mrbsizeR

Scale space multiresolution analysis in R

Master's Thesis in Biostatistics (STA495)

by

Thimo Schuster
12-482-444

supervised by

Prof. Dr. Reinhard Furrer

University of Zurich
Master Program in Biostatistics
Zurich, March 2017

University of Zurich[UZH]

Abstract

The purpose of this master's thesis is the implementation of a scale space multiresolution analysis procedure in **R**. Based on the MRBSiZer method by Holmström *et al.* (2011), the **R**-package `mrbsizeR` was created. MRBSiZer is a method to capture scale-dependent features in a random signal. It is based on scale space smoothing and has its focus on the analysis of images or spatial field defined on a regular grid. In contrast to common scale space procedures, differences of smooths at neighboring scales are used to separate features into distinct scale categories. A random signal can then be represented as a sum of such differences where each difference represents features relevant at a particular scale. Bayesian credibility analysis is used to infer which features found are "really there" and which are artifacts of random variation. Signal-independent and signal-dependent methods for finding useful smoothing parameters are presented. Even though these methods provide good smoothing level estimates, user fine tuning is inevitable.

To demonstrate the use of `mrbsizeR`, a vignette accompanying the **R**-package was created. The vignette consists of three examples which illustrate the method. NARCCAP data (Mearns *et al.*, 2014) is used for a detailed explanation of the single components and functions of `mrbsizeR`. The second example ensures the concordance of `mrbsizeR` with an existing Matlab implementation and in the third example, the application of `mrbsizeR` on spherical data is demonstrated.

The **R**-package `mrbsizeR` is an implementation of MRBSiZer which includes the functionality of its Matlab equivalent plus some additions. The results of this thesis extend the portfolio of scale space analysis methods in **R**. Possibilities for further research exist, for example the implementation of certain functions in **C** or further development on methods for finding useful smoothing levels.

Keywords mrbsizeR, MRBSiZer, SiZer, statistical scale space analysis, scale space smoothing, image analysis, analysis of spatial fields

Contents

1. Introduction

In the analysis of random signals, multi-scale representations are a helpful tool for understanding the structure and the different components of a signal. Such representations are usually generated by smoothing on different levels, where each smooth provides information relevant on a particular scale. Applied to spatial fields or images, this makes it possible to extract scale-dependent details that can be hard to spot when observing the random signal as a whole. The **R**-package `mrbsizeR` implements a scale space multiresolution analysis procedure based on the MRBSiZer method developed by Holmström *et al.* (2011), making the methodology available for a wider audience. The name of the package is an abbreviation for **M**ulti**R**esolution **B**ayesian **SI**gnificant **ZE**ro Crossings of Derivatives in **R**. It consists of a Bayesian scale space analysis method to (1) capture scale-dependent features in random signals and (2) to infer which of the captured features are credibly there and which are just artifacts of random variation. Where usual scale space procedures use a wide range of different smoothing levels for finding scale-dependent features, `mrbsizeR` employs differences of smooths at neighboring scales. The random signal can then be represented as a sum of these differences, where each difference of smooths represents features relevant on a particular scale. Such a representation is useful in the analysis of images and especially in the analysis of spatial fields, see for example Heersink (2013), where a similar approach was used for the analysis of earthwork compaction roller measurement values.

1.1. Outline

This thesis is structured as follows: The second chapter *Theory and Methods* gives an overview about the theory behind `mrbsizeR`. This includes the Bayesian model used (Section 2.2), the selection of useful smoothing levels (Section 2.4) as well as information to the credibility analysis methods used (Section 2.5). The application of `mrbsizeR` to real-world data is the focus of Chapter 3, *Application of mrbsizeR*. The application is documented in the vignette of `mrbsizeR`, a long-format guide created for the **R**-package with the goal to be a standalone introduction to the method for potential users. Chapter 4 provides a conclusion, summarizes the limitations of `mrbsizeR` and lists possibilities for future research.

1.2. Notation

The notation used for this thesis is inspired by Holmström *et al.* (2011). Scalar parameters, e.g. λ_i, are represented by lowercase letters. Vectors, e.g. $\boldsymbol{x} = [x_1, \ldots, x_N]$, are printed in boldface and matrices, e.g. $\boldsymbol{S}_\lambda$, are printed as uppercase letters in boldface.

1.3. Software and Hardware

`mrbsizeR` was developed in **R** (Version 3.3.1, 2016-06-21), a statistical software that is freely available at https://www.r-project.org/. Additional to the base packages, `spam`, `fields`, `maps`, `devtools`, `roxygen2`, `rmarkdown` and `knitr` have been used for analyses and the compilation of the report, see also Figure 1.1. The author's computing environment had the following specifications: Windows 10 Pro, Version 1607, 2.0 GHz Intel Core i7-3537U, 4 GB 1600 MHz DDR3.

```
sessionInfo()

## R version 3.3.1 (2016-06-21)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 14393)
##
## locale:
## [1] LC_COLLATE=German_Switzerland.1252  LC_CTYPE=German_Switzerland.1252
## [3] LC_MONETARY=German_Switzerland.1252 LC_NUMERIC=C
## [5] LC_TIME=German_Switzerland.1252
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] devtools_1.12.0 fields_8.10     spam_1.4-0      mrbsizeR_1.0.1
## [5] maps_3.1.1      knitr_1.15.1
##
## loaded via a namespace (and not attached):
## [1] magrittr_1.5  tools_3.3.1   withr_1.0.2   memoise_1.0.0 stringi_1.1.2
## [6] highr_0.6     digest_0.6.12 stringr_1.1.0 evaluate_0.10
```

Figure 1.1.: Information about the **R**-version, **R**-packages and OS used for creating `mrbsizeR` and this report.

2. Theory and Methods

This chapter provides an introduction and an overview of the methods and the theory used in `mrbsizeR`. It is divided in five sections. Section 2.1 puts `mrbsizeR` in context with scale space analysis and SiZer. Sections 2.2–2.5 discuss the theory behind `mrbsizeR`. For more information, see Holmström *et al.* (2011), Pasanen *et al.* (2013) and Erästö and Holmström (2005). An extensive review of different statistical scale space methods is available in Holmström and Pasanen (2016). Examples on how to use `mrbsizeR` can be found in the vignette in Chapter 3.

2.1. From Scale Space Analysis to mrbsizeR

The concept of scale space was introduced by Lindeberg (1994) as a tool for analyzing structures at different scales. As an example, Lindeberg mentions the concept of a branch of a tree: This concept makes only sense at a scale from centimeters to a few meters. It is meaningless on much smaller or larger scales, say, nanometers or kilometers. On a nanometer scale, one could describe the molecules that form the branch and on a kilometer scale, it would be possible to describe the forest in which the tree grows. This example highlights that, when observed on different scales, distinct features of an object can be discovered. Such a multi-scale representation can be of use in the image analysis and specifically in the analysis of real-world measurements. When analyzing collected data, one usually has only little knowledge about the scale-different features contained in the measurements. They might be hidden under some measurement noise or they could simply be unknown. Using the concept of scale space, it can be possible to reconstruct them.

For finding scale-different features in an object, scale space methods usually smooth the object on a wide range of smoothing levels. With an increasing amount of smoothing, small-scale details get lost but larger-scale details appear. In `mrbsizeR`, a similar approach is used. The difference is that, instead of a wide range, only a few, in some sense optimal, smoothing levels are used. Differences of the resulting smooths at neighboring scales are used for the subsequent `mrbsizeR` analysis. This separates the features into distinct scale categories more aggressively than usual scale space methods do.

SiZer, an abbreviation for **Si**gnificant **Zer**o crossings of derivatives, was introduced by Chaudhuri and Marron in 1999 (Chaudhuri and Marron, 1999) based on scale space ideas. Originally developed for smooths of curves and time series, it is a method that helps to assess which features of the smooth are “really there” and which are just sampling artifacts. In the original approach, the derivative of the smooth is investigated for significant increases and decreases over a wide range of smoothing levels. The result is the SiZer map, a graphical device that displays the significance of features with respect to both location and scale. A Bayesian version of SiZer for the analysis of images and spatial fields (Erästö and Holmström, 2005; Holmström and Pasanen, 2012) is used in `mrbsizeR`. The SiZer map can then be drawn for every difference of smooths and shows significance only with respect to the location.

The single steps of the scale space multiresolution procedure `mrbsizeR` are summarized in Figure 2.1. As the method focuses on the analysis of noisy images and random fields, a Bayesian model is used

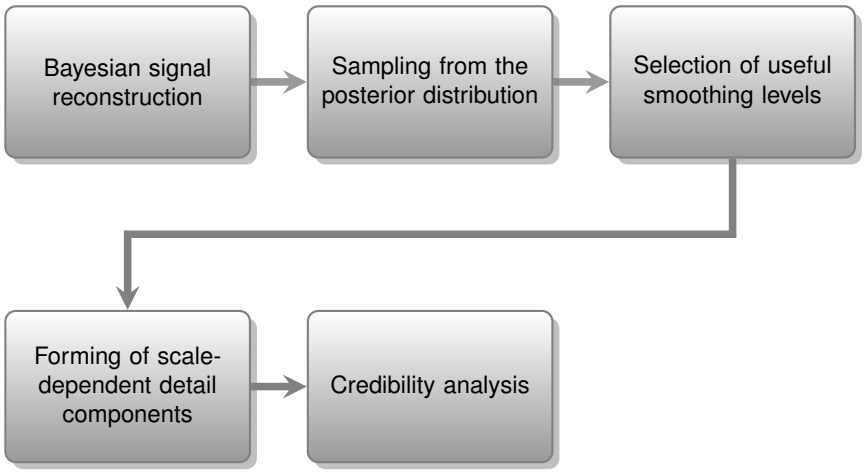

Figure 2.1.: Single steps of a `mrbsizeR` analysis.

to extract the underlying signal from its noisy observation. Then, samples from the resulting posterior distribution are drawn. After the selection of useful smoothing levels, either with or without taking the underlying signal into account, differences of smooths at neighboring scales can be created. As those scale-dependent detail components are random samples, a credibility analysis is inevitable for being able to tell which of the features found are “really there”. Detailed information to the single steps can be found in Sections 2.2–2.5.

