y}) &\propto \kappa_y^{\alpha_y-1}\exp\left(-\kappa_y\beta_y\right)\kappa_y^{n/2}\exp\left(-\frac{\kappa_y}{2}\left(\boldsymbol{y}^\mathsf{T}-\boldsymbol{x}^\mathsf{T}\right)(\boldsymbol{y}-\boldsymbol{x})\right)\\
&\propto \kappa_y^{\alpha_y+n/2-1}\exp\left(-\kappa_y\left(\sum_i(y_i-x_i)^2+\beta_y\right)\right),\\
\kappa_y|\boldsymbol{x},\boldsymbol{y} &\sim \text{Gamma}\left(\alpha_y+\frac{n}{2},\beta_y+\sum_i(y_i-x_i)^2\right).
\end{aligned}$$

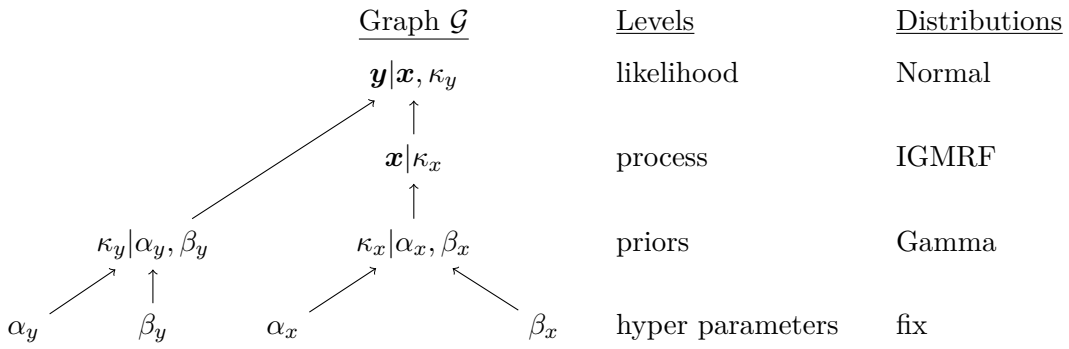

Figure 3.1: Summary of the extended Bayesian hierarchical model, for arbitrary precision matrices.

Scale-dependent details Through its more general from, the fast eigendecomposition, by the discrete cosine transformation of the precision matrix $\boldsymbol{Q}$ can not be applied anymore. Hence, the scale dependent details, as defined in equations (1.8) and (1.9), can not be computed efficiently for large dimensional problems though their eigendecomposition expression in equation (1.11). But, while still using the roughness penalty smoother from equation (1.10), one can use the Cholesky factorization to solve $(\mathbf{I}+\lambda\boldsymbol{Q})^{-1}\boldsymbol{x}$, instead of calculating the inverse directly. Therefore one computes the details $\boldsymbol{z}_i$ for $i=1,\ldots,L-1$ from

$$\boldsymbol{z}_i=(\mathbf{I}+\lambda_i\boldsymbol{Q})^{-1}\boldsymbol{x}-(\mathbf{I}+\lambda_{i+1}\boldsymbol{Q})^{-1}\boldsymbol{x}$$

and detail L from

$$\boldsymbol{z}_L = (\mathbf{I} + \lambda_L \boldsymbol{Q})^{-1} \boldsymbol{x}.$$

Tapering functions The tapering functions defined in equations (1.15), (1.16) and (1.17) rely on the eigenvalues of the precision matrix $\boldsymbol{Q}$ and the smoothing levels λ_i only, therefore an approximation approach is used to calculate efficiently its eigenvalues. This approximation is explained and derived in Section 3.2 below.

Posteriori credibility analysis The posteriori credibility analysis remains as outlined for the the original model in Chapter 1.

3.1.2 Model to sample missing values

Under the same assumptions as in the previous section and under the circumstances that the true underlying signal $\boldsymbol{x}$ contains k missing values. One denotes $\boldsymbol{H}$ as a matrix operator from $\mathbb{R}^n \to \mathbb{R}^m$, such that $m = n - k$, where n is the number of locations on the lattice. Hence the observed signal $\boldsymbol{y} \in \mathbb{R}^m$ Then $\boldsymbol{H}$ shall map the m non-missing values of the field, identically to the corresponding entries of $\boldsymbol{y}$. With the help of $\boldsymbol{H}$ one can transform the likelihood function of the model to $\boldsymbol{y}|\boldsymbol{H}\boldsymbol{x}, \boldsymbol{\kappa} \sim \mathcal{N}\left(\boldsymbol{H}\boldsymbol{x}, \frac{1}{\kappa_y}\mathbf{I}_m\right)$. In this sense only the data, which is not missing has an influence on the posterior sample. The missing values are sampled, based on the neighbor values and the prior influence. According to this transformation, the hierarchical model depicted in Figure 3.1, is modified to

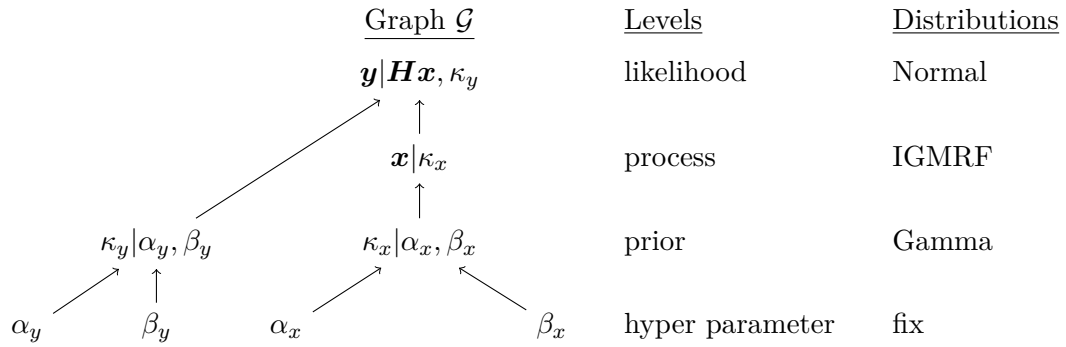

Figure 3.2: Summary of the modified Bayesian hierarchical model, to sample missing values.

Full conditional distributions The full conditional distributions, used for the modified Gibbs sampler become,

$$\begin{aligned}\pi(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\kappa}) &\propto \exp\left(-\frac{\kappa_y}{2}\left(\boldsymbol{y}^\top-\boldsymbol{x}^\top\boldsymbol{H}^\top\right)(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x})-\frac{\kappa_x}{2}\boldsymbol{x}^\top\boldsymbol{Q}_x\boldsymbol{x}\right)\\ &\propto \exp\left(\kappa_y\boldsymbol{y}^\top\boldsymbol{H}\boldsymbol{x}-\frac{1}{2}\boldsymbol{x}^\top\left(\kappa_y+\kappa_x\boldsymbol{Q}_x\right)\boldsymbol{x}\right),\\ \boldsymbol{x}|\boldsymbol{y},\boldsymbol{\kappa} &\sim \mathcal{N}_{\mathcal{C}}\left(\kappa_y\boldsymbol{H}^\top\boldsymbol{y},\kappa_y+\kappa_x\boldsymbol{Q}_x\right),\end{aligned}$$

$$\kappa_x|\boldsymbol{x},\boldsymbol{y}\sim\text{Gamma}\left(\alpha_x+\frac{n-2}{2},\beta_x+\frac{1}{2}\boldsymbol{x}^\top\boldsymbol{Q}_x\boldsymbol{x}\right),$$

$$\begin{aligned}\pi(\kappa_y|\boldsymbol{x},\boldsymbol{y}) &\propto \kappa_y^{n/2+\alpha_y-1}\exp\left(-\frac{\kappa_y}{2}\left(\boldsymbol{y}^\top-\boldsymbol{x}^\top\boldsymbol{H}^\top\right)(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x})-\kappa_y\beta_y\right),\\ \kappa_y|\boldsymbol{y},\boldsymbol{x} &\sim \text{Gamma}\left(\alpha_y+\frac{n}{2},\beta_y+\frac{1}{2}\left(\boldsymbol{y}^\top-\boldsymbol{x}^\top\boldsymbol{H}^\top\right)(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x})\right).\end{aligned}$$

Illustration To illustrate this concept, one considers a random draw from a normal distribution of size 100, with mean equal to ten and a standard deviation of five. This random signal can be interpreted as a 10×10 spatial field, with some randomly picked locations assumed to be missing values. Figure 3.3 shows the described scenario and Figure 3.4 the mean of 1'000 draws according to the modified Gibbs sampler. It is easy to see that on average the values in the locations, with former missing values, are plausible.

It is intuitive, that the locations with missing values should have higher uncertainties, because they will be propagated throughout the detail decomposition step to the credibility analysis. As described above, transforming the likelihood function such that the missing values rely on the priors and the neighbor values only, this can be achieved. For a better understanding, the sample draws of the random signal is plotted in one dimension, instead of a 10×10 field. Figure 3.5 shows this matter, in black is the observed signal, which is incomplete and in red are the 1'000 draws of the modified Gibbs sampler. One can clearly see, that in the locations where the observed signal is missing, the scatter is far wider around the true value than in the other locations. Nevertheless the draws for the missing values are not totally random, but still signal dependent through the influences by the neighbors and the prior distributions. Missing locations at the border are exposed to less values and therefore a skewed and larger uncertainty can be observed.

3.1.3 Removing missing values

In Subsection 2.1.1, it was argued, that the Lägeren mountain data, contains two types of missing values. On one hand truly missing values, that means, the reason lies in the fact that for the respective locations, no or a obviously false measurement is available. Otherwise, there are also locations with values equal to zero, since there is topological or context based reason. For example if one is interested in the composition of the first earth layer in a forest and in a part of the observed area is a lake. Then one knows for sure that in the area of the lake, there is no measurement available and setting the the value for the respective locations to zero or

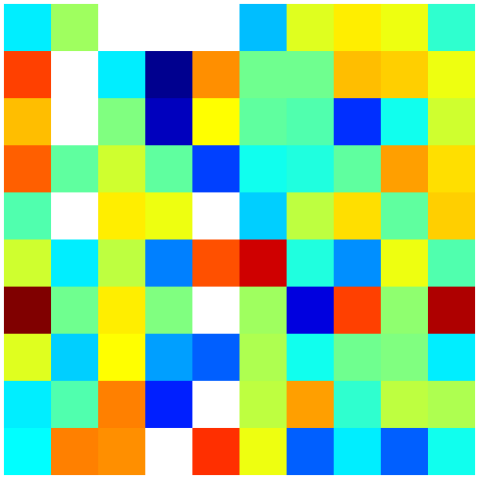

Figure 3.3: Normal distributed random signal in a 10×10 field, with missing values at ten random locations.

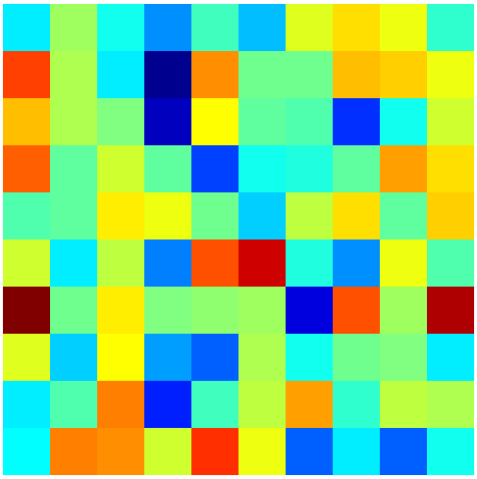

Figure 3.4: Sample mean over 1'000 draws, with resampled missing values of the same field.

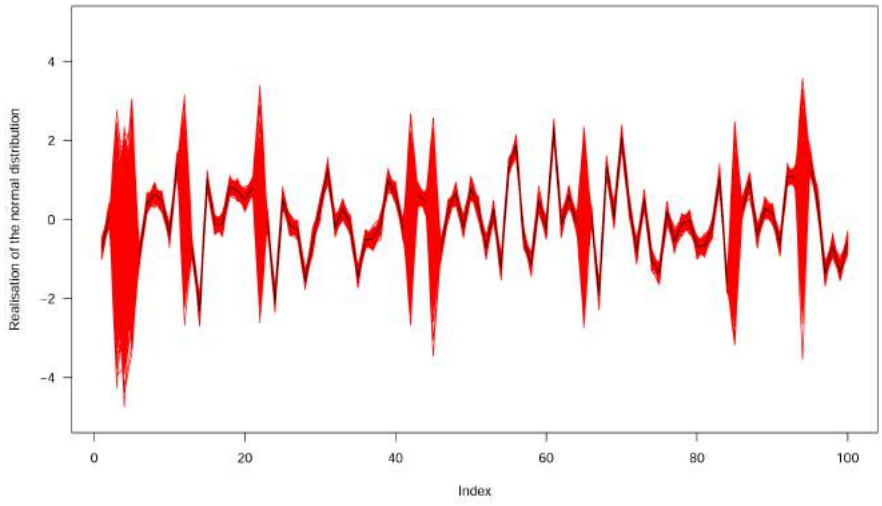

Figure 3.5: Normal signal displayed in one dimension. In black is the observed incomplete signal and in red 1'000 draws of the Gibbs sampler.

another value would be wrong. Hence, the option which remains, is to interpret the value as missing or not prevalent. Therefore a second approach to deal with this kind of missing values, is introduced.

Since those values are known to be not prevalent, one can account for this fact, by excluding all respective locations. Furthermore the entries of the precision matrix need to be modified as well, in particular one needs to remove the rows and the columns of the locations in question. The diagonal entries of the precision matrix depend on the neighboring row and column values and need therefore to be adjusted too. Otherwise, one needs not to make further adjustments and can work with the model from Figure 3.1 and the associated signal decomposition.

3.2 Eigenvalue approximation

In this section, a method to approximate eigenvalues for precision matrices of random walk models is derived. To control the accuracy of the approximated eigenvalues and vectors, the implicitly restarted Arnoldi algorithm is used.

This algorithm is available in the Fortran library ARPACK (Lehoucq *et al.*, 1998), which was implemented in the `R` package `spam`, during the course of this thesis. The ARPACK library is designed to compute a few eigenvalues and corresponding eigenvectors of a general $n \times n$ matrix. When a matrix is symmetric it reduces to a variant of the Lanczos process. These variants may be viewed as a synthesis of the Arnoldi/Lanczos process with the implicitly shifted QR technique that is suitable for large scale problems (Golub and Van Loan, 2013). The `R` implementation thereof, consists basically of an `R` wrapper function, which calls two different Fortran routines, dependent whether the input matrix is symmetric or not, see the `R` code in Appendix B.1. The Fortran routines are based on examples, provided by the ARPACK library and include all necessary functions and data structures to run successfully the algorithm. Appendix B.2 contains the subroutine `dn_eigen_f` for the non symmetric case, the subroutine `ds_eigen_f` is analog to the former one. A matrix, in `spam`'s compressed sparse row format, is handed over to the Fortran routine, where the entire eigenvalue and eigenvector computations are carried out. The results are then returned back to the `R` wrapper, which returns the nicely formatted eigenvalues and eigenvectors.

3.2.1 Circulant and Toeplitz matrices

For the eigenvalue approximation one uses the fact, that for a large enough dimensions a circulant and a Toeplitz matrix are approximately equivalent. Since, eigenvalues of circulant matrices can be calculated with a closed from expression, the eigenvalues of large dimensional matrices can be computed efficiently.