2.2. The Bayesian Model

Collecting data usually leads to observations that are, in some sense, blurred. Due to measurement inaccuracies from the data-collecting device it is often not possible to make observations without having at least a small amount of noise in the data. Using a Bayesian model, `mrbsizeR` reconstructs the original signal using prior knowledge about the inaccuracy of the measurements. The Bayesian model used assumes that

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\epsilon},$$

where $\boldsymbol{y}$ is the observed noisy random signal, $\boldsymbol{x}$ is the true underlying signal and $\boldsymbol{\epsilon} = [\epsilon_1, \ldots, \epsilon_n]^T \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$ is the noise. For σ^2, an Inv-$\chi^2(\nu_0, \sigma_0^2)$ prior with density

$$\frac{1}{\Gamma\left(\frac{\nu_0}{2}\right)} \left(\frac{\nu_0 \sigma_0^2}{2}\right)^{\frac{\nu_0}{2}} (\sigma^2)^{-\left(\frac{\nu_0}{2}+1\right)} \exp\left(-\frac{\nu_0 \sigma_0^2}{2\sigma^2}\right) \tag{2.1}$$

is used. The prior for the underlying signal $\boldsymbol{x}$ is of the form

$$p(\boldsymbol{x}|\lambda_0, \sigma^2) \propto \left(\frac{\lambda_0}{\sigma^2}\right)^{(n-1)/2} \exp\left(-\frac{\lambda_0}{2\sigma^2} \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x}\right), \tag{2.2}$$

where $\boldsymbol{Q}$ is a precision matrix and defined in equation (2.6) (Holmström *et al.*, 2011). Combining priors (2.1) and (2.2) results in a $\mathcal{N}$-Inv-χ^2 prior distribution, see equation (A.1) in Appendix A.1. After combin-

ing this prior with the likelihood $p(\boldsymbol{y}|\boldsymbol{x}, \sigma^2)$ it is possible to write down the full posterior density $p(\boldsymbol{x}, \sigma^2|\boldsymbol{y})$. For being able to generate samples $\hat{\boldsymbol{x}}$ from the underlying signal $\boldsymbol{x}$, the marginal posterior density $p(\boldsymbol{x}|\boldsymbol{y})$ needs to be derived. This is possible by integrating σ^2 out of the full posterior. $p(\boldsymbol{x}|\boldsymbol{y})$ then follows a multivariate t-distribution $t_\nu(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ from which samples can be generated. A detailed derivation of $p(\boldsymbol{x}|\boldsymbol{y})$ can be found in Appendix A.1.

2.3. The Sum of Differences of Smooths

Let the underlying signal $\boldsymbol{x}$ be represented as an n-dimensional random vector. $\boldsymbol{S}_\lambda$ is a smoother, represented as an $n \times n$-matrix. $\lambda \geq 0$ is the corresponding smoothing parameter and defines the level of smoothing. A small λ corresponds to little smoothing, a large λ to heavy smoothing.

It is now possible to define a set of smoothing parameters,

$$0 = \lambda_1 < \lambda_2 < \cdots < \lambda_{L-1} < \lambda_L = \infty. \tag{2.3}$$

$\boldsymbol{S}_{\lambda_1} = \boldsymbol{S}_0$ corresponds to the so-called identity mapping where $\boldsymbol{S}_0\boldsymbol{x} = \boldsymbol{x}$. The smooth $\boldsymbol{S}_{\lambda_L}\boldsymbol{x} = \boldsymbol{S}_\infty\boldsymbol{x}$ represents the mean of $\boldsymbol{x}$. With (2.3), it is now possible to represent $\boldsymbol{x}$ as

$$\boldsymbol{x} = \sum_{i=1}^{L-1}(\boldsymbol{S}_{\lambda_i} - \boldsymbol{S}_{\lambda_{i+1}})\boldsymbol{x} + \boldsymbol{S}_{\lambda_L}\boldsymbol{x} \equiv \sum_{i=1}^{L-1}\boldsymbol{z}_i + \boldsymbol{z}_L, \tag{2.4}$$

the sum of differences of smooths at neighboring scales $\boldsymbol{z}_i$. Decomposition (2.4) can then be seen as some kind of multiresolution analysis of $\boldsymbol{x}$, where each component $\boldsymbol{z}_i$ represents details of $\boldsymbol{x}$ relevant at scale λ_i. As the true underlying signal $\boldsymbol{x}$ is not available, samples $\hat{\boldsymbol{x}}$ are used instead. Therefore, a posterior credibility analysis has to be done in order to find out which of the features found in $\hat{\boldsymbol{x}}$ are "really there" and which are only artifacts of random variation. Details about this are presented in Section 2.5.

2.4. The Selection of Smoothing Levels

For being able to apply transformation (2.4) to the samples $\hat{\boldsymbol{x}}$, the set of smoothing levels (2.3) has to be known. In comparison to common scale space methods, where a wide range of smoothing levels is used, `mrbsizeR` only requires a few of them. The crucial question is how to choose the smoothing levels such that the scale-dependent features in $\hat{\boldsymbol{x}}$ are captured in an optimal manner.

The smoother used in `mrbsizeR` is a roughness penalty smoother

$$\boldsymbol{S}_\lambda = (\boldsymbol{I} + \lambda\boldsymbol{Q})^{-1} \tag{2.5}$$

that minimizes the penalized loss defined by $\boldsymbol{Q}$,

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

$\boldsymbol{Q}$ is a positive semidefinite matrix and has the property that $\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x}$ measures the "roughness" of $\boldsymbol{x}$. Furthermore, $\boldsymbol{Q} = \boldsymbol{C}^T\boldsymbol{C}$, where $\boldsymbol{C}$ is the Laplacian matrix, also known as discrete Laplace operator.

$\boldsymbol{Q}$ is defined by

$$\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} = \sum_t \left(\sum_{s\sim t} x_s - 4x_t \right)^2 \tag{2.6}$$

where s and t are two grid locations and $s \sim t$ describes that they are neighbors. The inner summation of (2.6) is over all unordered pairs of neighboring locations. The resulting smoother (2.5) is used in all applications that involve data on a regular grid. For data on a sphere, see Section 2.6. Extensive information about $\boldsymbol{S}_\lambda$ can be found in the appendix of Holmström *et al.* (2011).

Using the eigendecomposition

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

it is possible to rewrite a smooth of $\boldsymbol{x}$ as

$$\boldsymbol{S}_\lambda \boldsymbol{x} = \sum_{j=1}^{n} (1+\lambda\gamma_j)^{-1} (\boldsymbol{v}_j^T \boldsymbol{x}) \boldsymbol{v}_j, \tag{2.7}$$

where $0 \leq \gamma_1, \cdots \leq \gamma_n$ are the eigenvalues and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ the corresponding orthonormal eigenvectors of $\boldsymbol{Q}$. A detailed derivation of (2.7) can be found in Appendix A.2.

Holmström *et al.* (2011) showed that it is possible to express a difference of smooths $\boldsymbol{z}_i$ as

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

with

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

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

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

for $i = L-1$, and

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

for $i = L$, using the eigenvalues γ_j and the corresponding eigenvectors $\boldsymbol{v}_j$ of $\boldsymbol{Q}$. The L sequences $\boldsymbol{\alpha}_i = [\alpha_j^{(i)}]_{j=1}^n, i = 1, \ldots, L$ (hereinafter referred to as "tapering functions") can now be used for finding appropriate smoothing parameters λ_i. The idea is to choose the set of smoothing parameters (2.3) in a way that the corresponding tapering functions $\boldsymbol{\alpha}_i$ are approximately disjoint (Holmström *et al.*, 2011). If transformation (2.4) is then applied to the data, each difference of smooths will show details relevant at other scales.

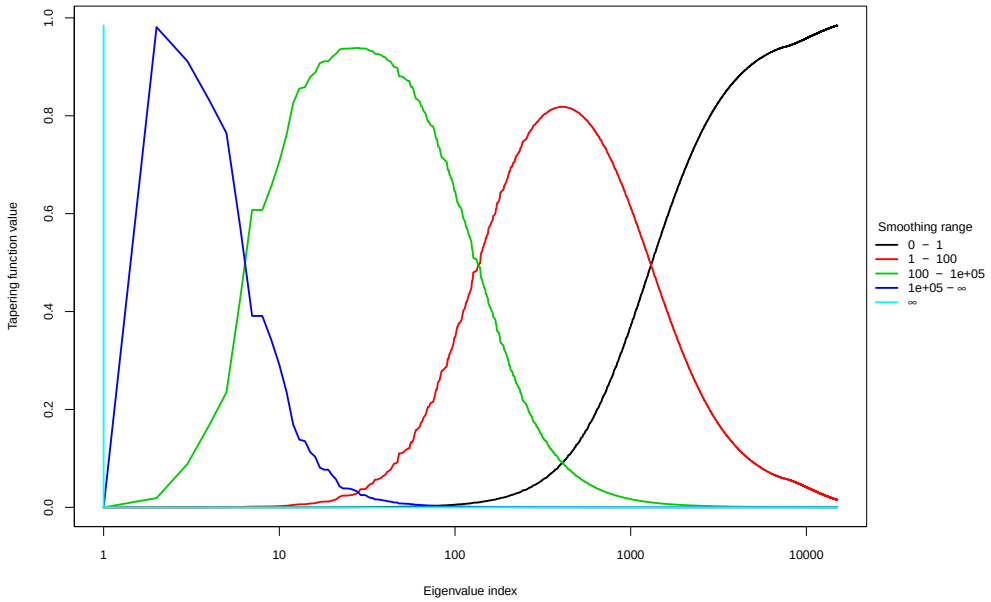

Figure 2.2.: Plot of the signal-independent tapering functions for the smoothing parameter sequence $[\lambda_1, \ldots, \lambda_5] = [0, 1, 100, 100000, \infty]$ on a 150-by-100 object.

Figure 2.2 is an exemplary plot of five tapering functions for a 150-by-100 object. The x-axis shows the index of the smoothing operator eigenvalues, where an index of 1 corresponds to the smallest eigenvalue. Tapering functions of small λ's involve large γ's, whereas large λ's involve small γ's. The tapering functions are approximately disjoint and will, when used for the `mrbsizeR` analysis, result in meaningful differences of smooths. Still, these tapering functions are solely dependent on the smoother and the size of the object used. If the analysis object is also taken into account, better estimates of useful smoothing levels can be achieved. One option for doing this is to draw signal-dependent tapering functions and do again some visual inspection of the disjointedness. Unfortunately, signal-dependent tapering functions are usually very irregular and visual inspection is difficult (Holmström *et al.*, 2011). The approach taken in `mrbsizeR` is to draw moving averages of the absolute values of these signal-dependent tapering functions to facilitate the visual inspection. An example can be found in the vignette in Chapter 3.

A more formal approach is the optimization of a suitable objective function G

$$G(\lambda_2, \lambda_3) = \sum_{i,j=1,2,3|i<j} \frac{|\tilde{\boldsymbol{\alpha}}_i^T \tilde{\boldsymbol{\alpha}}_j|}{\|\tilde{\boldsymbol{\alpha}}_i\| \|\tilde{\boldsymbol{\alpha}}_j\|}, \tag{2.8}$$

with respect to the λ's. Objective function (2.8) optimizes over two smoothing parameters and the result is the smoothing parameter sequence $[0, \lambda_2, \lambda_3, \infty]$. An extension to three smoothing parameters and the λ-sequence $[0, \lambda_2, \lambda_3, \lambda_4, \infty]$ is straightforward and also implemented in `mrbsizeR`. The optimization would theoretically also be possible in higher dimensions but this would cause additional complexity due to the non-linear optimization of G. Furthermore, for most applications, a smoothing parameter sequence $[0, \lambda_2, \lambda_3, \lambda_4, \infty]$ is sufficient.

All λ-estimation-methods proposed have one thing in common: User interaction can help to improve

the results. The λ-estimates provided are usually a very good starting point for finding scale-different details. Still, with small adjustments or the addition of another smoothing level it can be possible to highlight some scale-dependent details even better. The NARCCAP data example in the vignette (Chapter 3) gives an idea of what is meant. The smoothing level parameters provided by the visual inspection of the tapering functions and the optimization of G are slightly modified before being used for the `mrbsizeR` analysis.

2.5. The Posterior Credibility Analysis

As only a noisy observation $\boldsymbol{y}$ of the underlying signal $\boldsymbol{x}$ is available, the reconstructed signal $p(\boldsymbol{x}|\boldsymbol{y}) = \hat{\boldsymbol{x}}$ is used for analysis. $\hat{\boldsymbol{x}}$ is a random sample and one can generally not be sure about the validity of the results obtained. It is therefore vital to do a credibility analysis. Consider a `mrbsizeR` analysis where the smoothing levels were selected and the differences of smooths $\boldsymbol{z}_i$ are created. To infer which of the details found in the $\boldsymbol{z}_i$'s are really there and which are just artifacts of random variation, a posterior credibility analysis is necessary. `mrbsizeR` offers three different methods, illustrated next.

2.5.1. Pointwise Maps

Credibility analysis with pointwise (PW) maps is done for each location of every detail component $\boldsymbol{z}_i$ separately. The index set I of an array $[z_s]_{s\in I}$ (for simplification, the index i is left out from the components z_s) is split into three disjoint subsets (Erästö and Holmström, 2005; Holmström *et al.*, 2011):

$$\begin{aligned} I^b &= \{s|P(z_s > 0|\boldsymbol{y}) \geq \alpha\}, \\ I^r &= \{s|P(z_s < 0|\boldsymbol{y}) \geq \alpha\}, \\ I^g &= I\backslash(I^b \cup I^r). \end{aligned} \tag{2.9}$$

α is the credibility level and 0.95 is a typical choice. The allocation to the subsets (2.9) is done for each location independently over all samples. A location is therefore colored blue if $s \in I^b$, red if $s \in I^r$ and gray if $s \in I^g$. Blue, red and gray are the default colors chosen and could also be replaced by any other color.

As each location is treated independently, pointwise maps do often exhibit very small islands of credibility. This can make it hard to draw conclusions from them. The simultaneous approaches explained in Sections 2.5.2 and 2.5.3 flag locations as credible more conservatively than PW maps, but maximize the connectedness of credible locations.

2.5.2. Highest Pointwise Probability Maps

Instead of using pointwise inference, highest pointwise probability (HPW) maps employ simultanous inference over all locations. In simultaneous maps, the index-subsets J^b, J^r and $J^g = I\backslash J^b \cup J^r$ are used. The set I is decomposed into the J's so that

$$P(z_s > 0 \text{ for } s \in J^b \text{ and } z_s < 0 \text{ for } s \in J^r|\boldsymbol{y}) \geq \alpha. \tag{2.10}$$

As not all of the indices in sets I^b and I^r are also flagged credible in simultaneous maps, $J^b \subset I^b$, $J^r \subset I^r$ and $J^g \supset I^g$. Simultaneous approaches therefore exhibit more non-credible areas than pointwise maps,

but do on the other hand show a better connectedness of credible locations (Erästö and Holmström, 2005).

The HPW algorithm selects location in descending order with respect to their associated marginal posterior probabilities for as long as their joint probability is at least α (Pasanen *et al.*, 2013). Let S be a set and $|S|$ its number of elements. $N = |I^b| + |I^r|$ is the number of locations flagged as credible using PW maps. The events $(z_s > 0|\boldsymbol{y})|s \in I^b$ and $(z_s < 0|\boldsymbol{y})|s \in I^r$ are denoted with E_s. For the permutations $s_1, \ldots, s_N$ of the locations s in $I^b \cup I^r$

$$P(E_{s_1}) \geq P(E_{s_2}) \geq \cdots \geq P(E_{s_N}) \geq \alpha$$

is valid.

$$k = \max\{l|P(E_{s_1} \& \ldots \& E_{s_l}) \geq \alpha\}$$

is the maximum number of events E_s such that the joint probability is at least α. If $l \in \{1, \ldots, k\}$, s_l is credible and colored blue if $s_l \in I^b$ or red if $s_l \in I^r$. The rest of the locations s is non-credible and colored gray.

2.5.3. Simultaneous Credible Intervals

Another possibility for simultaneous inference over all locations are simultaneous credible intervals (CI). Let $\Delta > 0$ satisfy

$$P\left(\max_{s \in I} \left| \frac{z_s - E(z_s|\boldsymbol{y})}{\mathrm{Std}(z_s|\boldsymbol{y})} \right| \leq \Delta|\boldsymbol{y}\right) = \alpha.$$

where $E(z_s|\boldsymbol{y})$ is the posterior mean and $\mathrm{Std}(z_s|\boldsymbol{y})$ is the posterior standard deviation of z_s. The subsets J^b, J^r and J^g are defined as

$$\begin{aligned} J^b &= \{s|E(z_s|\boldsymbol{y}) - \Delta\mathrm{Std}(z_s|\boldsymbol{y}) > 0\} \\ J^r &= \{s|E(z_s|\boldsymbol{y}) + \Delta\mathrm{Std}(z_s|\boldsymbol{y}) < 0\} \\ J^g &= I \backslash J^b \cup J^r. \end{aligned}$$

Then (2.10) is satisfied. Simultaneous credible intervals flag locations as credible more conservative than HPW maps, which strive to maximize the number of credible locations (Erästö and Holmström, 2005).

2.6. mrbsizeR for Spherical Data

`mrbsizeR` can also be used to analyze data defined on a sphere. The main difference to the non-spherical version is that no Bayesian signal reconstruction is provided. Instead, samples for the scale space multiresolution analysis need to be available in advance. When analyzing spherical data, smoothing must be carried out in a way that takes into account the fact that the data is not defined on a rectangular grid but on a sphere. A smoother based on an analog of (2.6) is used in `mrbsizeR`. Details can be found in the appendix of Holmström *et al.* (2011).

3. Application of mrbsizeR

To simplify the usage of `mrbsizeR` to potential users, a vignette accompanying the package has been created. A vignette is a long-format guide of an **R**-package with the goal to provide important information and usability examples to users. Many existing **R**-packages have vignettes, and using `browseVignettes()` one can get an overview of all vignettes that are installed.

The vignette of `mrbsizeR` consists of three examples. For the first example, data from the North American Regional Climate Change Assessment Program (NARCCAP) was analyzed (Mearns *et al.*, 2014). The example gives detailed instructions on how to use `mrbsizeR`. Furthermore, background information about the `mrbsizeR` method and some theory is provided. The second example analyzes the photograph of a sketch pad. This photograph has already been used in Holmström *et al.* (2011), where the original Matlab implementation of MRBSiZer is presented. The main purpose of the sketch pad example is to show the concordance of the **R** implementation with its Matlab equivalent. Lastly, surface air temperature data from the Community Climate System Model 4.0 (Gent *et al.*, 2011) is analyzed to show how `mrbsizeR` can be used for measurements defined on a sphere.

As the vignette is a standalone document accompanying the **R**-package, it must be understandable independent of this thesis. An overlap of the content, especially with Chapter 2, is therefore inevitable.

`mrbsizeR`: Scale space multiresolution analysis in R

Thimo Schuster

2017-04-02

Contents

1 Introduction

`mrbsizeR` is an **R** package based on the MRBSiZer method by Holmström et al. (2011). The name is an abbreviation for **M**ulti**R**esolution **B**ayesian **SI**gnificant **ZE**ro crossings of derivatives in **R** and the method extends the portfolio of Bayesian SiZer methods for images and spatial fields, originally introduced by Erästö and Holmström (2005).

In the analysis of spatial fields or images (i.e. an object), scale space methods are often useful. When observed on different scales, distinct features of the object can be detected. Imagine the concept of a branch of a tree (Lindeberg, 1994): This concept makes sense on a scale from a few centimeters to a few meters. On a much smaller scale, one could describe the molecules that form the branch and on a much larger scale, it would be possible to describe the forest the tree grows in. The goal in scale space analysis is therefore to represent an object on different scales, and this is done by dividing it into a family of smooths. The smooths are made on different smoothing levels and each smooth provides information relevant at a particular scale, which makes it possible to extract scale-dependent features from the object.

Significant Zero Crossings of Derivatives (SiZer) is a method based on scale space ideas and has originally been developed for smooths of curves and time series. The goal is to find out whether a certain feature of the curve is "really there" or if it is just a sampling artifact. For curves and time series this is usually done by investigating the significance of the increases and decreases of the derivatives. Within the last years, this concept has been extended to various directions.

In contrast to usual scale space procedures, where a wide range of smooths is used, `mrbsizeR` employs differences of smooths at neighboring scales. This attempts to separate the features into distinct scale categories more aggressively. In a next step, the resulting differences are investigated with a Bayesian version of SiZer to see which of the features found are "really there" and which are only artifacts of sampling.

For data on a regular grid, one can summarize the analysis procedure in three steps:

1. Bayesian signal reconstruction
2. Forming of scale-dependent details using differences of smooths at neighboring scales
3. Posterior credibility analysis of the differences of smooths

For spherical data, no Bayesian signal reconstruction is implemented. The analysis procedure for this type of data therefore consists of the forming of scale-dependent details and the subsequent credibility analysis. The single steps and their application in `mrbsizeR` are explained by the following three examples. For further theory and algorithm details, see Holmström et al. (2011) and Schuster (2017). An extensive review of different statistical scale space methods including their applications is available in Holmström and Pasanen (2016).

2 Example: Data On A Regular Grid

For the first example, data from the North American Regional Climate Change Assessment Program (NARCCAP) is analyzed. NARCCAP is an international program producing climate change simulations for Canada, the United States and northern Mexico. The data used for this example is based on the MM5I regional model (Mearns et al., 2007, updated 2014) and is a simulation of the surface air temperature during summer 1995 (Mearns et al., 2007, updated 2014), see also http://www2.mmm.ucar.edu/mm5/ and http://www.narccap.ucar.edu/index.html. The simulation was carried out on a 120-by-98 regular grid, therefore 11'760 data points are available in total. The data set is not part of the `mrbsizeR` package.

```
# Structure of the dataset
str(tas.su.1995.MM5I)
```

```
## List of 3
##  $ lon: num [1:120, 1:98] 241 241 242 242 242 ...
##  $ lat: num [1:120, 1:98] 23 23.1 23.2 23.3 23.5 ...
##  $ su : num [1:120, 1:98] 21.8 21.7 21.7 21.7 21.7 ...
```

The variables `lon` and `lat` describe the longitude and latitude of each simulated surface air temperature in summer 1995 in degrees Celsius (`su`). The data covers the United States, the southern part of Canada and the northern part of Mexico.

As the output of the `mrbsizeR` analysis are plots on a rectangular grid, it makes sense to display Figure 1 also like this (compare Figure 2). By combining Figures 1 and 2, it is still possible to recognize all the important features such as coastlines, the Baja California or the Great Lakes.

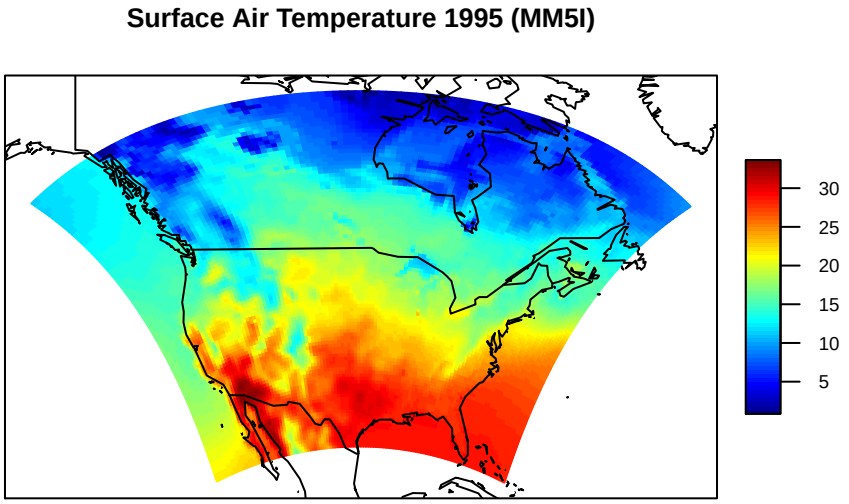

Figure 1: Simulated surface air temperature in summer 1995 for the United States, the southern part of Canada and the northern part of Mexico. The unit of the temperature is degrees Celsius.