A circulant matrix $\boldsymbol{C}$ is a matrix of the form

$$\boldsymbol{C} = \begin{pmatrix} c_0 & c_1 & c_2 & & \cdots & c_{n-1} \\ c_{n-1} & c_0 & c_1 & c_2 & \cdots & c_{n-2} \\ & \ddots & \ddots & \ddots & \ddots & \\ & & & & \ddots & c_2 \\ c_2 & \cdots & & \ddots & \ddots & c_1 \\ c_1 & c_2 & \cdots & & c_{n-1} & c_0 \end{pmatrix}, \tag{3.1}$$

where each row is a cyclic shift of the row above. Properties of circulant matrices are well known and derived by Gray (2005). Those, which sketch the theoretical basis of the proposed eigenvalue approximation, are stated in the following.

Closed form for eigenvalues and eigenvectors of circulant matrices Every circulant matrix $\boldsymbol{C}$ has eigenvectors

$$y^{(m)} = \frac{1}{\sqrt{n}} \left(1,\ \exp\left(-2\pi i m/n\right),\ \ldots,\ \exp\left(-2\pi i m(n-1)/n\right)\right)^{\mathsf{T}}$$

and corresponding eigenvalues

$$\psi_m = \sum_{k=0}^{n-1} c_k \exp\left(-2\pi i m k/n\right),$$

with $m = 1, \ldots, n$ and $i = \sqrt{-1}$. In this sense, all circulant matrices bear eigenvectors of the same form.

This result grants the possibility, to calculate eigenvalues and eigenvectors of every circulant matrix. The next step is to link circulant and Toeplitz matrices, the latter will then be used for a connection to precision matrices of random walk models.

Banded Toeplitz matrix A banded $n \times n$ Toeplitz matrix $\boldsymbol{T}_n$, possessing a finite number of diagonals with nonzero entries and zeros everywhere else, is of the form

$$\boldsymbol{T_n} = \begin{pmatrix} t_0 & t_{-1} & \cdots & t_{-m} & & & & & & \\ \vdots & \ddots & \ddots & & \ddots & & & & & \\ t_m & & & & & & & & & \\ & \ddots & & \ddots & \ddots & & \ddots & & & \\ & & t_m & \cdots & t_0 & t_{-1} & \cdots & t_{-m} & & \\ & & & \ddots & & \ddots & \ddots & & \ddots & \\ & & & & & & & & & t_{-m} \\ & & & & & & & & \ddots & \vdots \\ & & & & & & \ddots & & \ddots & t_{-1} \\ & & & & & & & t_m & \cdots & t_0 \end{pmatrix}. \tag{3.2}$$

This banded Toeplitz matrix $\boldsymbol{T}_n$ can be extended to a circulant matrix $\boldsymbol{C}_n$, by adding respective values in the upper right and the lower left of the matrix

$$\boldsymbol{C_n} = \begin{pmatrix} t_0 & t_{-1} & \cdots & t_{-m} & & & & t_m & \cdots & t_1 \\ \vdots & \ddots & \ddots & & \ddots & & & & \ddots & \vdots \\ t_m & & & & & & & & & t_m \\ & \ddots & & \ddots & \ddots & & \ddots & & & \\ & & t_m & \cdots & t_0 & t_{-1} & \cdots & t_{-m} & & \\ & & & \ddots & & \ddots & \ddots & & \ddots & \\ & & & & & & & & & t_{-m} \\ & & & & & & & & \ddots & \vdots \\ t_{-m} & & & & & & \ddots & & \ddots & t_{-1} \\ \vdots & \ddots & & & & & & t_m & \cdots & t_0 \\ t_{-1} & \cdots & t_{-m} & & & & & & & \end{pmatrix}. \tag{3.3}$$

In order to justify an approximation of Toeplitz matrices by circulant ones, as n becomes large, one needs the principle of asymptotic equivalent matrices.

The strong norm Let $\boldsymbol{A}$ be a matrix with eigenvalues α_k and $\lambda_k \geq 0$ the eigenvalues of the Hermitian positive semi definite $\boldsymbol{A}^*\boldsymbol{A}$. The strong norm $||\boldsymbol{A}||$ is defined by

$$||\boldsymbol{A}|| = \max_{\boldsymbol{z}:\boldsymbol{z}^*\boldsymbol{z}=1} \{\boldsymbol{z}^*\boldsymbol{A}^*\boldsymbol{A}\boldsymbol{z}\}^{1/2}.$$

The weak norm (Hilbert-Schmidt norm) of an $n \times n$ matrix $\boldsymbol{A} = [a_{k,j}]$ is defined by

$$|A| = \left(\frac{1}{n} \sum_{k=0}^{n-1} \sum_{j=0}^{n-1} |a_{k,j}|^2 \right)^{1/2}.$$

Asymptotic equivalent matrices Two sequences $\{\boldsymbol{A}_n\}$ and $\{\boldsymbol{B}_n\}$ of matrices are said to be asymptotically equivalent if $\boldsymbol{A}_n$ and $\boldsymbol{B}_n$ are uniformly bounded in the strong norm,

$$||\boldsymbol{A}_n|| \leq M \text{ and } ||\boldsymbol{B}_n|| \leq M, \text{ such that } M < \infty, \quad n = 1, 2, \ldots$$

and the difference of $\boldsymbol{A}_n$ and $\boldsymbol{B}_n$ goes to zero in the weak norm as n becomes large,

$$\lim_{n \to \infty} |\boldsymbol{A}_n - \boldsymbol{B}_n| = 0.$$

Given, that the two sequences $\{\boldsymbol{A}_n\}$ and $\{\boldsymbol{B}_n\}$ are asymptotically equivalent, one can show that for the respective eigenvalues $\{\alpha_{n,m}\}$ and $\{\beta_{n,m}\}$

$$\lim_{n \to \infty} \frac{1}{n} \sum_{m=1}^{n} (\alpha_{n,m} - \beta_{n,m}) = 0$$

and if either limit exists individually, then

$$\lim_{n \to \infty} \frac{1}{n} \sum_{m=1}^{n} (\alpha_{n,m}) = \lim_{n \to \infty} \frac{1}{n} \sum_{m=1}^{n} (\beta_{n,m}).$$

In other words, from the asymptotic equivalence of two sequences of matrices, it follows directly, that if n becomes large enough, one can use the respective eigenvalues as an approximation.

In Conclusion Under the additional assumption, that the banded Toeplitz matrices are absolutely summable, i.e.

$$\sum_{k=-\infty}^{\infty} |t_k| < \infty$$

it can be shown, that the matrices $\boldsymbol{T_n}$ and $\boldsymbol{C}_n$ are asymptotically equivalent. In other words both are bounded in the strong norm and in the weak norm and hence

$$\lim_{n \to \infty} |\boldsymbol{T}_n - \boldsymbol{C}_n| = 0.$$

3.2.2 Random walk precision matrix

In the following it is argued for the eigenapproximation by circulant matrices of precision matrices, which are typically used for the introduced models. Therefore, the specific structure of those precision matrices $\boldsymbol{Q}$ are introduced. A more rigorous derivation of it, is given in Rue and Held (2005) and the `R` implementation thereof is available in the `spam` package.

One assumes $\mathcal{I}_{\boldsymbol{n}}$ is a regular lattice, such that $\boldsymbol{n} = n_1 n_2$ the corresponding lattice dimensions and (r, s) denotes a node in the ith row and the jth column. In the interior of the lattice, the nearest four sites of (r, s) are defined as its neighbors, i.e $(r+1, s)$, $(r-1, s)$, $(r, s+1)$ and $(r, s-1)$. Furthermore, in the interior of the lattice it holds, that $\pi(x_{r,s}|\boldsymbol{x}_{-r,s}, \kappa)$ is normal with mean

$$\frac{1}{4}(x_{r+1,s} + x_{r-1,s} + x_{r,s+1} + x_{r,s-1}) \text{ and the conditional precision } 4\kappa.$$

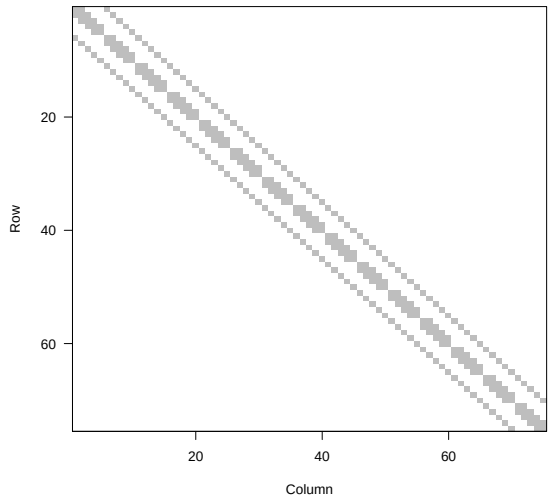

Figure 3.6: random walk of order one precision matrix $\boldsymbol{Q}$ of a 5×15 regular grid, according to equation (3.4).

The corresponding precision matrix is defined as

$$\boldsymbol{Q}_{r,s} = \kappa \begin{cases} n_r, & r = s, \\ -1, & r \sim s, \\ 0, & \text{otherwise}, \end{cases},$$

where n_r denotes the number of neighbors in region r.

Typically, an extended anisotropic model can be used, i.e. weighing the horizontal and vertical neighbors differently. More precisely, one supposes that the conditional mean is

$$\frac{1}{4}(\alpha'(x_{r+1,s} + x_{r-1,s}) + \alpha''(x_{r,s+1} + x_{r,s-1})).$$

Where α' and α'' are positive parameters, such that $\alpha' + \alpha'' = 2$. The conditional precision is unchanged and one can write the precision matrix as

$$\boldsymbol{Q} = \alpha' \boldsymbol{R}_{n_1} \otimes \mathbf{I}_{n_2} + \alpha'' \mathbf{I}_{n_1} \otimes \boldsymbol{R}_{n_2}. \tag{3.4}$$

Thereby is $\boldsymbol{R}_n$ the structure matrix and $\mathbf{I}_n$ the $n \times n$ dimensional identity matrix. The structure matrix of the order one random walk model of dimension $n \times n$ is

$$\boldsymbol{R}_n = \begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & \ddots & \ddots & \ddots \end{pmatrix}.$$

As Figure 3.6 shows, typical precision matrices, for the model in use, are not of an exact Toeplitz form. Therefore, it is necessary to investigate, how to choose the circular matrix for the approximation and for which lattice dimensions the approximation becomes accurate.

3.2.3 Precision matrix approximation by circular matrices

In the following the approximation of the eigenvalues of the precision matrix $\boldsymbol{Q}$ is shown. Since the principle is not proofed in theory, its applicability and accuracy is illustrated with a cohort study of matrices, with different respective dimensions.

From the previous definition, it becomes clear, that the precision matrices $\boldsymbol{Q}$ are not of exact Toeplitz form, like in equation (3.2). I.e. an element t_k, for $k = -m, \ldots, -1, 0, 1, \ldots, m$, is not equal to the same value for every row, this is due to the construction of $\boldsymbol{Q}$ through the Kronecker product. One can account for this structure the best, if one uses the mth row of the precision matrix and constructs a circulant matrix according to equation (3.3). With this choice one can speculate, that for a large enough dimension of the matrix, the number of different entries of the two matrices becomes small. Or in other words, the difference becomes bounded with respect to the weak matrix norm.

Random walk of order one precision matrices For the first matrix cohort for the eigenvalue approximation study, precision matrices according to the random walk model of order one are investigated. To start with non computationally intensive examples, the eigenvalues of precision matrices of dimensions $400 \times 400, \ldots,$ 4'900 $\times$ 4'900, 6'400 $\times$ 6'400 are approximated and compared to the corresponding true value. In Figures 3.7 and 3.8 are the differences between the approximated and the true eigenvalues displayed. This plot already demonstrates, that one can hope for useful eigenvalue approximations as the dimensions become larger. Moreover, this Figure shows for all the cases the same patter, the approximation is closest at the border and the middle of the corresponding eigen spectra. Also the comparison between Figure 3.9 and Figure 3.10 shows, that the corresponding tapering functions of the approximated eigenvalues are tolerably similar. One notes, that small eigenvalues include large smoothing levels and vice versa. The obvious differences are concerning the blue and green line, which correspond to the two largest smoothing ranges, except the mean. If one looks at the same plots with non log transformed x-axes, one could see the same properties, but less obvious. The motivation one can take from this example, is that an underlying grid for this precision is only of the size of 60×60 points.

The same procedure was applied to precision matrices of the same model but of higher dimensions. Since ordinary `R` functions cannot allocate matrices of those sizes and also are not able to do it in reasonable time, the aforementioned implicit restarted Arnoldi algorithm is used to control the accuracy of the approximated eigenvalues.

The underlying lattice for the precision matrix is not necessary of quadratic form, hence the precision matrices investigated do need to cover also those cases, since the structure of the the respective precision matrix is different than from a quadratic lattice. The cohort of matrices of this approximation study does also involve large matrices, to save computation time not all of the approximated eigenvalues are controlled with the iterated ones from the Arnoldi algorithm. In fact only $n^2/20$ are calculated for a $n \times m$ lattice. For those values the number of eigenvalues which differ more than a certain threshold, are counted and divided by the total number of controlled eigenvalues. This number is referred to, as the relative number of rejected eigenvalues with the threshold $d = 0.01$. One can argue for this threshold, with the resulting tapering

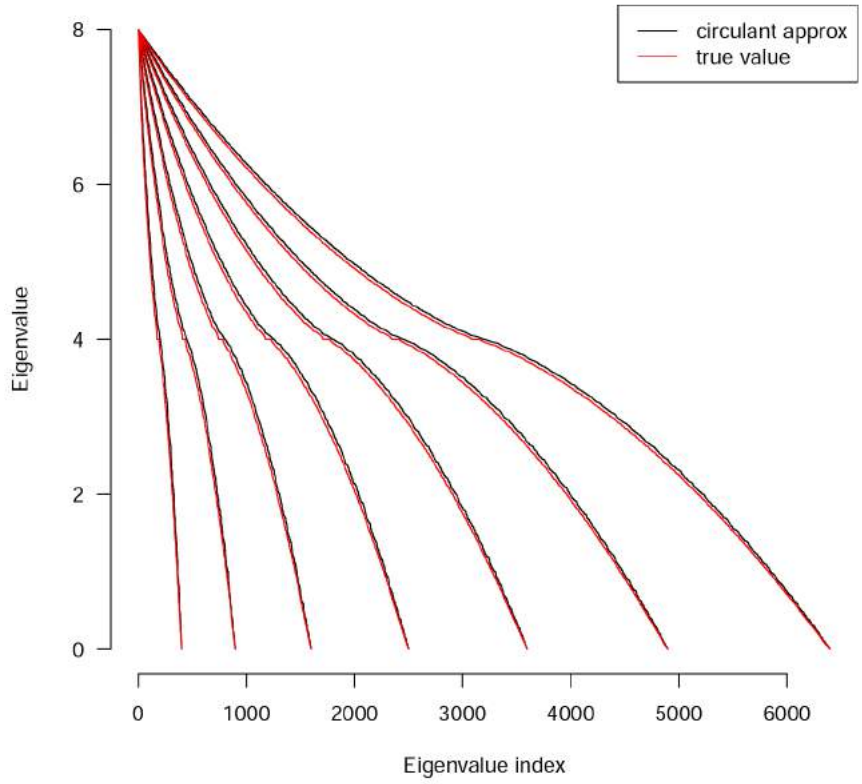

Figure 3.7: Relatively small precision matrices of a random walk model of order one, such that the dimensions increase quadratic. Every pair of black and red lines represent the eigenvalues of another precision matrix.

functions, for which such a value is small enough do derive correct smoothing levels.

The attentive reader notes, that for a lattice of dimension $n \times m$ a precision matrix of dimension $n^2 \times m^2$ is necessary. In the following the value for the underlying lattices of the data is used to differentiate between different precision matrices. Figure 3.11 shows for a cohort of matrices, regarding the mentioned relative number of rejected eigenvalues, which is displayed with bubbles of diameter sizes corresponding to its value. Due to the symmetry of precision matrices, one could exchange lattice rows and columns for the same results. The diagonal of the Figure shows the precision matrices for the quadratic lattices, which have a reasonable good approximation for lattices of the size 240×240 and larger. For non quadratic lattices, larger dimensions are needed, if one side of the lattice is smaller than 120, the approximation is imprecise the most.

random walk of order two precision matrices A similar pattern can be observed for a cohort of precision matrices of a random walk model of order two. Figure 3.12 shows those results analogue to Figure 3.11. But compared to the order one precision, the approximations with circulant matrices are not very close for the respective lattices. This is due to the fact, that the structure of such matrices is less Toeplitz, as a comparison between Figure 3.13 and

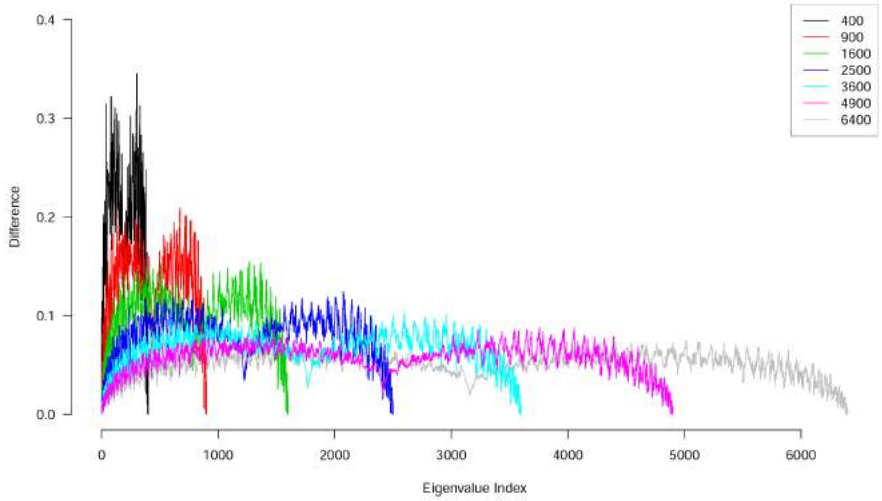

Figure 3.8: Relatively small precision matrices of a order one random walk model, such that the dimensions increase quadratically. The legend denotes the dimension of the corresponding matrix and hence the number of eigenvalues. Every line corresponds to the difference of the approximated and the true eigenvalues.

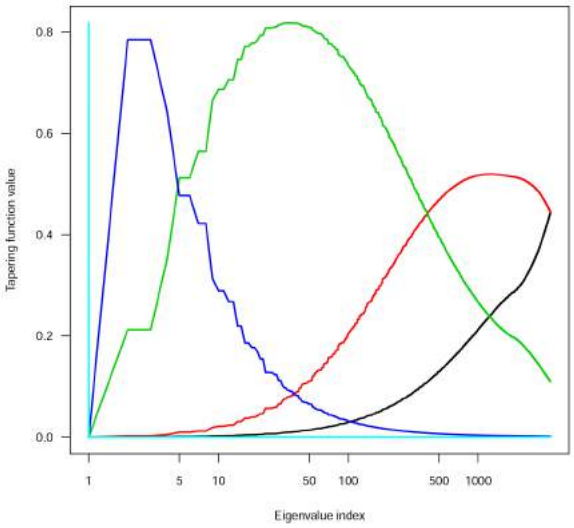

Figure 3.9: Tapering functions according to the true eigenvalues of a 3'600 × 3'600 random walk precision matrix of order one.

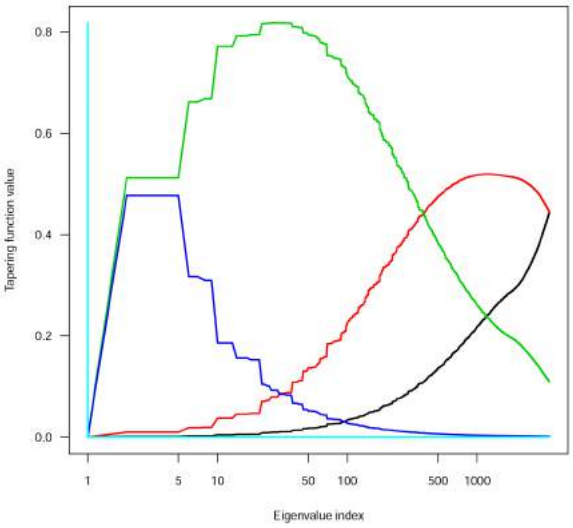

Figure 3.10: Tapering functions according to the approximated eigenvalues of a 3'600 × 3'600 random walk precision matrix of order one.

Figure 3.6 clearly shows. Individual tests for precision matrices of random walk models of order two showed, that the underlying lattice needs to be even larger than of size 600 × 600, for a reasonable approximation.

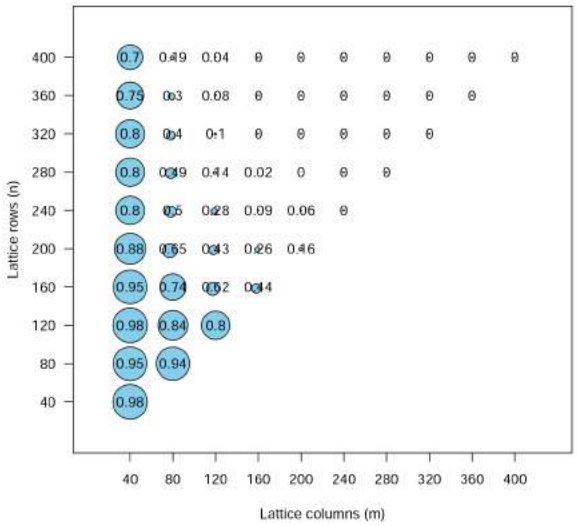

Figure 3.11: Relative number of rejected eigenvalues according to precision matrices of a random walk model of order one.

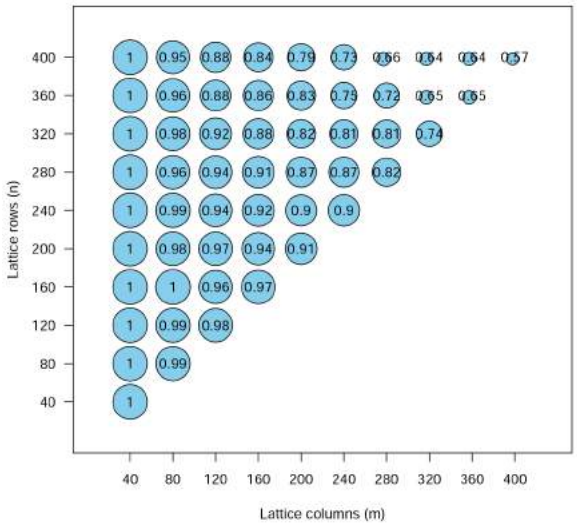

Figure 3.12: Relative number of rejected eigenvalues according to precision matrices of a random walk model of order two.

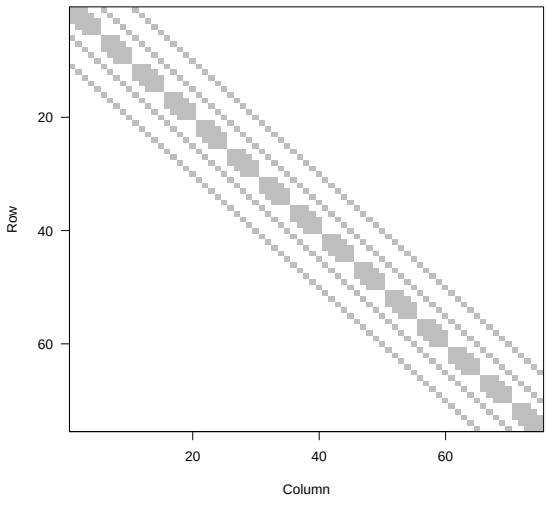

Figure 3.13: random walk of order two precision matrix of a 5×15 regular lattice.

Chapter 4

Applications

In this chapter, the original multiresolution decomposition from Chapter 1 is compared with the enhanced models, derived in Chapter 3. For this purpose the Lägeren mountain data is used, which is introduced in Chapter 2. To show, that the extended model of the multiresolution decomposition allows for data distributed on irregular lattices, also the oral cavity cancer SMR data is applied.

4.1 Multiresolution decomposition of the Lägeren data

First the original multiresolution decomposition method, from Chapter 1, is used to analyze the Lägeren mountain data. As an anterior approach, the data is analysed without changing zeros or low measurements to missing values, despite the arguments from Chapter 2 to interpret those as missing values. For the analysis the statistical software R with the provided functions from

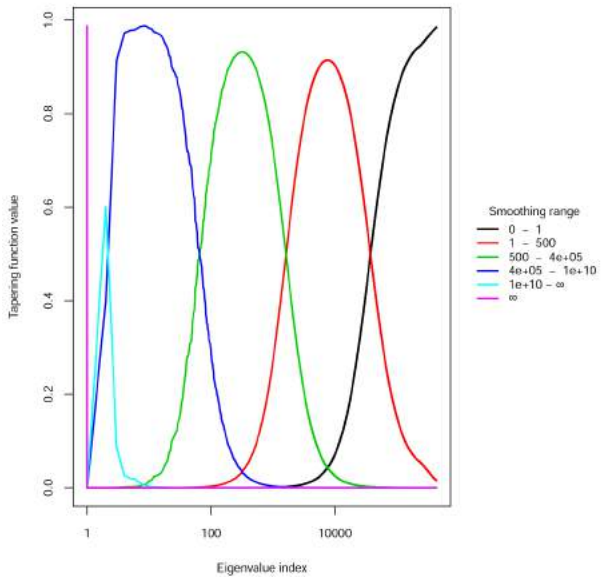

Figure 4.1: Signal independent tapering functions, for an underlying grid of size 400 × 1'100.

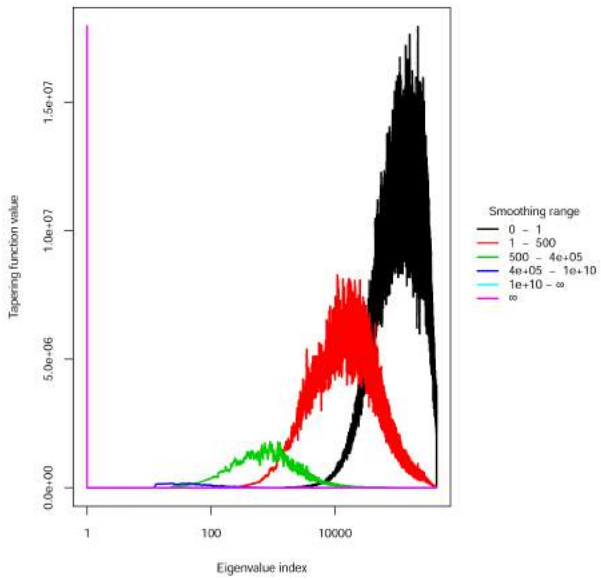

Figure 4.2: Signal dependent tapering functions, for an underlying grid of size 400 × 1'100.

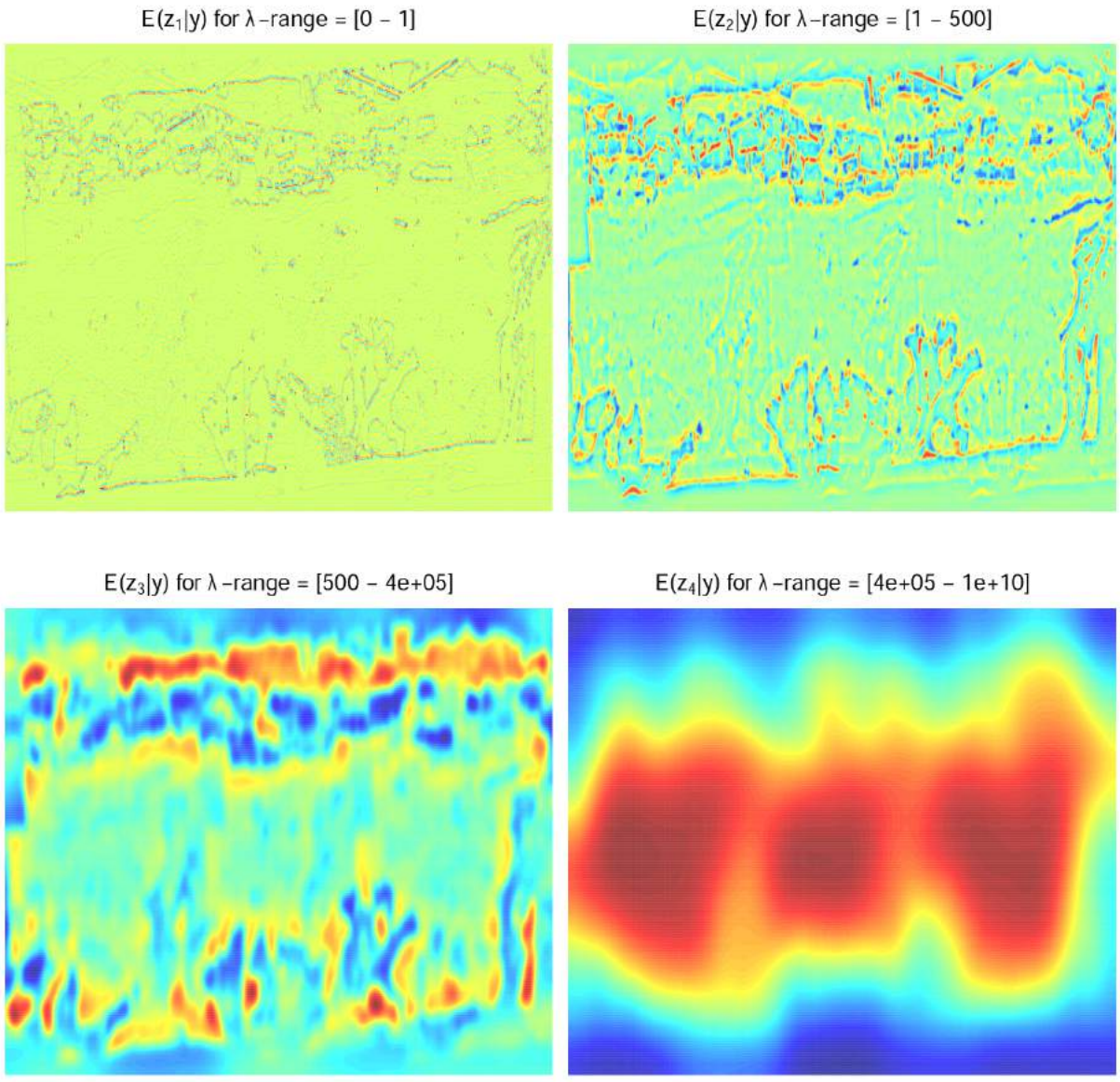

Figure 4.3: Scale dependent details of raw "Canopy Heights", summarized by their posterior means corresponding to the `mrbsizeR` analysis.

the package `mrbsizeR` is used.