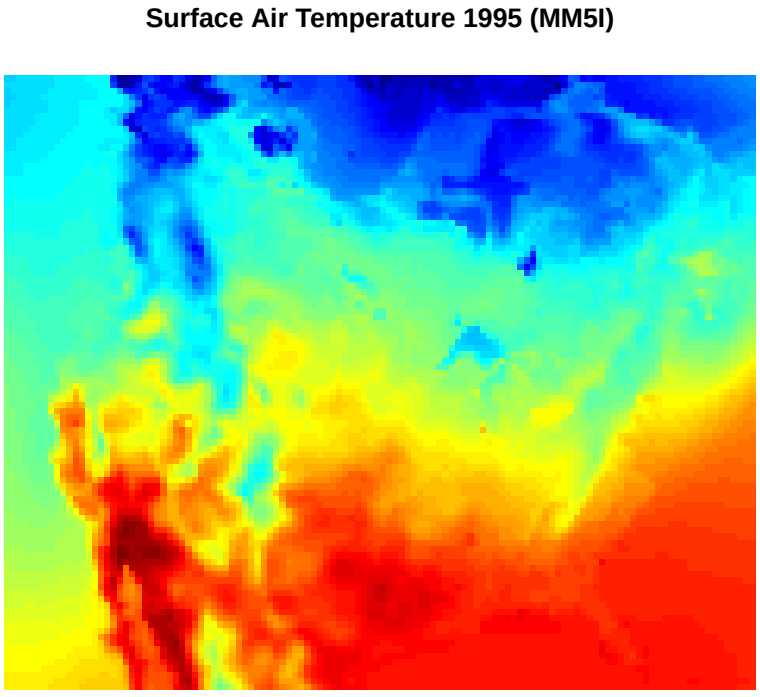

Figure 2: Simulated surface air temperature in summer 1995 for the United States, the southern part of Canada and the northern part of Mexico on a rectangular grid. Red describes warmer areas, colder areas are colored blue.

2.1 Bayesian Signal Reconstruction

The first step of the `mrbsizeR` analysis is the Bayesian signal reconstruction. The dataset is assumed to be a random signal which might be noisy. In order to account for this uncertainty in the data, a Bayesian model is used to reconstruct the original signal. The model used is

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\epsilon},$$

where $\boldsymbol{y}$ is the observed random signal (compare Figures 1 and 2), $\boldsymbol{x}$ is the unobserved underlying original signal and $\boldsymbol{\epsilon}$ is the noise. A $\mathcal{N}$-Inv-χ^2 prior distribution is assumed. In `mrbsizeR`, not the full posterior $p(\boldsymbol{x}, \sigma^2|\boldsymbol{y})$ is of interest, but the marginal posterior $p(\boldsymbol{x}|\boldsymbol{y})$. This marginal posterior follows a multivariate t-distribution t_ν and sampling from it results in samples from the reconstructed original signal $\boldsymbol{x}$.

Depending on how much is known about the noise in $\boldsymbol{y}$, the prior distribution parameters can be adjusted. For the example of the surface air temperature, the parameters $\lambda_0 = 0.2$, $\nu_0 = 15$ and $\sigma_0^2 = 36^2$ were used. This prior has little influence on the posterior as no information about possible noise on $\boldsymbol{y}$ is available. Using the function `rmvtDCT()`, samples from a t_ν-distribution can be generated. The sampling algorithm uses a discrete cosine transform (DCT) to speed up computations. For further information on the distributions and the sampling algorithm, see Holmström et al. (2011) and Schuster (2017).

```
# Sampling from a multivariate t-distribution
tas.post.samp <- rmvtDCT(object = tas.su.1995.MM5I$su,
                         lambda = 0.2, sigma = 36, nu0 = 15, ns = 1000)
```

1000 samples of the posterior distribution of the surface air temperature in Canada, Mexico and the United States were generated. These samples now form the reconstructed signal $\boldsymbol{x}$ and are used for further analysis.

2.2 Forming of scale-dependent detail components

Now that the original signal has been reconstructed, one can start forming scale-depending detail components. For being able to create differences of smooths at neighboring scales $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_L$, a set of appropriate smoothing levels $\lambda_1, \ldots, \lambda_L$ needs to be known. In contrast to other scale space methods, where a wide range of smoothing levels is used, `mrbsizeR` only requires a few of them. The goal is to separate the features of the object (here: the surface air temperature data) into scale-distinct categories. If too many smoothing levels are chosen, this will result in categories that do not feature relevant detail components. If, on the other hand, too few smoothing levels are chosen, detail components on different scales might mix up and are not recognizable anymore. It is therefore not only crucial to determine which smoothing levels are useful, but also the number of smoothing levels that should be used.

All methods proposed for the selection of smoothing levels have one thing in common: They offer a good starting point for finding useful smoothing levels - but to make sure that all detail components are captured optimally, user interaction is usually inevitable. Typically, a few iterations are necessary until satisfying smoothing levels are found.

The first method for the smoothing level selection depends on the smoother implemented in `mrbsizeR` and the dimension of the object analyzed only. By plotting so-called tapering functions of the eigenvalues of $\boldsymbol{Q}$, a precision matrix used in the smoother of `mrbsizeR`, and the smoothing levels λ_i, it is possible to determine which λ's could be useful. The eigenvalues and the smoothing parameters are related as follows:

- Small λ's involve large eigenvalues of $\boldsymbol{Q}$.
- Large λ's involve small eigenvalues of $\boldsymbol{Q}$.

The idea is to plot the tapering functions for different ranges of λ so that the functions are approximately disjoint. When using these λ-ranges for calculating the differences of smooths, this will result in orthogonal detail components. For detailed information about the smoother used and the properties of the tapering functions, see Holmström et al. (2011) and Schuster (2017).

The surface air temperature data was simulated on a 120-by-98 grid, and this information has to be passed to the plotting function `TaperingPlot()`. The vector `lambdaSmoother` contains the smoothing levels that should be used for drawing the tapering functions.

```
# Plot of signal-independent tapering functions
TaperingPlot(lambdaSmoother = c(1, 100, 10000), mm = 120, nn = 98)
```

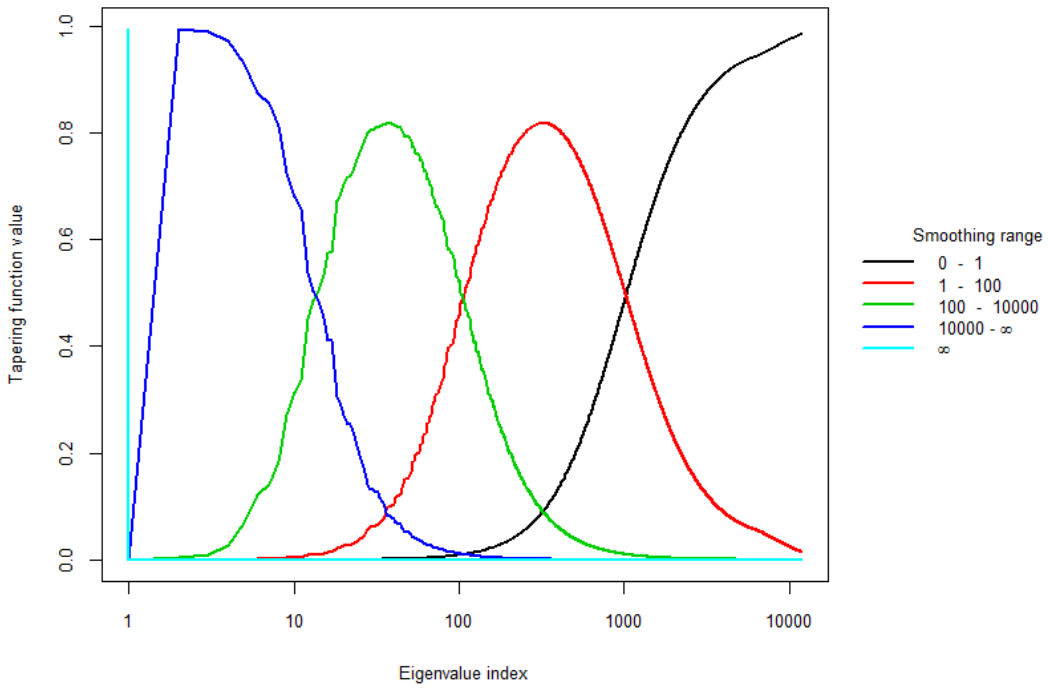

Figure 3: Signal-independent tapering functions for a 120-by-98 object with the smoothing level (λ) ranges 0–1, 1–100, 100–10000, 10000–∞ and ∞. The tapering functions are approximately disjoint.

The corresponding tapering functions are shown in Figure 3. The smoothing levels 0 and ∞ are added to the set of smoothing levels, as both of them have a special meaning. Whereas a smoother with $\lambda = 0$ is the so-called identity smoother ($\boldsymbol{S}_0\boldsymbol{x} = \boldsymbol{x}$), smoothing with $\lambda_L = \infty$ results in the global mean. The five

tapering functions in Figure 3 are approximately disjoint and a good starting point when trying to find useful smoothing levels.

An even better starting point can be attained if the underlying signal $\boldsymbol{x}$ is also taken into account. `TaperingPlot()` allows to draw signal-dependent tapering functions using the (optional) argument `Xmu`. As the original signal $\boldsymbol{x}$ is unknown, it is replaced by its posterior mean.

```
# Plot of signal-dependent tapering functions
TaperingPlot(lambdaSmoother = c(1, 100, 10000),
             mm = 120, nn = 98, Xmu = tas.post.samp$mu)
```

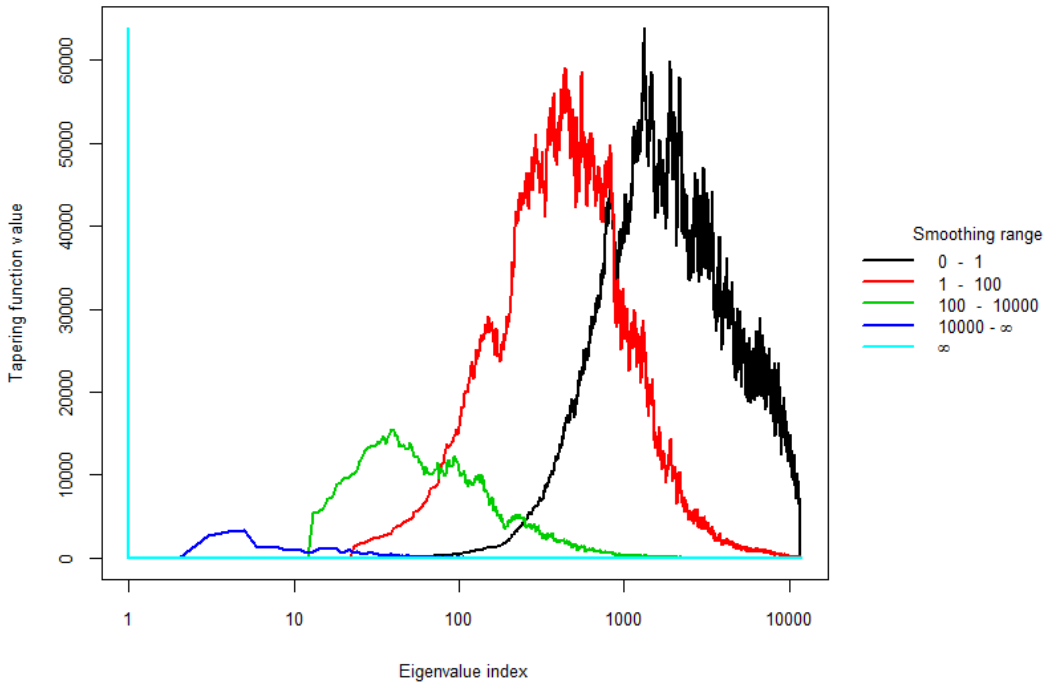

Figure 4: Moving average of the absolute values of signal-dependent tapering functions for the surface air temperature data with the smoothing level (λ) ranges 0–1, 1–100, 100–10000, 10000–∞ and ∞.