Due to the large amount of values equal to zero, the data requires a transformation. Otherwise the analysis would be entirely dominated by those values and the normal assumption of the model would be violated, hence it would not show reliable results in any sense. The original data, including zeros and small values, are therefore log transformed for the first approach to analyze this data.

In the signal reconstruction step, 1'000 samples are generated with the multivariate $t_\nu(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ distribution and the hyper parameters are non informatively fixed to $\lambda = 0.2$, $\sigma = 6$ and $\nu_0 = 15$.

The tapering functions, shown in Figure 4.1 and 4.2, exhibit the desired pattern of approximately disjoint supports, for the chosen smoothing levels. Pursuant smoothing levels are

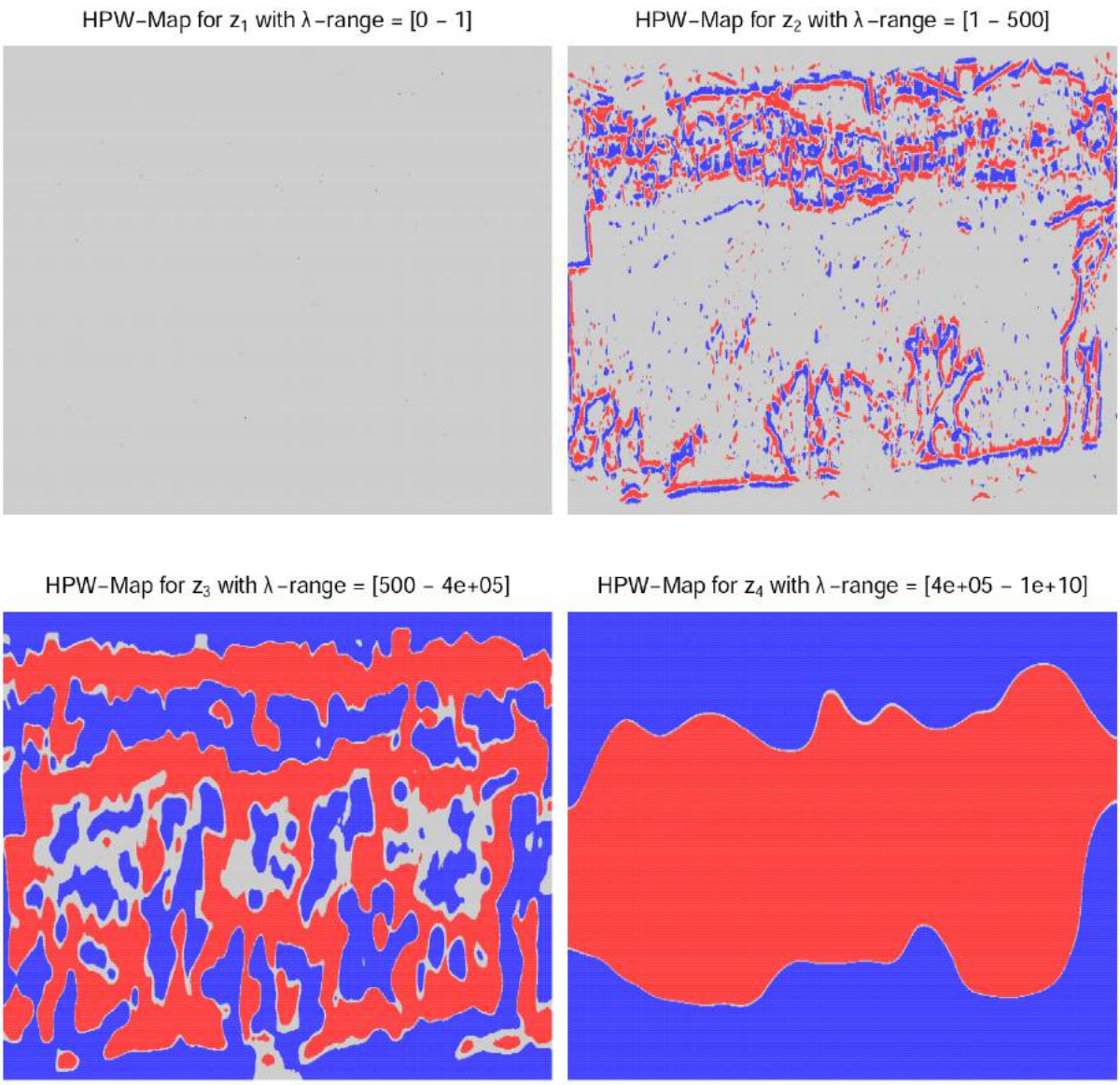

Figure 4.4: Highest pointwise maps of raw "Canopy Heights", corresponding to the `mrbsizeR` analysis.

determined with the help of the optimization procedure, which is implemented in the same `mrbsizeR` package. The following scale-dependent details, in Figure 4.3, seem to be fitting only at the first glance. If one examines larger areas with former values equal to zero, one recognizes, that large features appear which are not prevalent in the original data. This leads to wrong conclusions, especially in the point of view, that all of those original locations contain values equal or close to zero. Those artificial features are even better recognizable in the details $\boldsymbol{z}_2$ and $\boldsymbol{z}_3$. Moreover, the details $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ seem to be very blurred, and almost only the border points of the Lägeren forest are apparent. The credible features, according to the highest pointwise maps in Figure 4.4 do mirror all of those facts. High credible "Canopy Heights" measurements are colored in red and low ones in blue. One would expect, that at least detail $\boldsymbol{z}_2$ shows a

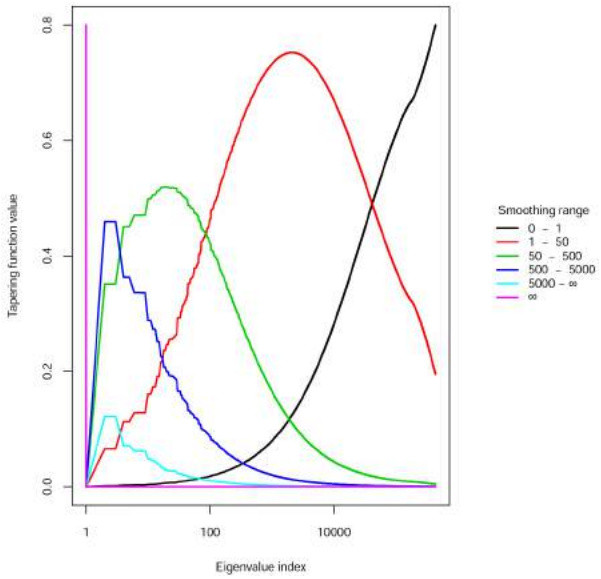

Figure 4.5: Tapering function according random walk model precision matrix, of dimension 440'000 × 440'000.

large amount of detailed credible features, but due to the huge amount of values equal to zero, those get washed out. Only in detail $\boldsymbol{z}_4$ the true underlying features are, according to its high smoothing levels, identified acceptably well. The same behavior occurs, if one imputes an overall mean value, for the missing values (see Appendix C).

4.1.1 Sample missing values

The modified model to sample missing values is applied next to the Lägeren mountain data and the signal reconstructed accordingly. A suitable precision matrix of a random walk model of order one, with dimensions 440'000 × 440'000 is used. The hyper parameters are chosen non informatively as $\alpha_x = 10$, $\beta_x = 0.1$ and $\alpha_y = 0.1$, $\beta_y = 0.0005$. The attentive reader notes, since the model includes a Gibbs sampler, further diagnostic of the resulting samples need to be concerned. According to Brooks and Gelman (1998), typical tools like trace, auto correlation plots and 95% as well as 5% quantiles are considered to assure the convergence of the sampler. Therefore, 2'000 samples are taken into account after a burn-in period of 1'000 draws.

Since the precision matrix of this model is different to the model above, the smoothing levels have to be determined newly. To compute the according tapering functions, the depicted eigenapproximation, from Chapter 3, is used. In order to receive comparable smoothing levels, to the model above, the known smoothing levels from above are used as a starting point. As Figure 4.5 illustrates, the smoothing levels for this model are slightly different. The chosen smoothing levels, $\lambda_1 = 0$, $\lambda_2 = 1$, $\lambda_3 = 50$, $\lambda_4 = 500$ and $\lambda_5 = 5'000$ result in necessary approximately disjoint supports of the tapering functions.

Figure 4.6 shows the details found by the multiresolution decomposition, with the new model to sample missing values, summarized by their posterior means. Detail $\boldsymbol{z}_1$, with a narrow smoothing range and small smoothing levels, shows large yellow areas, specifically in the locations where the missing values used to be. Moreover, very delicate features are recognizable in this

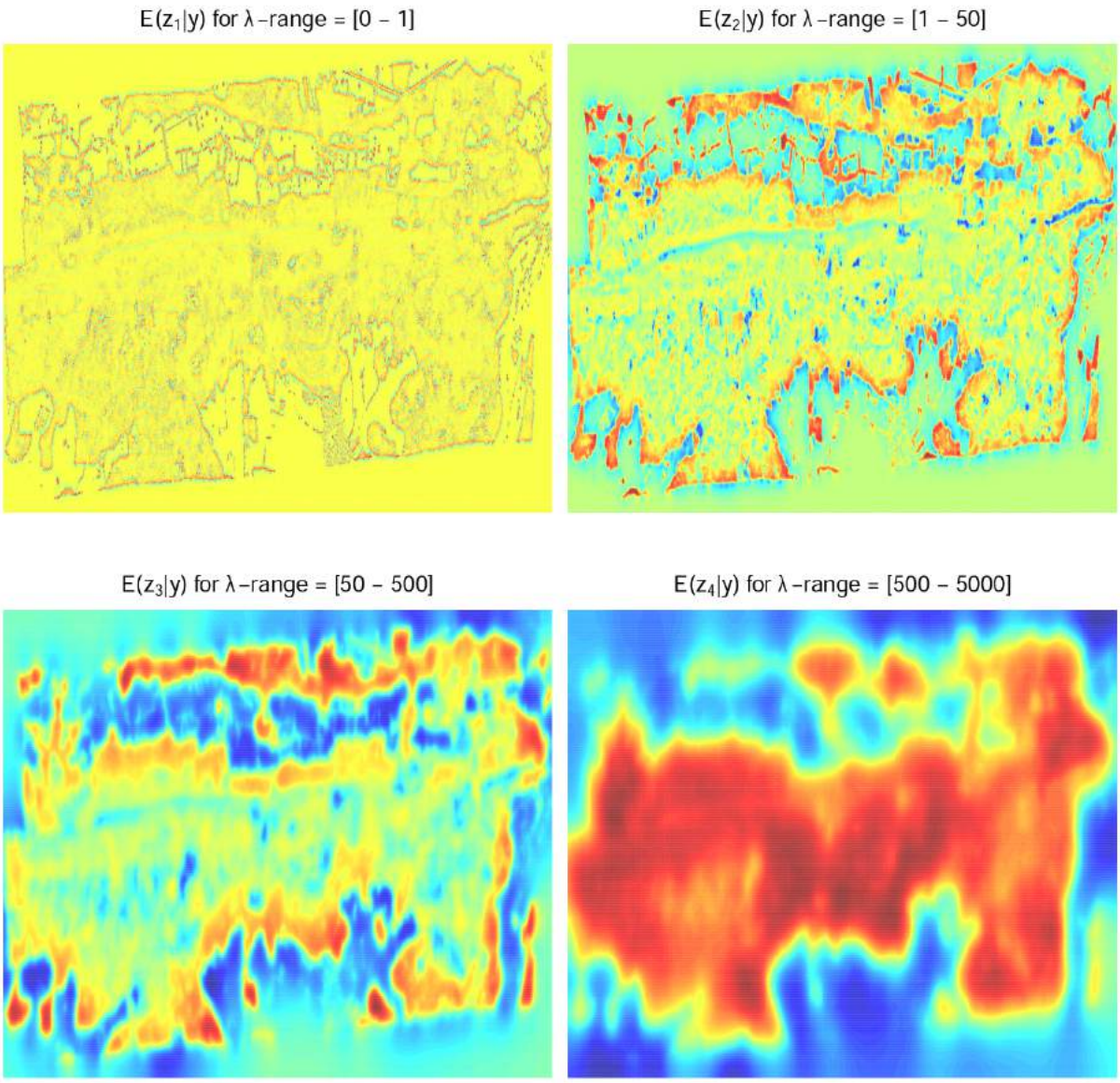

Figure 4.6: Scale dependent details of "Canopy Heights", summarized by their posterior means, whereupon missing values are sampled.

detail. Comparing the detail to its highest point wise maps counterpart in Figure 4.7, shows that the yellow areas have different underlying uncertainties. One can now recognize very well the credible features and as expected, are the large areas with former missing values not credible. Also, detail $\boldsymbol{z}_2$ shows exciting results, since in its highest probability map all known features, like the mountain ridge and so forth, are recognizable. It is still on a close scale and therefore reasonable that areas with former missing values are still not credible. On the other hand, detail $\boldsymbol{z}_3$ and $\boldsymbol{z}_4$, with distant smoothing levels, are able to color parts of those areas in blue. Most importantly, the model does not recognize artificial or non prevalent features and in every detail there is a reasonable amount of grey area between credible features.

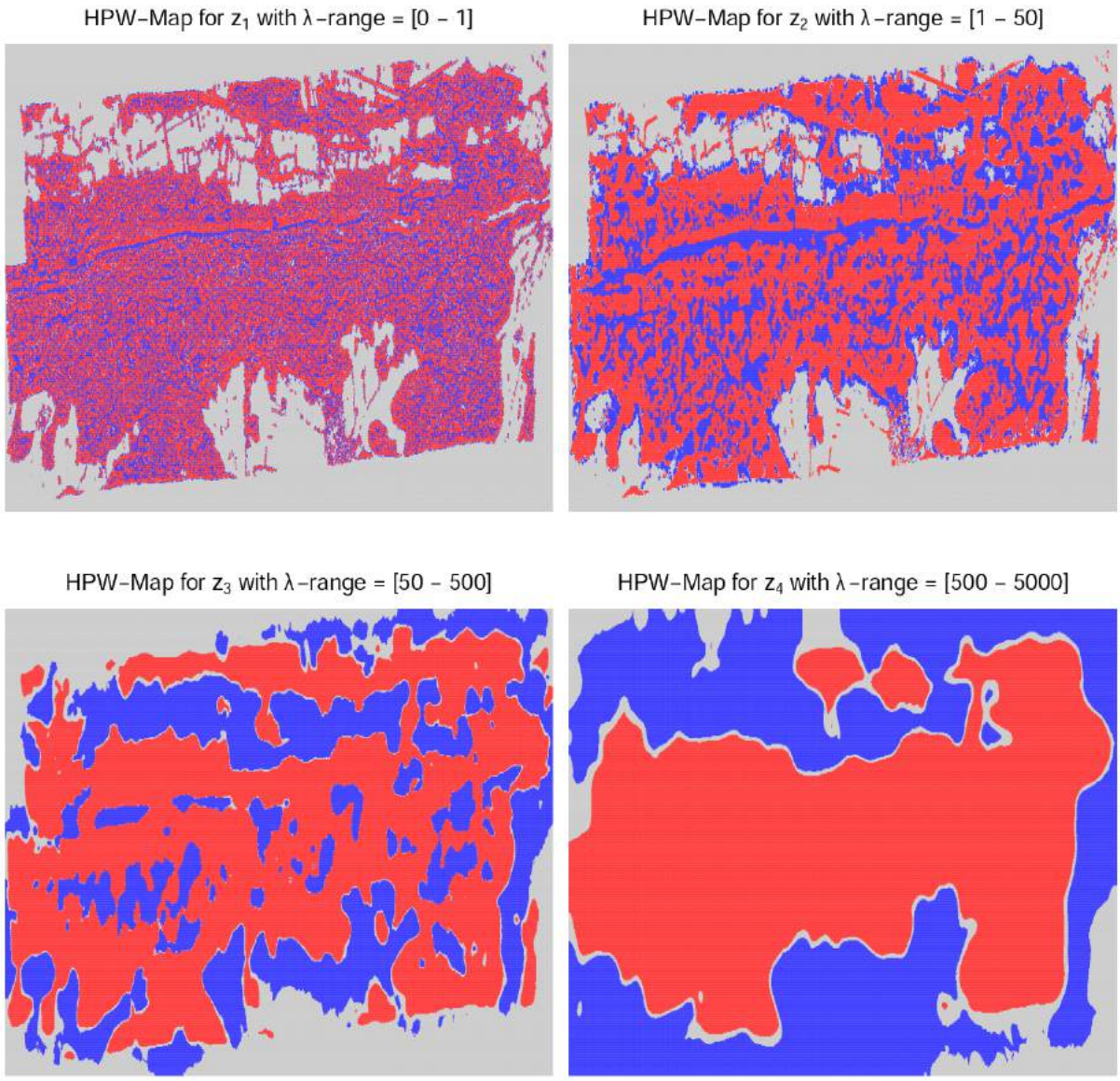

Figure 4.7: Highest pointwise maps of "Canopy Heights", whereupon missing values are sampled.

4.1.2 Removing missing values

The second model to deal with with missing values is also applied to the Lägeren mountain data. As outlined in the previous chapter, the respective locations are excluded from the analysis. For this reason, Figures 4.8 and 4.9 consists of a large amount of white locations. The smoothing levels are adjusted for this model to $\lambda_1 = 0$, $\lambda_2 = 1$, $\lambda_3 = 100$, $\lambda_4 = 4'000$, $\lambda_5 = 1000'000$, since the dimension of the precision matrix differs to both of the former models (see the tapering plot in Figure C.3). Nevertheless equal non informative hyperparamters are chosen, as well as the number of samples and burn-in period length, like in the previous application.

The corresponding scale dependent details and credible features are shown in Figure 4.8 and 4.9. The first detail, with the closest smoothing level, shows the data in clear details and the the corresponding highest probability map recognizes almost no points as credible. Typically,

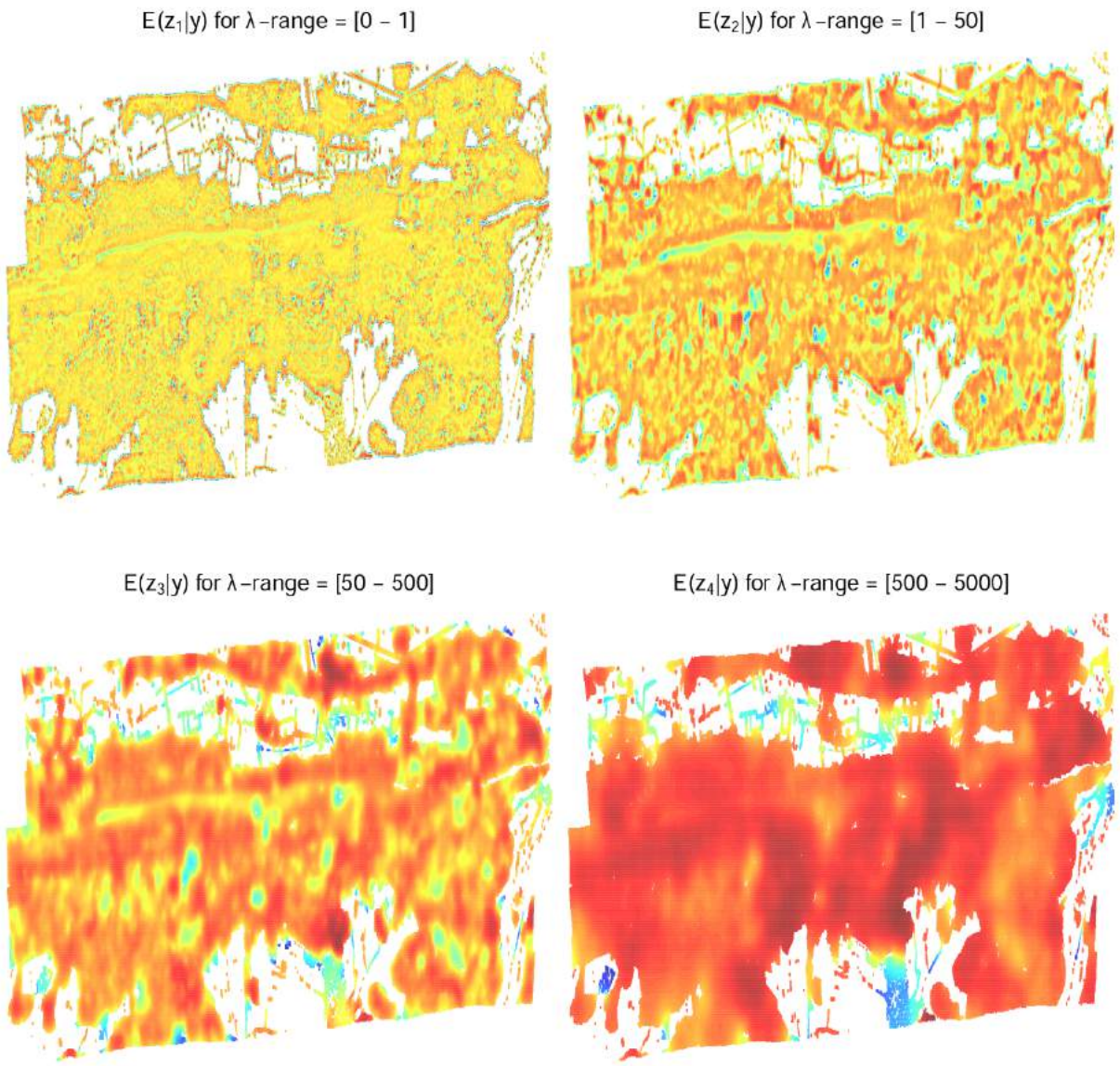

Figure 4.8: Scale dependent details of "Canopy Heights", summarized by their posterior means, whereupon missing values are removed.

in details with smoothing range including zero and another low smoothing level, portray the data in close detail and therefore not all locations are credible. In detail $\boldsymbol{z}_2$, the data is well recognized also, but despite of that, only few features are credible and large grey areas are depicted. On the other hand, the third detail distinguishes reasonably the important features of the data, which are also in the credible analysis apparent. Detail $\boldsymbol{z}_4$ is again dominated by a red area and contains areas with low measurements only at the borders. Otherwise, its highest pointwise map illustrates nicely prudential large credible areas, containing of sufficient enough grey areas in between. More experience is needed to treat corresponding smoothing at the borders of such areas. Different approaches to further modify the precision matrix are possible, as the work of Tiefelsdorf *et al.* (1999) shows.

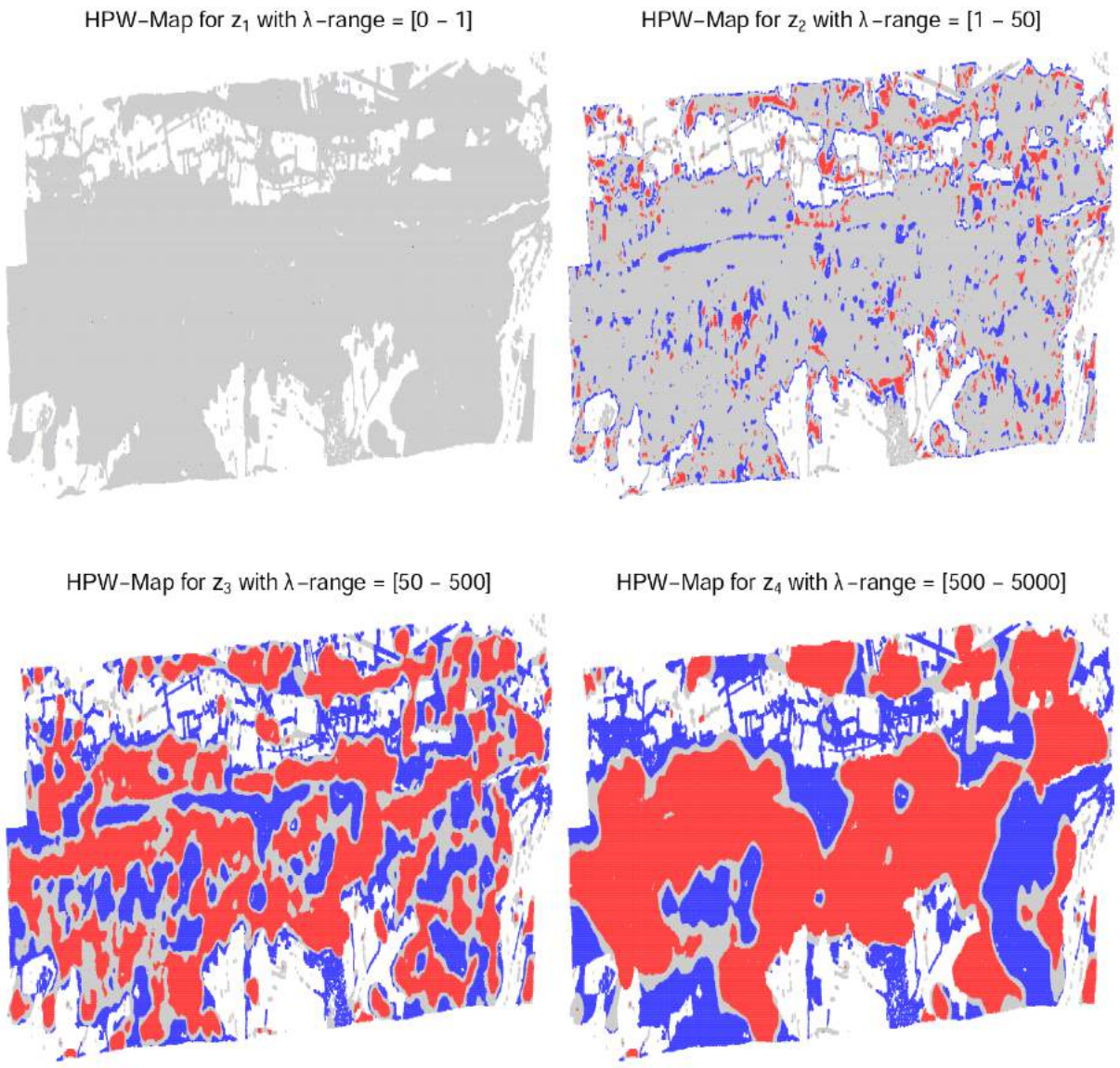

Figure 4.9: Highest pointwise maps of "Canopy Heights", whereupon missing values are removed.

4.2 Multiresolution decomposition of irregular lattice data

The direct consequence of the generalization for the signal reconstruction model of the multiresolution decomposition method, without any further modification, is the expansion from regular to irregular lattice data. Therefore the standardized mortality ratios from the oral cavity cancer data are applied to the generalized multiresolution decomposition. The usual precision matrix of a random walk model of order one is used, but since the lattice is irregular, one needs to weight the precision matrix according to the first neighbors of the respective districts. This is done with the adjacency matrix from the `spam` package and depicted in Figure 2.5. The hyperparameters are chosen non informatively also, as $\alpha_x = 0.001$, $\beta_x = 0.01$ and $\alpha_y = 0.005$, $\beta_y = 1$. Drawn are 2'000 samples by the normal response Gibbs sampler, after a burn-In period of 1'000

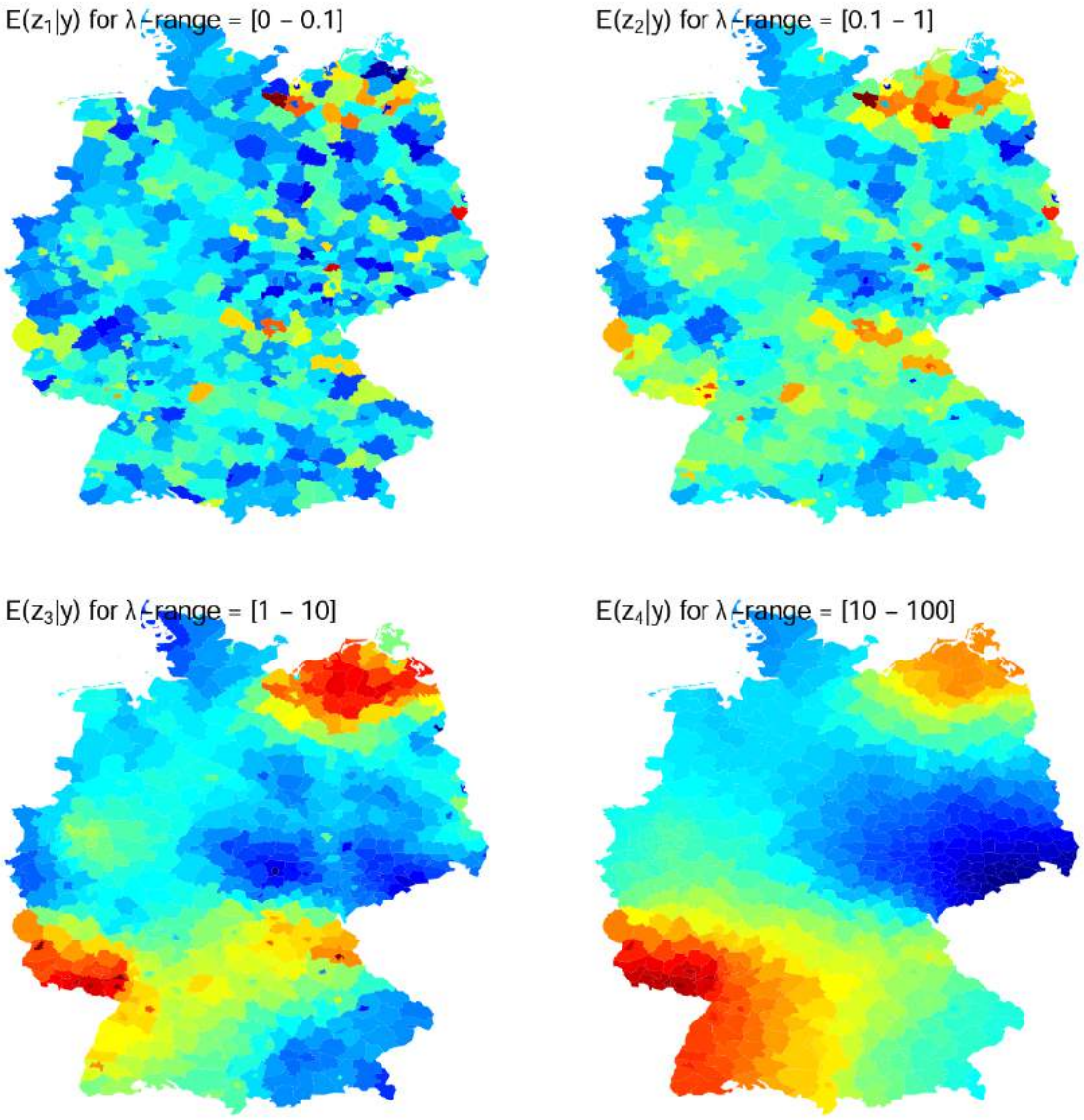

Figure 4.10: Scale dependent details of the oral cavity cancer SMRs in German districts, summarized by their posterior means.

draws. Moreover, the usual diagnostics are considered, to assure the convergence of the sampling procedure. The precision matrix of this application is of relatively small dimension, hence the eigenvalues can be calculated naively and an approximation is not necessary. The resulting tapering functions lead to the following smoothing levels, $\lambda_1 = 0$, $\lambda_2 = 0.1, \lambda_3 = 1, \lambda_4 = 10, \lambda_5 = 100$ (see Figure C.4).