As signal-dependent tapering functions have values that vary wildly, visual inspection of these functions is difficult. `TaperingPlot()` therefore uses moving averages on the absolute values of these tapering functions to facilitate visual inspection. Figure 4 shows that the disjointedness of the smoothed tapering functions is not as pronounced as in Figure 3, but can still be observed.

A more formal approach is the numerical optimization of a suitable objective function with respect to the smoothing parameters. With `MinLambda()` it is possible to conduct this optimization for 2 or 3 λ's. The resulting smoothing levels are, in terms of disjointedness of the signal-dependent tapering functions, optimal. Still, it is often necessary to adjust the smoothing levels manually. For being able to extract also the smallest-scale details it can for instance be useful to include an additional, small λ. Furthermore, the number of "optimal" smoothing levels found with `MinLambda()` is limited. When optimizing over 3 λ's, one ends up with a sequence of 5 smoothing levels in total (3 optimized λ's, $\lambda_0 = 0$, $\lambda_L = \infty$). In cases where more

smoothing levels are necessary, i.e. where the object contains details relevant on more distinct scales, it is necessary to add smoothing levels manually. However, five smoothing levels turned out to be sufficient in many cases.

```
# Minimization of objective function with respect to the smoothing parameters
tas.min.lambda.out <- MinLambda(Xmu = tas.post.samp$mu, mm = 120, nn = 98,
                                nGrid = 45, nLambda = 3, sphere = FALSE,
                                lambda=10^seq(-12, 10, len = 45))
```

Most of the arguments of `MinLambda()` are already known from `TaperingPlot()`. In addition to this, it is necessary to specify `nGrid` (size of the grid the optimization should be carried out on, nGrid-by-nGrid), `nLambda` (either 2 or 3, number of λ's to be optimized) and `sphere` (logical; is the analysis on spherical data?). The optimal λ values evaluated for the surface air temperature data are 3.16e-11, 1e+02 and 1e+04.

```
# Minimal smoothing parameter values
tas.min.lambda.out$lambda[tas.min.lambda.out$minind]
```

```
## [1] 3.162278e-11 1.000000e+02 1.000000e+04
```

The minimization result can also be shown visually. For each optimized pair of λ's, a plot is drawn. Figure 5 shows the optimization for the surface air temperature example. The optimized λ's have the indices λ_2, λ_3 and λ_4, because $\lambda_1 = 0$ and $\lambda_5 = \infty$.

```
# Plot of the minimization result
plot(x = tas.min.lambda.out)
```

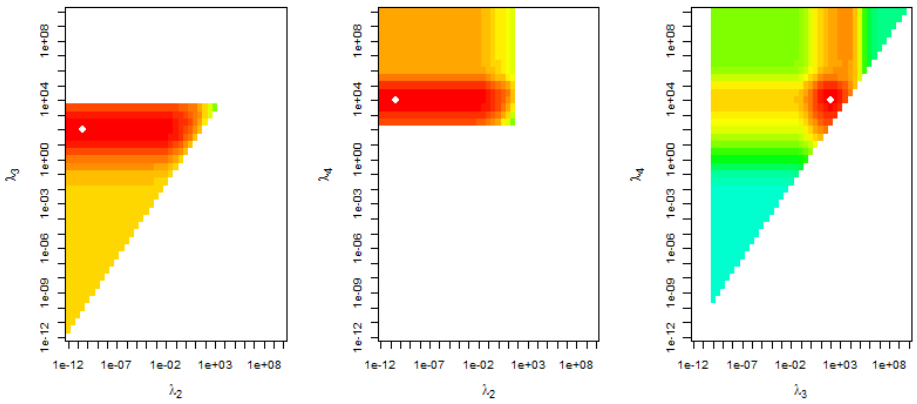

Figure 5: Minimization of the objective function for three smoothing levels λ. The minimum is indicated by a white point.

The smoothing levels found with the different methods can now be used as a starting point for finding scale-dependent details in the analyzed object. Usually, a few iterations with some smoothing level adjustments are necessary until the results found are satisfying. For the surface air temperature example, the smoothing level sequence $[\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5] = [0, 0.1, 90, 16'000, \infty]$ turned out to be useful. Using `mrbsizeRgrid()`, differences of smooths at neighboring scales are created. For the surface air temperature example, the samples generated with `rmvtDCT()` are used as input. If other samples are already available and no Bayesian signal reconstruction is needed, it is important to store them as a matrix where each column vector represents one sample.

```
# Creation of differences of smooths at neighboring scales
tas.mrb.out <- mrbsizeRgrid(posteriorFile = tas.post.samp$sample, mm = 120, nn = 98,
                            lambdaSmoother = c(0.1, 90, 15000), prob = 0.95)
```

The resulting object `tas.mrb.out` is a list containing three sublists. `smMean` contains the mean of each difference of smooths $\boldsymbol{z}_i$ over all `ns = 1000` samples from the posterior $p(\boldsymbol{x}|\boldsymbol{y})$. The lists `hpout` and `ciout` are relevant for the posterior credibility analysis, which is discussed in detail in the next subsection.

The `smMean`-plots for the surface air temperature example are displayed in Figure 6. With increasing smoothing level λ, the features are getting larger. The first component $\boldsymbol{z}_1$ shows the smallest scale details of $\boldsymbol{x}$. The contours of areas where a rapid change from warmer to colder temperature can be observed are clearly visible. In $\boldsymbol{z}_2$, larger-scale details are identifiable. The Great Lakes seem to be colder than the surrounding regions and the areas surrounding the Gulf of California are clearly warmer than the Gulf itself. $\boldsymbol{z}_3$ shows that the Gulf of California is the hottest region in Northern America. Furthermore, the Rocky Mountains are identifiable as a cool, longish band across the map. The next difference of smooths $\boldsymbol{z}_4$ shows a north-south temperature gradient, and $\boldsymbol{z}_5$ is the mean across the whole map.

```
# Posterior mean of the different detail components
plot(x = tas.mrb.out$smMean, color.pallet = fields::tim.colors(), turnOut = FALSE,
     aspRatio = 98/120)
```

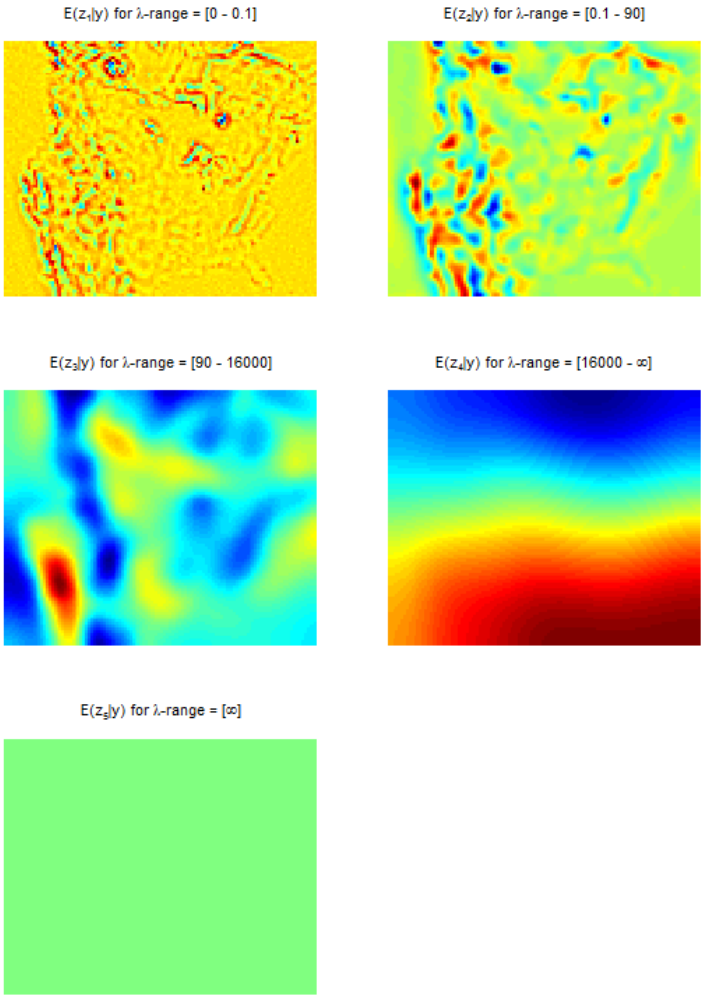

Figure 6: Decomposition of Figure 2 into differences of smooths at neighboring scales. The details $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_5$ are summarized by their posterior means. Warmer areas are colored red, colder areas are colored blue.

2.3 Posterior Credibility Analysis

As the detail components $\boldsymbol{z}_i$ are random samples from the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$, a credibility analysis has to be done. The goal is to infer which details are truly there and which are only artifacts of random variation. Three different methods for posterior credibility analysis are implemented in `mrbsizeR`:

- **Pointwise Maps (PW)**: Each location / pixel is tested separately for credibility.
- **Highest Pointwise Probability Maps (HPW)**: The inference is done jointly over all locations. In comparison to PW maps, this results in more non-credible areas. The advantage of HPW maps is that the credible areas are better connected than in PW maps. Small credibility islands with low expressiveness do appear less frequently and simplify the interpretation of the results.

- **Simultaneous Credible Intervals (CI)**: Inference is also done jointly over all locations. Here, simultaneous credible intervals centered on the posterior means are calculated. CI maps flag locations as credible more conservatively than PW or HPW maps.