The results of its multiresolution decomposition are depicted in the usual manner in Figure 4.10 and 4.11. For the credibility analysis are pointwise maps used, since the data consists only of 544 measurements. The details show, according to their smoothing range, the desired patterns. Details with low smoothing ranges, contain only few credible features, but for higher smoothing ranges the large credible areas are well recognized. Districts with only one neighbor are thereby not well smoothed out, if those contain relative extreme values.

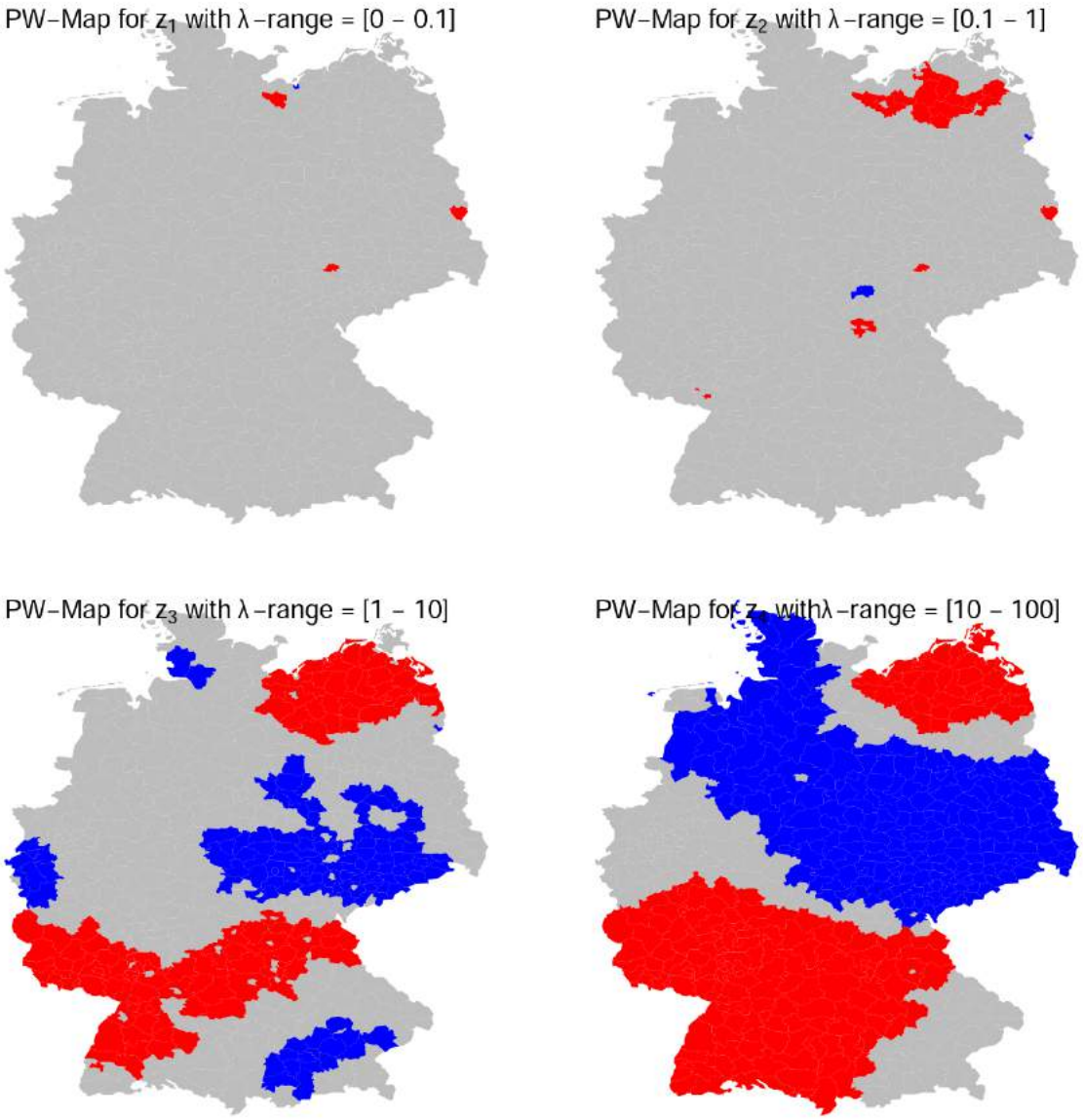

Figure 4.11: Pointwise maps of the oral cavity cancer SMRs in German districts.

Chapter 5

Conclusion

5.1 Results

In this master's thesis the existing multiresolution decomposition method was extended and generalized to incomplete signals and irregular lattice data. In order to efficiently apply the method, various steps were necessary, which are discussed separately in the following.

The generalized model for the Bayesian signal reconstruction of the multiresolution decomposition method showed promising results and proofs its flexibility with different approaches to deal with missing values. Moreover, it enabled the possibility to analyze data distributed on irregular lattices. In this context, the method is challenged only, with locations holding one nearest neighbor.

The derived model to sample missing values, yelledded exiting results and compared with the other approaches applied to the data, the most reasonable ones. The reduction of the influence of the likelihood function to non missing values, propagates the necessary uncertainties throughout the decomposition until the credibility analysis. Features are well recognized in all the different details and locations in large areas of former missing values became only credible for distant smoothing levels. Since, one could not differentiate between locations, where no measurements were taken and the measurement is actually zero, the data is still not fully accurately analyzed. But an improvement is only possible through this distinction in the data itself.

The model which removes missing values, was not in particular fitting for the Lägeren mountain data, since most of the values treated as missing, are equal to zero and for others, no observations were taken. Nevertheless, the application to the data proofed the usefulness of this model, since it still captured the important features of the data. One can imagine different purposes for this model, for instance if the area of interest includes a large feature like a lake or a river, where with absolute certainty no observation or measurement is available and if assigning a specific value to such a location, would be wrong.

The implementation of the implicitly restarted Arnoldi algorithm of the `spam` package was applied, in the course of this thesis, for various matrices of different dimensions and different forms. Nevertheless of the promising performance, while computing a small set of eigenvalues for large dimensional matrices, the algorithm provided no alternative to replace the fast Fourier based eigendecomposition, proposed in the original multiresolution decomposition. Henceforth, the algorithm proofed its usefulness, to test the accuracy of the eigenvalue approximation, out-

lined in Chapter 3.

The results of the eigenvalue approximation procedure, showed that the eigenvalue approximation of random walk precision matrices of order one by circulant matrices is sufficient for large enough lattices. On the other hand, for random walk precision matrices of order two, the approximation was not sufficiently accurate for the investigated lattice sizes. but, in the derived multiresolution decomposition models, the random walk precision matrix of order one is clearly favored, since it captures features of the data more precisely.

5.2 Outlook

The next step is to integrate the implicitly restarted Arnoldi method in the newest version of the `spam` package (which is currently version `spam 2.1-1`). Moreover, this method could be extended by making the implementation fit for sequential eigenvalue calculations. Therefore one needs only to save the intermediate results of the Fortran routine in the `R` environment and as soon as one starts further computations, handle those back to Fortran. Such an extension would be of high value in scientific procedures.

Further investigations regarding the eigenapproximation could cover the whole eigendecomposition of the matrix, i.e. to examine whether the circulant eigenvectors can be used as an approximation for precision matrices. With the help of eigenvectors, one can compute the signal-dependent tapering functions, in the outlined multiresolution decomposition method, to improve the smoothing selection procedure.

To make this method entirely fit for large dimensional problems, another important detail in the method needs an improvement. Namely, the optimization procedure to find the optimal smoothing levels. This problem was addressed in this thesis only for completeness, but the grid search which was proposed in the work of Holmström *et al.* (2011) is not suitable to deal efficiently with large dimensional precision matrices.

The multiresolution decomposition and its corresponding model, could be made even more generalized for non lattice data, i.e. to extend the method for data which has no grid structure and use the nearest distance to model a corresponding neighbor structure of the points.

To extend the functionality of the investigated method, one could introduce a smoothing level, which is dependent on the topology of the lattice points. With such a improved smooting level, the analyzed features would agree on the same level. Hence, it would be possible to count corresponding features and to make a probabilistic statement about the result.

Appendix A

Additional calculus

A.1 Gaussian Markov random field

After Rue and Held (2005), a Gaussian Markov random field (GMRF) is defined under the following preliminaries. Assume conditional dependence on a few neighbor sites only, where *neighbor* signifies sharing a common edge or boundary. Denote the neighbor relation with $r \sim s$ for sites $r \neq s$. The relation is symmetric, i.e. if $r \sim s$ then $s \sim r$. It is convenient to represent the dependence structure with an undirected, labelled graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes in a graph and $\mathcal{E}$ the set of edges $\{r, s\}$, $r \neq s \in \mathcal{V}$.

Definition A random vector $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^\mathsf{T}$ is a GMRF with respect to a labelled graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with mean $\boldsymbol{\mu}$ and (spd) precision matrix $\boldsymbol{Q}$ if its density is given by

$$\pi(\boldsymbol{y}) = (2\pi)^{-n/2} \det(\boldsymbol{Q})^{1/2} \exp\left(-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\mu})^\mathsf{T} \boldsymbol{Q} (\boldsymbol{y} - \boldsymbol{\mu})\right)$$

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

A.2 Canonical parameterization of a GMRF

After Rue and Held (2005), a Gaussian Markov random field (GMRF) $\boldsymbol{x}$ with respect to graph $\mathcal{G}$ with canonical parameters $\boldsymbol{b}$ and $\boldsymbol{Q} > 0$ has density

$$\pi(\boldsymbol{x}) \propto \exp\left(-\frac{1}{2}\boldsymbol{x}^\mathsf{T} \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{b}^\mathsf{T} \boldsymbol{x}\right)$$

i.e., the precision matrix is $\boldsymbol{Q}$ and the mean is $\boldsymbol{\mu} = \boldsymbol{Q}^{-1}\boldsymbol{b}$. One writes the canonical parameterization as

$$\boldsymbol{x} \sim \mathcal{N}_\mathcal{C}(\boldsymbol{b}, \boldsymbol{Q}).$$

A.3 Derivation of multivariate t-distribution parameters

The marginal posterior density function

$$\pi(\boldsymbol{x}|\boldsymbol{y}) \propto (\lambda_0 \boldsymbol{x}^\mathsf{T} \boldsymbol{Q} \boldsymbol{x} + ||\boldsymbol{y} - \boldsymbol{x}||^2 + \nu_0 \sigma_0^2)^{-(2n+\nu_0-1)(2)}, \tag{A.1}$$

which describes a multivariate t-distribution $\mathrm{t}_\nu(\boldsymbol{\mu}, \boldsymbol{\sigma})$. The parameters for this distribution can be determined with the hep of the definition of the smoother in equation (1.10) and the probability density function of the multivariate t-distribution. One starts with

$$\begin{aligned}
&\lambda_0 \boldsymbol{x}^\mathsf{T}\boldsymbol{Q}\boldsymbol{x} + ||\boldsymbol{y} - \boldsymbol{x}||^2 + \nu_0\sigma_0^2 \\
&\qquad = \boldsymbol{x}^\mathsf{T}(\mathbf{I} + \lambda_0\boldsymbol{Q})\boldsymbol{x} + \boldsymbol{y}^\mathsf{T}\boldsymbol{y} - 2\boldsymbol{x}^\mathsf{T}\boldsymbol{y} + \nu_0\sigma_0^2 \\
&\qquad = \boldsymbol{x}^\mathsf{T}\boldsymbol{S}_{\lambda_0}^{-1}\boldsymbol{x} + \boldsymbol{y}^\mathsf{T}\boldsymbol{y} - 2\boldsymbol{x}^\mathsf{T}\boldsymbol{y} + \nu_0\sigma_0^2 \\
&\qquad = (\boldsymbol{x} - \boldsymbol{S}_{\lambda_0}\boldsymbol{y})^\mathsf{T}\boldsymbol{S}_{\lambda_0}^{-1}(\boldsymbol{x} - \boldsymbol{S}_{\lambda_0}\boldsymbol{y}) - \boldsymbol{y}^\mathsf{T}\boldsymbol{S}_{\lambda_0}\boldsymbol{y} + \boldsymbol{y}^\mathsf{T}\boldsymbol{y} + \nu_0\sigma_0^2 \\
&\qquad = \frac{1}{-\boldsymbol{y}^\mathsf{T}\boldsymbol{S}_{\lambda_0}\boldsymbol{y} + \boldsymbol{y}^\mathsf{T}\boldsymbol{y} + \nu_0\sigma_0^2}(\boldsymbol{x} - \boldsymbol{S}_{\lambda_0}\boldsymbol{y})^\mathsf{T}\boldsymbol{S}_{\lambda_0}^{-1}(\boldsymbol{x} - \boldsymbol{S}_{\lambda_0}\boldsymbol{y}) + 1
\end{aligned}$$

The multivariate t-distribution includes the variance ν, otherwise its mean $\boldsymbol{\mu}$ is not defined and the distribution is a multivariate Cauchy (Genz and Bretz, 2009). To account for this, one replaces the 1 in the nominator, in the last line above, with ν.

Then the multivariate t-distribution parameters can be read of the expression as

$$\begin{aligned}
\boldsymbol{\mu} &= \boldsymbol{S}_{\lambda_0}\boldsymbol{y} \\
\boldsymbol{\Sigma} &= \boldsymbol{S}_{\lambda_0}\left(\frac{\boldsymbol{y}^\mathsf{T}\boldsymbol{y} - \boldsymbol{y}^\mathsf{T}\boldsymbol{\mu} + \nu_0\sigma_0^2}{\nu}\right) \\
\nu &= \nu_0 + n - 1.
\end{aligned}$$

Appendix B

Arnoldi algorithm

B.1 Spam R wrapper function

In ths appendix are the R functions displayed, which are necessary to call the Fortran routines `dn_eigen_f` or `ds_eigen_f`. In addition the functions to format the input as well as the return values of the aforementioned routines, are listed.

```
setMode <- function(sMode, symmetric, f_routine) {
  if (sMode != 'L' & sMode != 'S') { sMode <- 'S' }

  if (sMode == 'L') { mode <- as.integer(1L) }
  if (sMode == 'S') { mode <- as.integer(2L) }

  return(mode)
}

getEigenval <- function(values, mode, dim, nEig, retOrder = FALSE) {
  result <- rep(NA, dim)

  orderedInd <- order(values[1:nEig], decreasing = TRUE)
  values <- values[orderedInd]

  if (mode == 'L')  { result[1:nEig] <- values }
  if (mode == 'S' ) { result[(dim - nEig + 1):dim] <- values }
  if (!retOrder) { return(result) }
    else { return(list(values = result, newOrder = orderedInd)) }
}

mk_cmplxentries <- function(z = is.numeric()) {
  if (dim(z)[2] != 2)
    stop("wrong format from fortran return: dn_eigen_f")

  cmplx <- NULL
  cmplx <- sapply(1:length(z[ , 1]),
                  function(x) { complex(real = z[x, 1], imaginary = z[x, 2]) })
```

```
  if (is.null(cmplx))
    stop("error while aggregating fortran return from two vectors to a complex")

  return(cmplx)
}

mk_cmplxvectors <- function(V = is.matrix(), numbEig, eigVal, dimMat) {
  formVec <- matrix(complex(real = 0, imaginary = 0),
                    nrow = dimMat, ncol = numbEig)

  first <- TRUE
  sapply(1:numbEig, function(x) {

    if (abs(Im(eigVal[x])) > getOption("spam.eps")) {
      if (first) {
        sapply(1:dimMat, function(y) {
          formVec[y, x] <<- complex(real = V[y, x], imaginary = V[y, x+1])
          if (x + 1 < numbEig) {
            formVec[y, x+1] <<- complex(real = V[y, x], imaginary = -V[y, x+1])
          }
        })
        first <- FALSE
      } else {
        first <- TRUE
      }
    } else {
      sapply(1:dimMat, function(y) {
        formVec[y, x] <<- complex(real = V[y, x], imaginary = 0)
      })
    }
  })

  return(formVec)
}

getEigenvec <- function(v, values, sym, ncv, mode, dimen, nEig) {
  if (is.null(v))
    stop("fortran returned NULL eigenvectors")

  if (sym)
    tmpM <- matrix(v, nrow = dimen, ncol = ncv)
  else {
    tmpM <- matrix(v[1:(dimen*nEig*2)], nrow = dimen, ncol = nEig*2)
    tmpM <- mk_cmplxvectors(V = tmpM, numbEig = nEig, eigVal = values,
                            dimMat = dimen)
  }
```

```
  rm(v)
  result <- matrix(NA, nrow = dimen, ncol = nEig)
  orderedInd <- getEigenval(values = values, mode = mode,
                            dim = dimen, nEig = nEig, retOrder = TRUE)$newOrder

  if (is.null(orderedInd))
    stop("error while formatting eigenvectors")

  result <- tmpM[ , 1:nEig, drop = FALSE]
  result <- result[ , orderedInd, drop = FALSE]

  return(result)
}

cat_verbose <- function(routine) {
  if (routine == 'ds_eigen_f') {
    cat("\n  the fortran routine for symmetric matrices with double precision
        was used: ds_eigen_f\n")
  }
  if (routine == 'dn_eigen_f') {
    cat("\n  the fortran routine for nonsymmetric matrices with double precision
        was used: dn_eigen_f\n")
  }
}

eigen_approx <- function(X,
                         nEigenVal    = 1,
                         ncvOpt,
                         mode         = 'S',
                         eigenvectors = FALSE,
                         verbose      = FALSE,
                         f_routine    = "ds_eigen_f",
                         devMode      = FALSE){

  # check & parse arguments
  if (class(X) != 'spam')
    stop("input matrix 'X' must be of class spam!")

  if (isSymmetric.spam(X) & X@dimension[1] <= nEigenVal)
    stop("nEigenVal: the number of eigenvalues to calculate must be smaller than
         the matrix dimensions")

  if (f_routine != "ds_eigen_f" && f_routine != "dn_eigen_f")
    stop("non valid fortran routine is specified")

  f_mode <- setMode(sMode = mode, symmetric = isSymmetric.spam(X),
```

```
                    f_routine = f_routine)
fortran_object <- result <- list(NULL)

  fortran_object <- .Fortran (f_routine,
                          maxnev   = as.integer(nEigenVal),
                          ncv      = as.integer(ncvOpt),
                          n        = as.integer(X@dimension[1]),
                          iwhich   = as.integer(f_mode),
                          na       = as.integer(X@dimension[1]),
                          a        = as.array(X@entries),
                          ja       = as.array(X@colindices),
                          ia       = as.array(X@rowpointers),
                          v        = vector("double", X@dimension[1]*ncvOpt),
                          d        = vector("double", nEigenVal),
                          vf       = as.integer(ifelse(devMode, 1L, 0L)),
                          iparam   = vector("integer", 11),
                        NAOK       = getOption("spam.NAOK"),
                        PACKAGE    = "spam")

  result <- list ("nEigenVal"    = fortran_object$maxnev,
                  "Mode"         = fortran_object$which,
                  "Eigenvectors" = if (eigenvectors) { fortran_object$v }
                                    else { NULL },
                  "Eigenvalues"  = fortran_object$d,
                  "nconv"        = fortran_object$iparam[5])
}

if (f_routine == "dn_eigen_f") {
  fortran_object <- .Fortran (f_routine,
                          maxnev   = as.integer(nEigenVal),
                          ncv      = as.integer(ncvOpt),
                          n        = as.integer(X@dimension[1]),
                          iwhich   = as.integer(f_mode),
                          na       = as.integer(X@dimension[1]),
                          a        = as.array(X@entries),
                          ja       = as.array(X@colindices),
                          ia       = as.array(X@rowpointers),
                          v        = vector("double", X@dimension[1]*ncvOpt),
                          dr       = vector("double", nEigenVal),
                          di       = vector("double", nEigenVal),
                          vf       = as.integer(ifelse(devMode, 1L, 0L)),
                          iparam   = vector("integer", 11),
                        NAOK       = getOption("spam.NAOK"),
                        PACKAGE    = "spam")

  result <- list ("nEigenVal"    = fortran_object$maxnev,
```

```
                    "Mode"         = fortran_object$which,
                    "Eigenvectors" = if (eigenvectors) { fortran_object$v }
                                     else { NULL },
                    "Eigenvalues"  = mk_cmplxentries(cbind(fortran_object$dr,
                                                           fortran_object$di)),
                    "nconv"        = fortran_object$iparam[5])
  }

  if (is.null(result))
    stop("error while calling fortran routine, check (control) arguments")

  if (verbose)
    cat(paste("\n ", nEigenVal, "eigenvalues requested and", result$nconv,
              "converged\n"))

  if (nEigenVal != result$nconv)
    warning(paste("only", result$nconv, "instead of", nEigenVal,
                  "eigenvalues converged, try to increase 'control = list(ncv)'"))

  return(result)
}


eigen.spam <- function (x, nev = 1, symmetric = TRUE, only.values = TRUE,
                        control = list()) {
  con <- list(mode         = 'S',
              ncv          = NULL,
              spamflag     = FALSE,
              verbose      = FALSE)

  nmsC <- names(con)
  con[(namc <- names(control))] <- control
  if (length(noNms <- namc[!namc %in% nmsC]))
    warning("unknown names in control: ", paste(noNms, collapse = ", "))

  ifelse(!con$verbose, vFlag <- FALSE, vFlag <- TRUE)

  minDimARPACK <- 16 # arpack routines cant handle very small matrices

  resContainer <- list(NULL)

  # dispatching
  if ((!con$spamflag && prod(dim(x)) <= getOption("spam.inefficiencywarning"))||
      (con$spamflag && prod(dim(x)) <= minDimARPACK)) {
      warning("The eigenvalues are calculated with the function eigen from the
              base package.")
```

```
  if (con$spamflag) { warning(paste("Even, the 'spamflag = TRUE'. Since the
                                    matrix dimension is smaller than "),
                              minDimARPACK) }

  resContainer <- base::eigen(x           = x,
                              symmetric   = symmetric,
                              only.values = only.values)

  resEig       <- resContainer$values
  resVec       <- resContainer$vectors

} else {
# --- using ARPACK ------------------------------------------------------ #

  if (!is.spam(x))
    try(x <- as.spam(x))

  if (x@dimension[1] != x@dimension[2])
    stop("non-square matrix in 'eigen'")

  if (nev >= x@dimension[1] || nev >= x@dimension[2] || nev <= 0)
    stop ("the number of asked eigenvalues is higher or equal the dimension of
          the input matrix")

  if (symmetric) { ncvMaxMin <- 100 } else {  ncvMaxMin <- 100 }
  if (con$ncv > x@dimension[1] || con$ncv < nev || is.null(con$ncv)) {
    inpNcv <- min(x@dimension[1] - 1, max(2 * nev + 1, ncvMaxMin)) }
  else { inpNcv <- con$ncv }

  if (con$mode == 'S' | con$mode == 'L') {
    inpMode <- con$mode
  } else {
    inpMode <- 'S'
    warning("invalid mode selected; set to 'S'")
  }

  # symmetric double
  if (isSymmetric.spam(x) && symmetric) {
    resContainer <- eigen_approx(X           = x,
                                 nEigenVal   = nev,
                                 ncvOpt      = inpNcv,
                                 mode        = con$mode,
                                 eigenvectors = !only.values,
                                 verbose     = vFlag,
                                 f_routine   = 'ds_eigen_f',
                                 devMode     = FALSE)
```

```
    if (con$verbose) { cat_verbose(routine = 'ds_eigen_f') }

    resEig       <- getEigenval(values        = resContainer$Eigenvalues,
                                mode          = inpMode,
                                dim           = x@dimension[1],
                                nEig          = resContainer$nconv)
    if(!only.values) {
      resVec     <- getEigenvec(v             = resContainer$Eigenvectors,
                                values        = resContainer$Eigenvalues,
                                ncv           = inpNcv,
                                sym           = TRUE,
                                dimen         = x@dimension[1],
                                mode          = inpMode,
                                nEig          = resContainer$nconv)
    }
  }

  # nonsymmetric double
  if (!isSymmetric.spam(x) | !symmetric) {
    resContainer <- eigen_approx(X             = x,
                                 nEigenVal     = nev,
                                 ncvOpt        = inpNcv,
                                 mode          = con$mode,
                                 eigenvectors  = !only.values,
                                 verbose       = vFlag,
                                 f_routine     = 'dn_eigen_f',
                                 devMode       = FALSE)

    if (con$verbose) { cat_verbose(routine = 'dn_eigen_f') }

    resEig       <- getEigenval(values        = resContainer$Eigenvalues,
                                mode          = inpMode,
                                dim           = x@dimension[1],
                                nEig          = resContainer$nconv)
    if (!only.values) {
      resVec     <- getEigenvec(v             = resContainer$Eigenvectors,
                                values        = resContainer$Eigenvalues,
                                ncv           = inpNcv,
                                sym           = FALSE,
                                dimen         = x@dimension[1],
                                mode          = inpMode,
                                nEig          = resContainer$nconv)
    }
  }

# ------------------------------------------------------------------------ #
```

```
  }

  if (only.values) { return (list("values"  = resEig)) }
     else          { return (list("values"  = resEig, "vectors" = resVec)) }
}
```

B.2 Spam Fortran routine

The Fortran subroutine `dn_eigen_f` includes the ARPACK calls to start the implicitly restarted Arnoldi method. It is based on sample code, from the selfsame ARPACK library, named `dndrv1` (Lehoucq *et al.*, 1998).

```
c
      subroutine dn_eigen_f(maxnev, ncv, n, iwhich,
     &                      na, a, ja, ia,
     &                      v, dr, di, vf , iparam)
c
      implicit none
c
      integer            maxnev, ncv, n, na, vf,
     &                   iwhich
c
c     %--------------%
c     | Local Arrays |
c     %--------------%
c
      integer            iparam(11), ipntr(14),
     &                   ja(*), ia(na+1)
c
      logical            select(ncv)
c
      Double precision
     &                   dr(maxnev+1), di(maxnev+1), resid(n),
     &                   v(n, ncv), workd(3*n),
     &                   workev(3*ncv),
     &                   workl(3*ncv*ncv+6*ncv),
     &                   a(*)
c
c     %---------------%
c     | Local Scalars |
c     %---------------%
c
      character          bmat*1, which*2
      integer            ido, lworkl, info,
     &                   ierr, maxitr, ishfts, mode
      Double precision
```

```
     &                   tol, sigmar, sigmai
c
c      %------------%
c      | Parameters |
c      %------------%
c
       Double precision
     &                   zero
       parameter         (zero = 0.0D+0)
c
       include 'debug.h'
       ndigit = -3
       logfil = 6
       mngets = 0
       mnaitr = 0
       mnapps = 0
       mnaupd = vf
       mnaup2 = vf
       mneupd = 0
c
       bmat   = 'I'
c
       lworkl = 3*ncv*ncv+6*ncv
       tol    = zero
       ido    = 0
       info   = 0
c
       ishfts = 1
       maxitr = 1000
       mode   = 1
c
       iparam(1) = ishfts
       iparam(3) = maxitr
       iparam(7) = mode
c
       if (iwhich .eq. 1) then
             which = 'LR'
       else
             which = 'SR'
       end if
c
 10    continue
c
c        %-------------------------------------------%
c        | Repeatedly call the routine DNAUPD and take |
c        | actions indicated by parameter IDO until    |
c        | either convergence is indicated or maxitr   |
```

```
c         | has been exceeded.                            |
c         %-----------------------------------------------%
c
          call dnaupd ( ido, bmat, n, which, maxnev, tol, resid,
     &         ncv, v, n, iparam, ipntr, workd, workl, lworkl,
     &         info )
c
          if (ido .eq. -1 .or. ido .eq. 1) then
             call d_ope (na, workd(ipntr(1)), workd(ipntr(2)),
     &                   a, ja, ia)
             go to 10
       end if
c
       if ( info .lt. 0 ) then
          call errpr (info)
          goto 9000
       else
          call dneupd ( .true., 'A', select, dr, di, v, n,
     &          sigmar, sigmai, workev, bmat, n, which, maxnev, tol,
     &          resid, ncv, v, n, iparam, ipntr, workd, workl,
     &          lworkl, ierr )
          if ( ierr .lt. 0 ) then
             call errpr (ierr)
             goto 9000
          end if
       endif
c
 9000  continue
c
       end
c
```