For more detailed information about the three methods, see Holmström et al. (2011) and Schuster (2017). By default, credible regions of all three methods are calculated when executing `mrbsizeRgrid()`. The corresponding lists in the output of `mrbsizeRgrid()` are `hpout` and `ciout`.

```
# Plot of pointwise (PW) maps
plot(x = tas.mrb.out$hpout, plot_which = "PW", aspRatio = 98/120,
     color = c("dodgerblue3", "gainsboro", "firebrick1"), turnOut = FALSE)
```

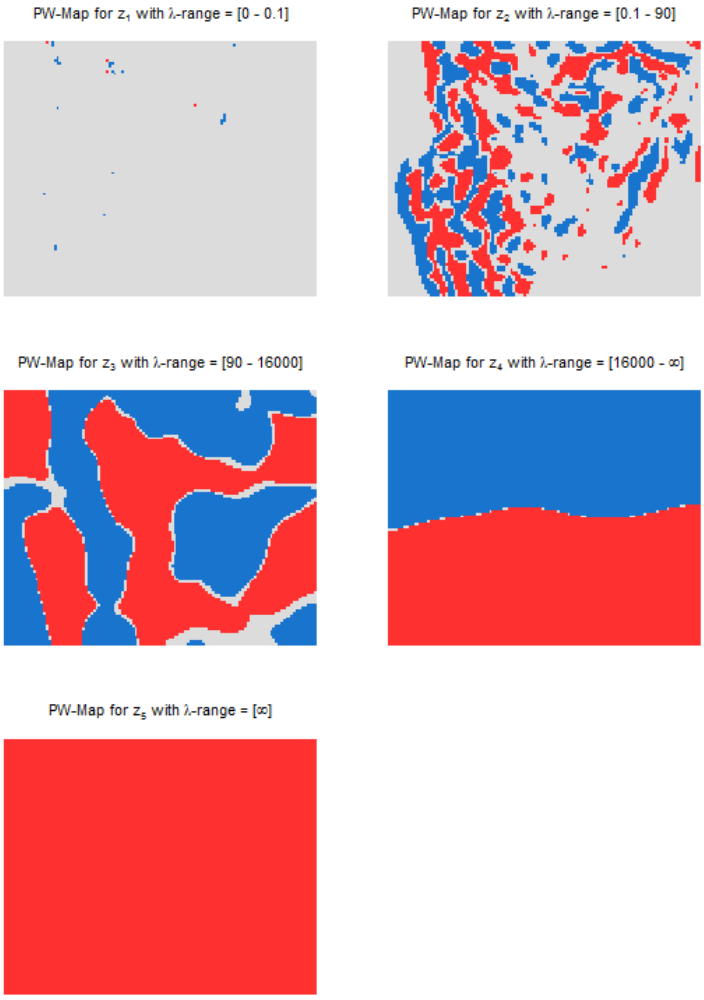

Figure 7: Pointwise (PW) maps for the surface air temperature example. Credibly warmer areas are colored red, credibly colder areas blue. Areas without credibility are gray.

The pointwise credibility maps in Figure 7 do show little credibility for the difference of smooths z_1. In z_2, the Gulf of California is credibly colder than Baja California and the Mexican mainland on the eastern side of the Gulf. In z_3, nearly everything is credible. This confirms that the Rocky Mountains are really colder and that the whole area of the Gulf of California is credibly warmer than surrounding regions. With z_4, it gets clear that the northern part of the map is credibly colder than the southern part. z_5 simply shows that the whole global mean is credible.

```
# Plot of highest pointwise probability (HPW) maps
plot(x = tas.mrb.out$hpout, plotWhich = "HPW", aspRatio = 98/120,
     color = c("dodgerblue3", "gainsboro", "firebrick1"), turnOut = FALSE)
```

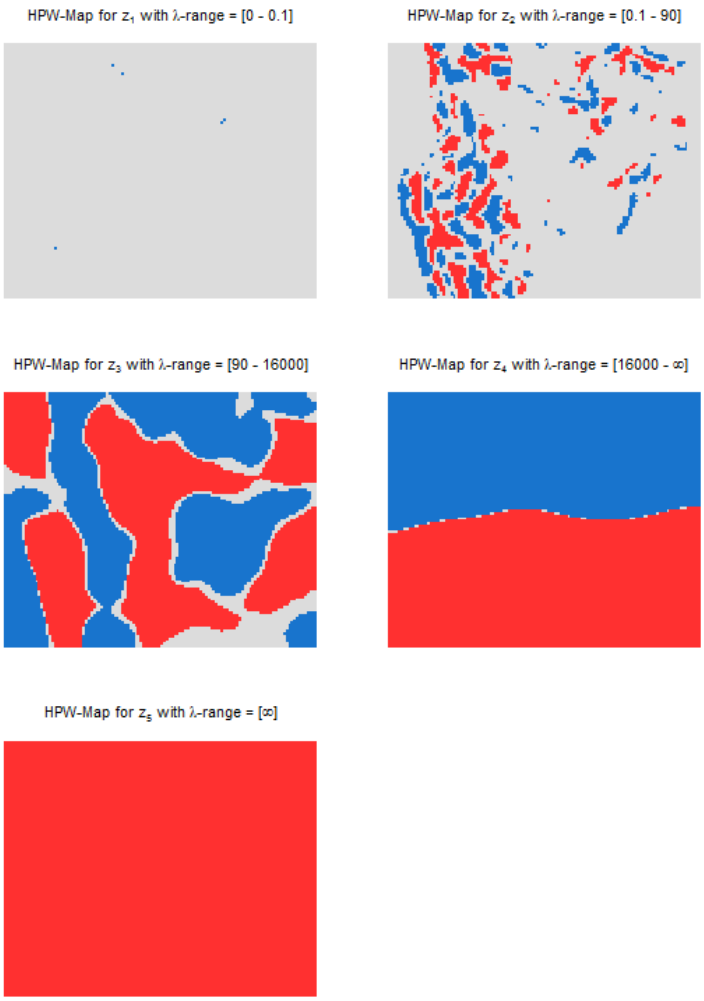

Figure 8: Highest pointwise probability (HPW) maps for the surface air temperature example. Credibly warmer areas are colored red, credibly colder areas blue. Areas without credibility are gray.

The HPW maps in Figure 8 exhibit less credibility than the PW maps, especially for small-scale details. However, the interpretation of the results stays the same. Due to the joint inference over all locations, small islands of credibility are less frequent.

```
# Plot of simultaneous credible interval (CI) maps
plot(x = tas.mrb.out$ciout, color = c("dodgerblue3", "gainsboro", "firebrick1"),
     turnOut = FALSE, aspRatio = 98/120)
```

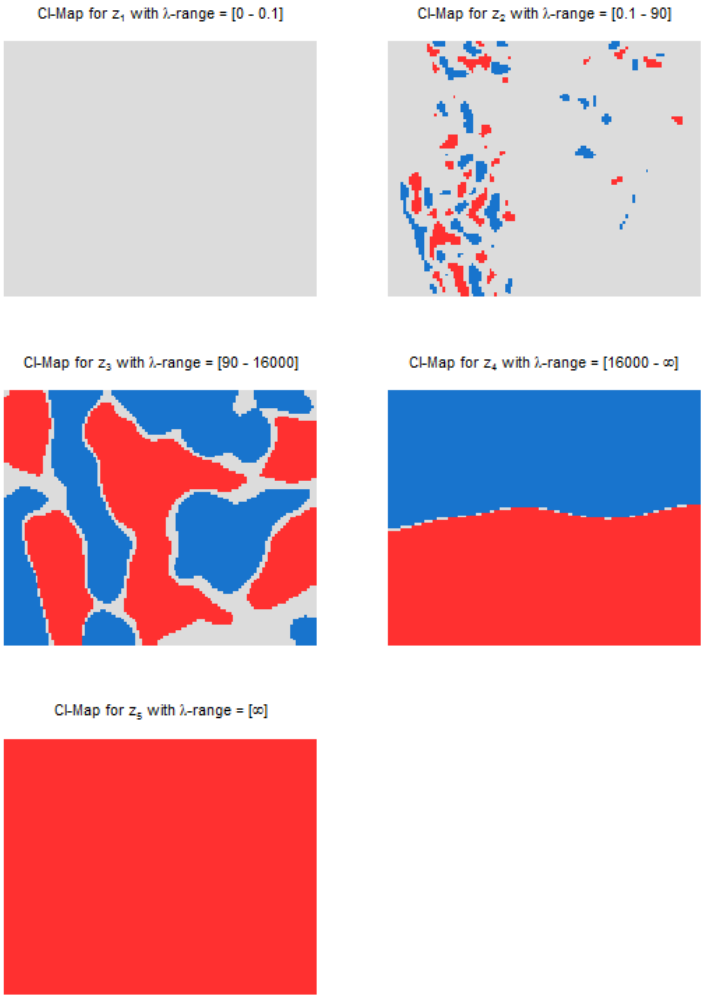

Figure 9: Simultaneous credible interval maps for the surface air temperature example. Credibly warmer areas are colored red, credibly colder areas blue. Areas without credibility are gray.

Detail component z_1 of the simultaneous credible interval maps in Figure 9 does not exhibit any credibility. CI maps are generally the most conservative credibility analysis method. It therefore is not surprising that in details z_2 and z_3 more gray areas can be observed than in the PW or HPW maps (compare Figures 7 and 8). Still, especially for the larger-scale details, the interpretation of the results does not change.

3 Example: Comparison to Matlab software

The `mrbsizeR` methodology was first implemented in the Matlab program MRBSiZer (Holmström and Pasanen, 2011), which is available at http://cc.oulu.fi/~lpasanen/MRBSiZer/. In order to ensure that the results obtained with R and Matlab are concordant, the sketch pad example from the original paper (Holmström et al., 2011) is reconstructed (compare Figure 10. The digital image is of the size 284-by-400 and the prior parameters λ_0, σ_0^2 and ν_0 had the values $0.2, 8.9^2$ and 10, respectively. The set of smoothing levels used in the multiresolution analysis was $[0, 1, 30, 6 \times 10^5, \infty]$. 3000 samples of the posterior $p(\boldsymbol{x}|\boldsymbol{y})$ were generated.

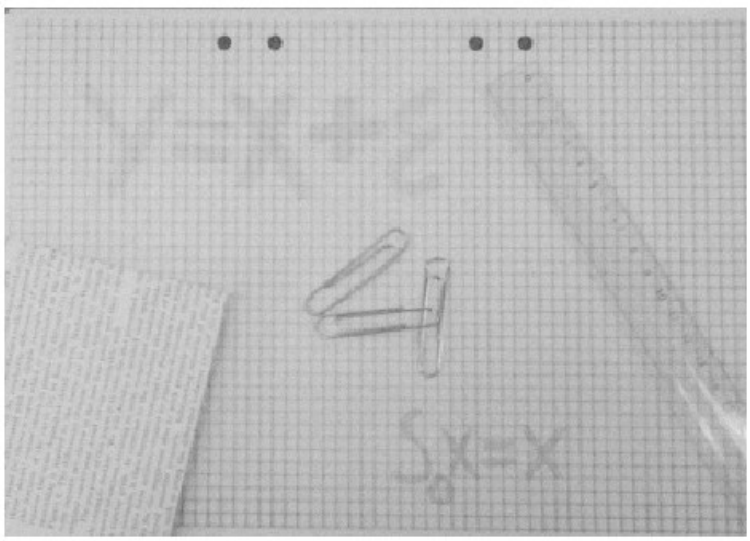

Figure 10: Original sketch pad image.

The detail components $\boldsymbol{z}_i$ are shown in Figure 11. As the output of `mrbsizeRgrid()` is based on random samples, small differences between two executions are inevitable. When compared to the Matlab figures in Holmström et al. (2011), no big differences can be detected. More differences can be found in the HPW maps, especially in detail component $\boldsymbol{z}_3$ (compare Figure 12). It seems that in the **R** implementation, a slightly larger part of the component was flagged as credible. Nevertheless, the detail components look very similar to their Matlab pendants and the interpretation of the results stays the same. The differences can be explained by random sampling and one can be confident that concordance across the systems holds.