Appendix C

Additional figures

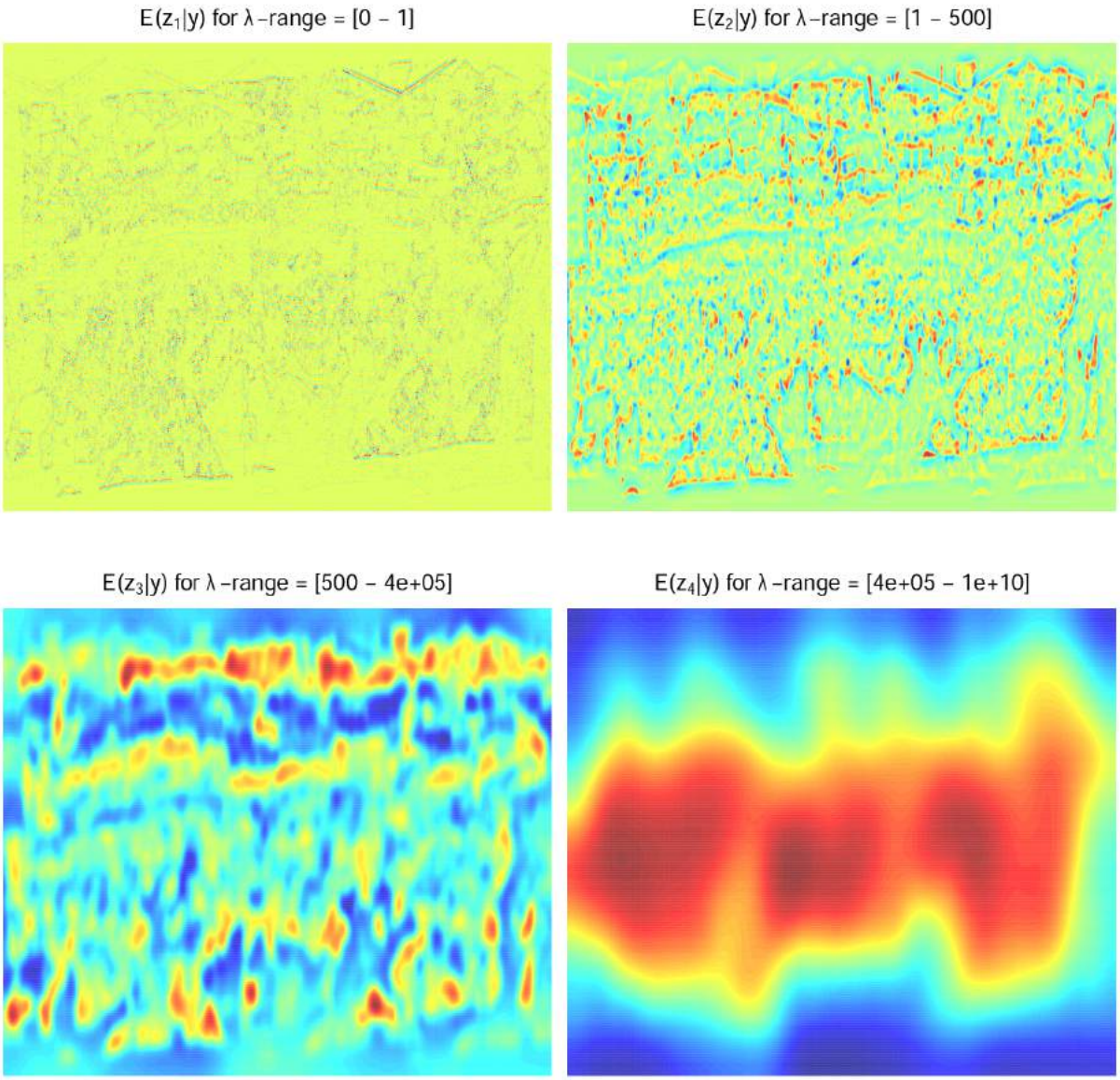

Figure C.1: Scale dependent details of mean imputed "Canopy Heights", summarized by their posterior means corresponding to the `mrbsizeR` analysis.

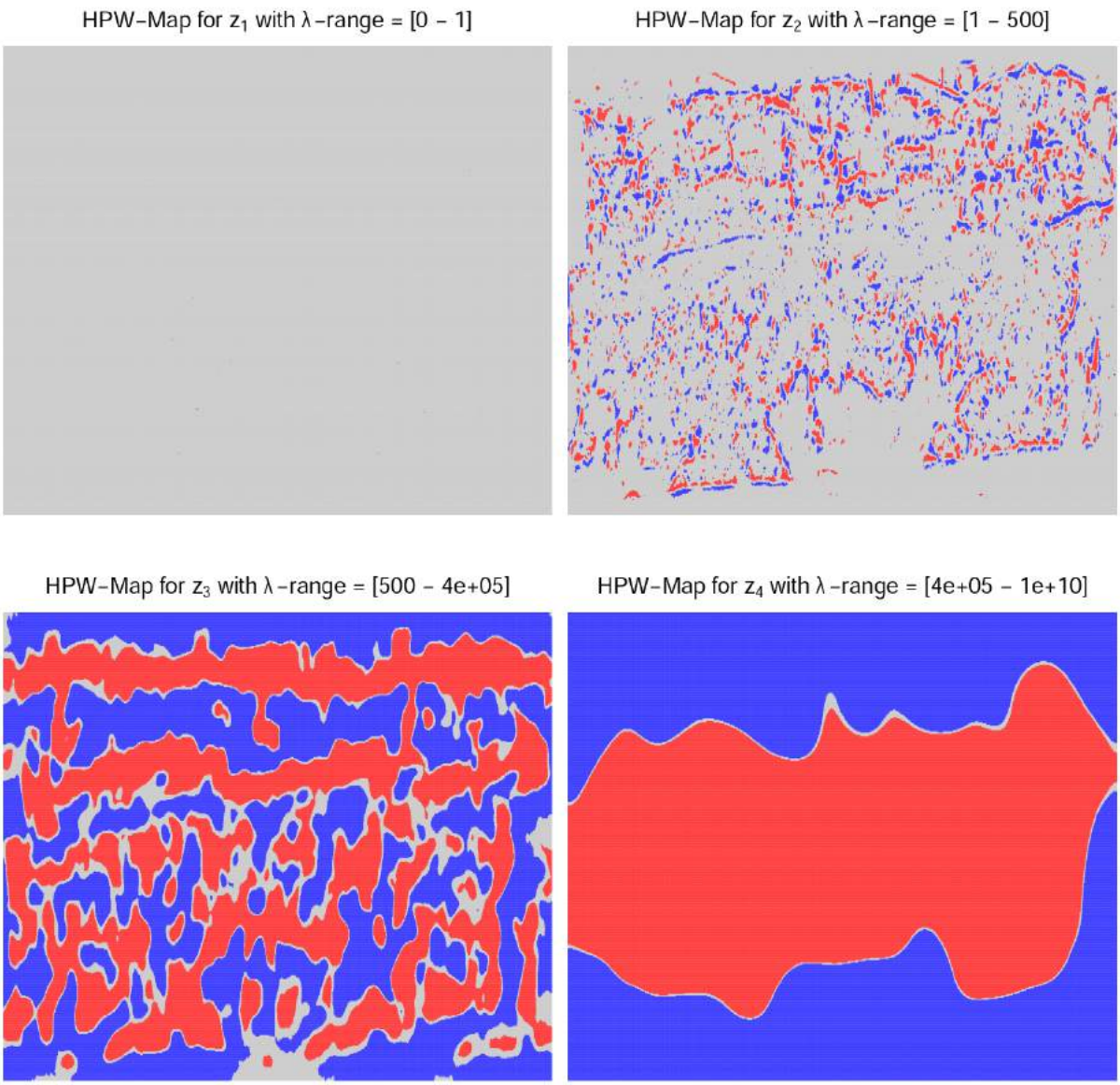

Figure C.2: Highest pointwise maps of mean imputed "Canopy Heights", corresponding to the `mrbsizeR` analysis.

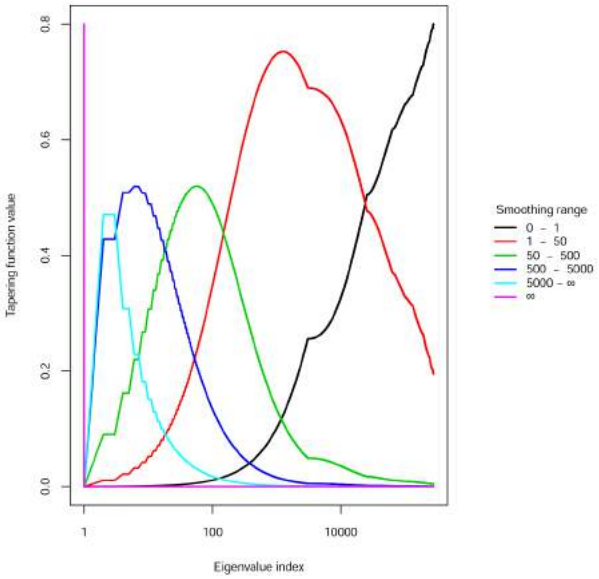

Figure C.3: Tapering function, according to the precision matrix to remove missing values of "Canopy Heights" measurements.

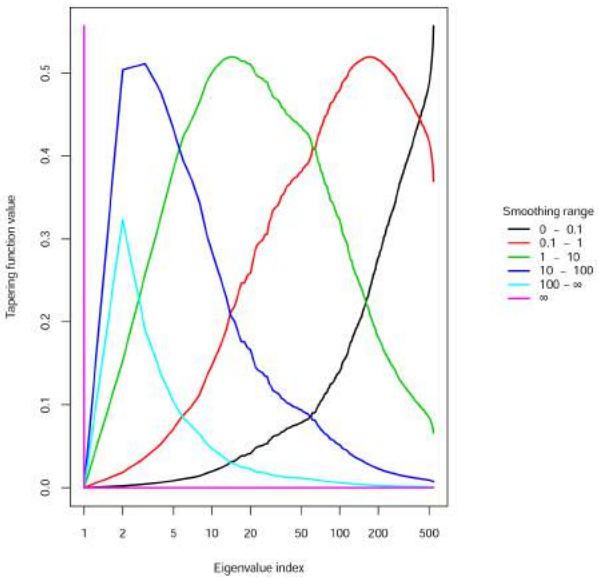

Figure C.4: Tapering function, according to the weighted precision matrix of the oral cavity cancer data.

Appendix D

Session Info

```
print(sessionInfo())
## R version 3.4.1 (2017-06-30)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.3 LTS
##
## Matrix products: default
## BLAS: /usr/lib/libblas/libblas.so.3.6.0
## LAPACK: /usr/lib/lapack/liblapack.so.3.6.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=de_CH.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=de_CH.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=de_CH.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=de_CH.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] spam_1.3-01   ggplot2_2.2.1 knitr_1.15.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.10     assertthat_0.1   digest_0.6.12    plyr_1.8.4
##  [5] gtable_0.2.0     magrittr_1.5     evaluate_0.10    scales_0.4.1
##  [9] highr_0.6        stringi_1.1.5    lazyeval_0.2.0   tools_3.4.1
## [13] stringr_1.2.0    munsell_0.4.3    compiler_3.4.1   colorspace_1.3-2
## [17] tibble_1.2
```

Bibliography

Biggs, N. (1993). *Algebraic Graph Theory*. Cambridge Mathematical Library. Cambridge University Press. 14

Brooks, S. P. and Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, **7**, 434–455. 32

Buluç, A., Fineman, J. T., Frigo, M., Gilbert, J. R., and Leiserson, C. E. (2009). *Parallel Sparse Matrix-vector and Matrix-transpose-vector Multiplication Using Compressed Sparse Blocks*. SPAA '09. ACM, New York, NY, USA. 15

Casella, G. and George, E. I. (1992). Explaining the Gibbs sampler. *The American Statistician*, **46**, 167–174. 15

Erästö, P. and Holmström, L. (2005). Bayesian multiscale smoothing for making inferences about features in scatterplots. *Journal of Computational and Graphical Statistics*, **14**, 569–589. 2, 9

Eugster, W., Zeyer, K., Zeeman, M., Michna, P., Zingg, A., Buchmann, N., and Emmenegger, L. (2007). Methodical study of nitrous oxide eddy covariance measurements using quantum cascade laser spectrometery over a Swiss forest. *Biogeosciences*, **4**, 927–939. 11

Furrer, R. and Sain, S. R. (2010). spam: A sparse matrix R package with emphasis on MCMC methods for Gaussian Markov random fields. *Journal of Statistical Software*, **36**, 1–25. 15

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (1995). *Bayesian Data Analysis*. Chapman & Hall, London. 2

Genz, A. and Bretz, F. (2009). *Computation of Multivariate Normal and T Probabilities*. Springer Publishing Company, 1st edition. 42

Golub, G. and Van Loan, C. (2013). *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press. 21

Gray, R. M. (2005). Toeplitz and circulant matrices: A review. *Commun. Inf. Theory*, **2**, 155–239. 21

Held, L. and Sabanés Bové, D. (2013). *Applied Statistical Inference: Likelihood and Bayes*. Springer Publishing Company, Incorporated. 16

Holmström, L. and Pasanen, L. (2011). MRBSiZer. `http://cc.oulu.fi/~lpasanen/MRBSiZer/`. Matlab 2008b Routines, accessed: 2017-04-20. 1, 3

Holmström, L. and Pasanen, L. (2012). Bayesian scale space analysis of differences in images. *Technometrics*, **54**, 16–29. 3

Holmström, L., Pasanen, L., Furrer, R., and Sain, S. R. (2011). Scale space multiresolution analysis of random signals. *Computational Statistics & Data Analysis*, **55**, 2840 – 2855. 1, 7, 15, 40

Knorr-Held, L. and Raßer, G. (2000). Bayesian detection of clusters and discontinuities in disease maps. *Biometrics*, **56**, 13–21. 14

Lehoucq, R., Sorensen, D., and Yang, C. (1998). *ARPACK Users' Guide: Solution of Large-scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*. SIAM e-books. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104). 21, 50

Morsdorf, F., Nichol, C., Malthus, T., and Woodhouse, I. H. (2009). Assessing forest structural and physiological information content of multi-spectral lidar waveforms by radiative transfer modelling. *Remote Sensing of Environment*, **113**, 2152 – 2163. 11

R Development Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0. v

Reuter, M., Biasotti, S., Giorgi, D., Patanè, G., and Spagnuolo, M. (2009). Discrete Laplace-Beltrami operators for shape analysis and segmentation. *Computers & Graphics*, **33**, 381 – 390. 3

Rue, H. and Held, L. (2005). *Gaussian Markov Random Fields: Theory and Applications*. Chapman & Hall/CRC, London. 2, 23, 41

Schneider, F. D., Leiterer, R., Morsdorf, F., Gastellu-Etchegorry, J.-P., Lauret, N., Pfeifer, N., and Schaepman, M. E. (2014). Simulating imaging spectrometer data: 3d forest modeling based on lidar and in situ data. *Remote Sensing of Environment*, **152**, 235 – 250. 11

Schuster, T. (2017). *mrbsizeR: Scale Space Multiresolution Analysis of Random Signals*. R package version 1.0.1. 1, 3

Strang, G. (1999). The discrete cosine transform. *SIAM Rev.*, **41**, 135–147. 3

Tiefelsdorf, M., Griffith, D. A., and Boots, B. (1999). A variance-stabilizing coding scheme for spatial link matrices. *Environment and Planning A*, **31**, 165–180. 35