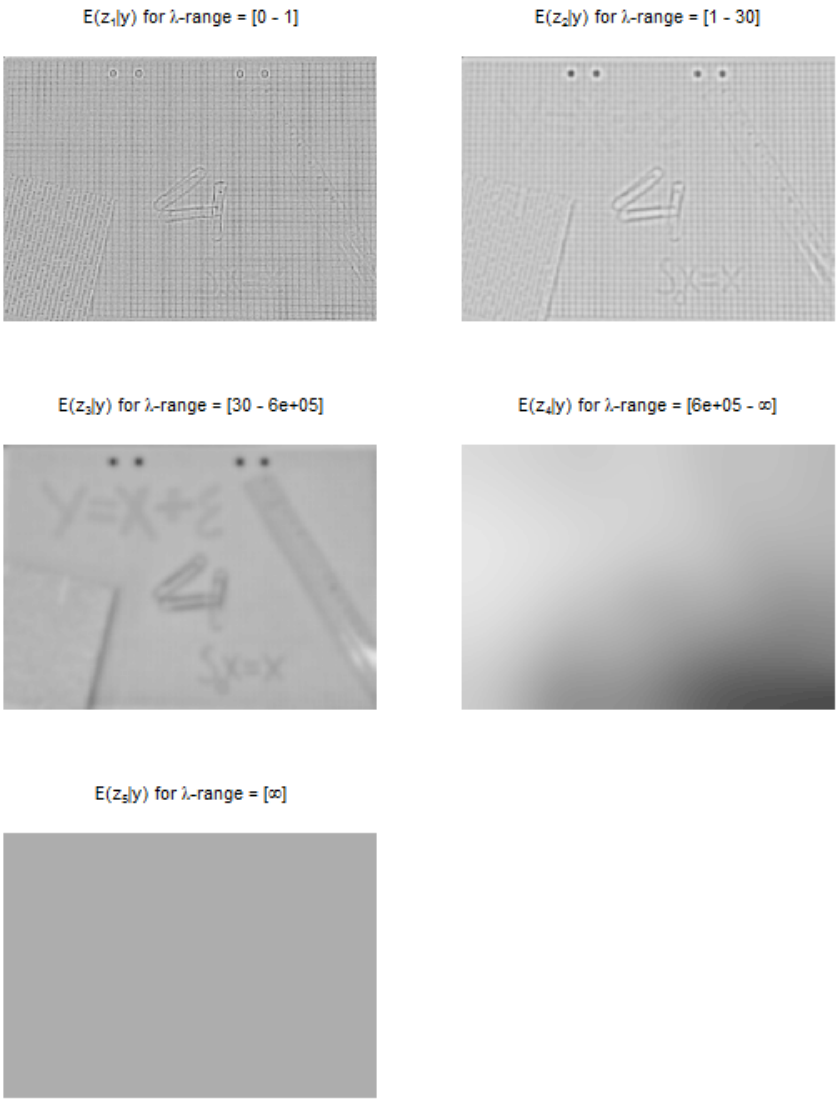

Figure 11: Posterior mean of detail components $\boldsymbol{z}_i$ from the sketch pad example.

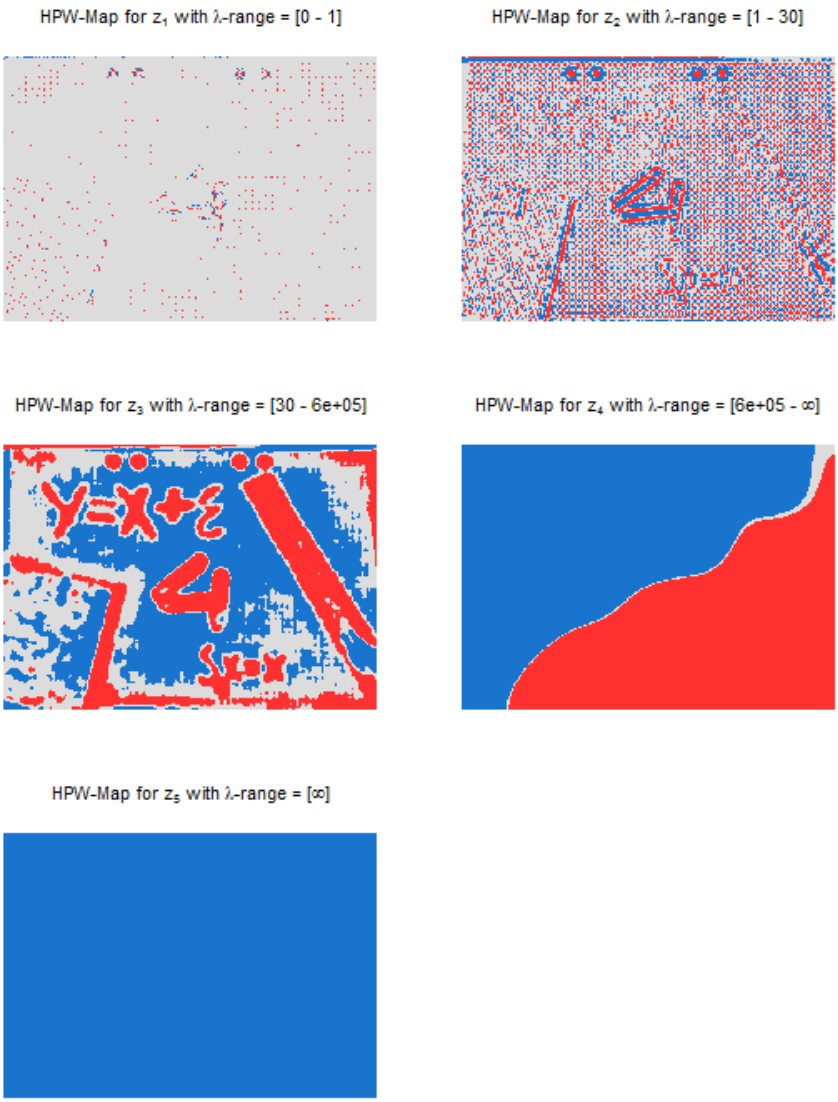

Figure 12: Highest pointwise probabilities maps of detail components $\boldsymbol{z}_i$ from the sketch pad example. Credibly darker areas are red, credibly brighter areas are blue.

4 Example: Data On A Sphere

The third example in this vignette demonstrates how `mrbsizeR` can be used to analyze spherical data. In contrast to the analysis of data on a grid, no Bayesian signal reconstruction is implemented. To form scale-dependent details, data samples need to be available beforehand. The analysis procedure for spherical data can therefore be summarized in two steps:

1. Forming of scale-dependent details using differences of smooths at neighboring scales
2. Posterior credibility analysis of the differences of smooths

Data from the Community Climate System Model 4.0 (CCSM4, see http://www.cesm.ucar.edu/models/ccsm4.0/ccsm/) is used to illustrate `mrbsizeR` on spherical data. CCSM4 is a climate model simulating the earth's climate system, see Gent et al. (2011). For this analysis, the simulated surface air temperature in June of the years 1870–2100 was considered. Instead of using the surface air temperature itself, its deviation to the yearly mean has been used. This detrends the data and makes the simulations of 231 consecutive years comparable. The resulting data set consisting of 231 observations is then used as samples for the `mrbsizeR` analysis. The data is not part of the `mrbsizeR` package.

Figure 13 summarizes the samples by their mean. It is clearly visible that the temperature is higher in areas around the equator and gets lower closer to the Polar Regions. The unit of the surface air temperature deviation is degrees Kelvin.

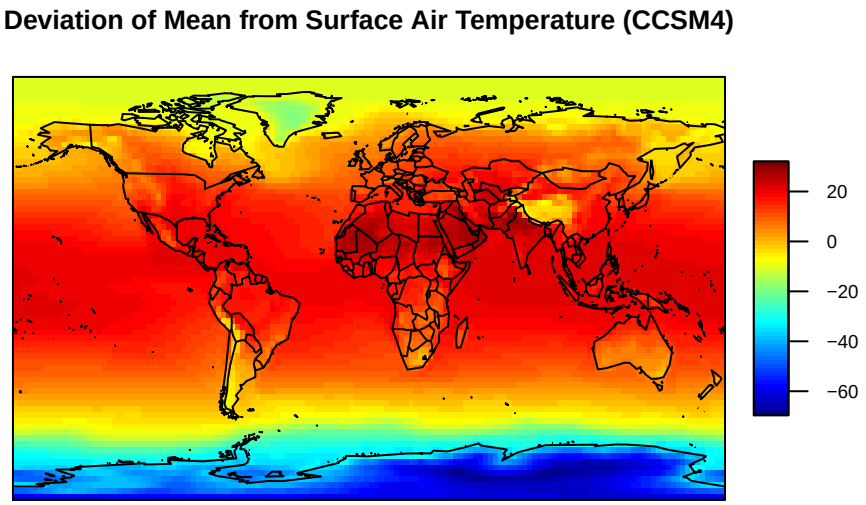

Figure 13: Deviation from mean of simulated air temperature measurements (CCSM4) for the years 1870–2100 in degrees Kelvin. The deviations are summarized by their mean.

For conducting scale space multiresolution analysis with `mrbsizeR`, useful smoothing parameters need to be evaluated first. As explained in the NARCCAP data example, `MinLambda()` offers the possibility to find useful smoothing parameters numerically. The function call for spherical data is nearly identical to the call for non-spherical data, the only difference is the argument `sphere` which has to be `TRUE`.

```
# Minimization of objective function with respect to the smoothing parameters
# for spherical data
spherical.min.lambda.out <- MinLambda(Xmu = dat.ccsm4.mu, mm = 144, nn = 72,
                                      nGrid = 35, nLambda = 2, sphere = TRUE)
```

Once useful smoothing levels have been selected, differences of smooths at neighboring scales can be created using the function `mrbsizeRsphere()`.

```
# Creation of differences of smooths at neighboring scales for spherical data
spherical.mrb.out <- mrbsizeRsphere(posteriorFile = dat.ccsm4, mm = 144, nn = 72,
                                    prob = 0.95, lambdaSmoother = c(0.0026))
```

For creating the differences of smooths at neighboring scales in Figure 14, the smoothing level sequence $[\lambda_1, \ldots, \lambda_3] = [0, 0.0026, \infty]$ was used. Only one smoothing level was added to the default sequence $[0, \infty]$. This is enough to capture features at all different scales. Whereas $\boldsymbol{z}_1$ shows small-scale details like colder regions in Tibet or Chile, $\boldsymbol{z}_2$ reveals a large red-colored area around the equator and large blue-colored areas at the Polar Regions. $\boldsymbol{z}_3$ shows the global mean.

```
# Posterior mean of the different detail components for spherical data
plot(x = spherical.mrb.out$smMean, lon = dat.ccsm4$lon, lat = dat.ccsm4$lat,
     color.pallet = fields::tim.colors())
```

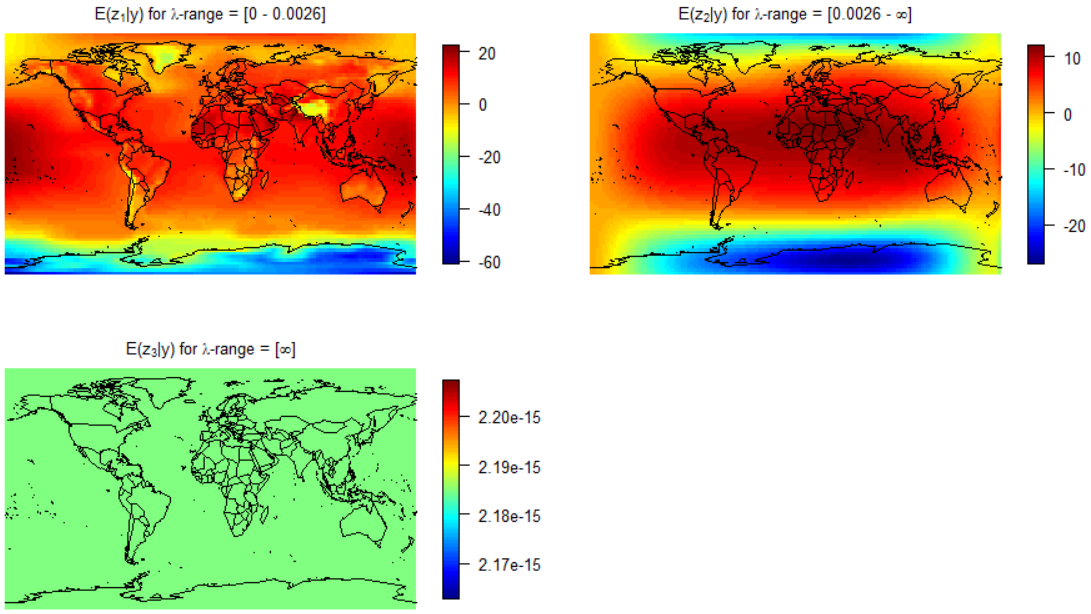

Figure 14: Decomposition of Figure 13 into differences of smooths at neighboring scales. The details $z_1, \ldots, z_3$ are summarized by their posterior means. Areas with larger data values are colored red, areas with smaller values are colored blue.

The posterior credibility analysis of detail component z_1 using highest pointwise probability (HPW) maps (compare Figure 15) reveals that regions like Chile or the eastern part of South Africa are credibly colder than surrounding areas. The large-scale components in z_2 are mostly credible and hence "really there". The global mean in z_3 is not credible in this example. The reason is the data used: Instead of considering the yearly surface air temperature, its deviations to the yearly mean are considered. z_3 is therefore not the average surface air temperature, but the average mean deviation, which always equals 0.

```
# Plot of highest pointwise probability (HPW) maps for spherical data
plot(x = spherical.mrb.out$hpout, lon = dat.ccsm4$lon, lat = dat.ccsm4$lat,
     plotWhich = "HPW", color = c("dodgerblue3", "gainsboro", "firebrick1"))
```

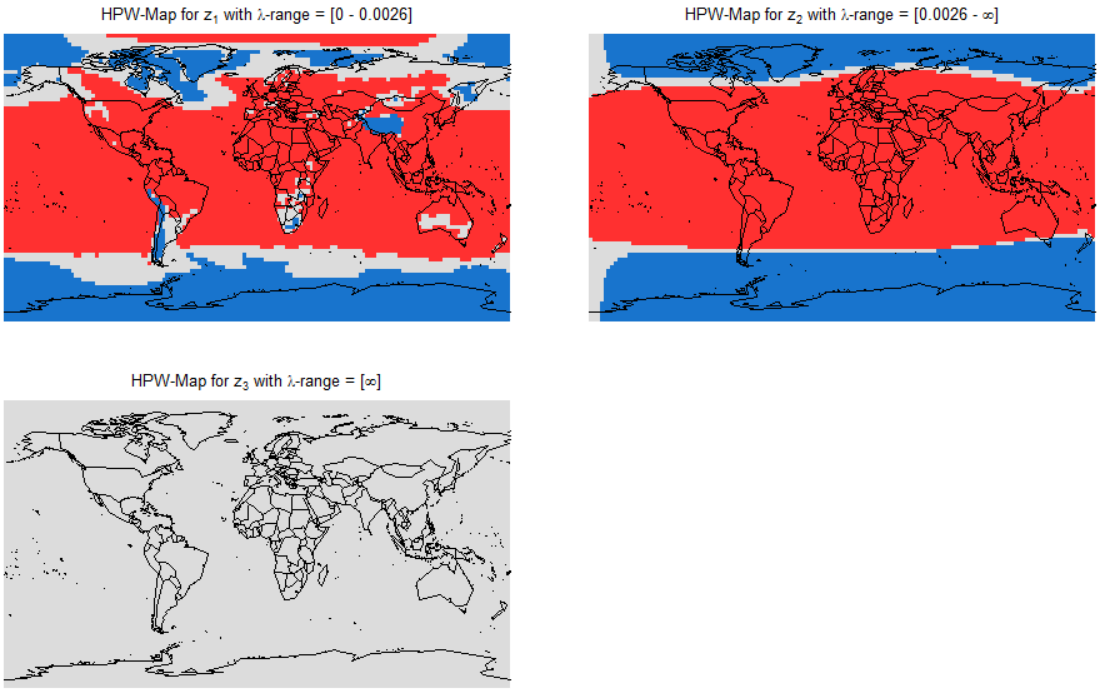

Figure 15: Highest pointwise probability maps. Areas with credibly larger data values are colored red, areas with credibly smaller values are blue. Areas without credibility are gray.

5 What If Not Enough Computing Power Is Available?

Especially for analyses with a large analysis object and/or many samples, an `mrbsizeR` analysis is resource-intensive. For cases where due to computational reasons not enough samples can be generated, the additional argument `smoothOut` has been added to `mrbsizeRgrid()` and `mrbsizeRsphere()`. If `smoothOut = TRUE`, the output list will also contain a sublist `smoothSamples`, which includes the differences of smooths for all samples. This makes it possible to manually increase the number of samples and get HPW maps and CI maps with a higher confidence. An example is provided in the following code chunk.

```
# Generate samples from posterior distribution
tas.post.samp <- rmvtDCT(object = tas.su.1995.MM5I$su,
                         lambda = 0.2, sigma = 36, nu0 = 15, ns = 1000)

# Do mrbsizeR analysis and output the differences of smooths for all samples
tas.mrb.out.1 <- mrbsizeRgrid(posteriorFile = tas.post.samp$sample, mm = 120,
                              nn = 98, lambdaSmoother = c(0.1, 90, 15000),
                              prob = 0.95, smoothOut = TRUE)

# Do the same procedure again
tas.post.samp <- rmvtDCT(object = tas.su.1995.MM5I$su,
                         lambda = 0.2, sigma = 36, nu0 = 15, ns = 1000)

tas.mrb.out.2 <- mrbsizeRgrid(posteriorFile = tas.post.samp$sample, mm = 120,
                              nn = 98, lambdaSmoother = c(0.1, 90, 15000),
                              prob = 0.95, smoothOut = TRUE)

# Combine all differences-of-smooths-samples and call CImap manually
smoothSamples <- list(); ciout <- list()

for(i in 1:length(tas.mrb.out.1$smoothSamples)) {
  smoothSamples <- cbind(tas.mrb.out.1$smoothSamples[[i]],
                         tas.mrb.out.2$smoothSamples[[i]])
  ciout[[i]] <- CImap(smoothVec = smoothSamples, mm = 120, nn = 98, prob = 0.95)
}

# Set the class correctly for visualizing the output.
# Titles need to be defined in this case!
# Class name CI maps: "CImapGrid" or "CImapSphere"
# Class name PW / HPW maps: "HPWmapGrid" or "HPWmapSphere"
class(ciout) <- "CImapGrid"
plot(ciout, title = c("Diff_1", "Diff_2", "Diff_3", "Diff_4", "Diff_5"))
```

6 Data Acknowledgments

We wish to thank the North American Regional Climate Change Assessment Program (NARCCAP) for providing the data used in this paper. NARCCAP is funded by the National Science Foundation (NSF), the U.S. Department of Energy (DoE), the National Oceanic and Atmospheric Administration (NOAA), and the U.S. Environmental Protection Agency Office of Research and Development (EPA).

We acknowledge the World Climate Research Program's Working Group on Coupled Modelling, which is responsible for CMIP, and we thank the climate modeling groups for producing and making available their model output. For CMIP the U.S. Department of Energy's Program for Climate Model Diagnosis and Intercomparison provides coordinating support and led development of software infrastructure in partnership with the Global Organization for Earth System Science Portals.

References

Panu Erästö and Lasse Holmström. Bayesian multiscale smoothing for making inferences about features in scatterplots. *Journal of Computational and Graphical Statistics*, 14(3):569–589, 2005.

Peter R. Gent, Gokhan Danabasoglu, Leo J. Donner, Marika M. Holland, Elizabeth C. Hunke, Steve R. Jayne, David M. Lawrence, Richard B. Neale, Philip J. Rasch, Mariana Vertenstein, Patrick H. Worley, Zong-Liang Yang, and Minghua Zhang. The community climate system model version 4. *Journal of Climate*, 24:4973–4991, 2011.

Lasse Holmström and Leena Pasanen. MRBSiZer. http://cc.oulu.fi/∼lpasanen/MRBSiZer/, 2011. Accessed: 2017-03-04.

Lasse Holmström and Leena Pasanen. Statistical scale space methods. *International Statistical Review*, 2016. ISSN 1751-5823. http://dx.doi.org/10.1111/insr.12155.

Lasse Holmström, Leena Pasanen, Reinhard Furrer, and Stephan R. Sain. Scale space multiresolution analysis of random signals. *Computational Statistics and Data Analysis*, 55:2840–2855, 2011.

Tony Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2):225–270, 1994.

Linda Mearns, Seth McGinnis, Raymond Arritt, Sebastien Biner, Phillip Duffy, William Gutowski, Isaac Held, Richard Jones, Ruby Leung, Ana Nunes, Mark Snyder, Daniel Caya, James Correia, David Flory, Daryl Herzmann, René Laprise, Wilfran Moufouma-Okia, Gene Takle, Haiyan Teng, Josh Thompson, Simon Tucker, Bruce Wyman, Anila Anitha, Lawrence Buja, Christopher Macintosh, Larry McDaniel, Travis O'Brien, Yun Qian, Lisa Sloan, Gary Strand, and Casey Zoellick. The north american regional climate change assessment program dataset. National Center for Atmospheric Research Earth System Grid data portal, Boulder, CO, 2007, updated 2014.

Thimo Schuster. mrbsizeR: Scale space multiresolution analysis in R. Master's thesis, University of Zurich, 2017.

4. Discussion

`mrbsizeR` is an **R**-package that implements the MRBSiZer method by Holmström *et al.* (2011) in **R**. It offers all the functionalities of the original Matlab implementation plus some extensions. A comparison between the Matlab and the **R** implementation shows that concordance is given (Chapter 3). The scale space multiresolution analysis method can be summarized in three steps. The first step is a Bayesian signal synthesis to reconstruct the original signal out of a noisy observation. In a second step, differences of smooths at neighboring scales are used to form scale-dependent signal components. The posterior credibility analysis of the features found in the scale-dependent signal components is the third step. The analysis is not only possible for data defined on a regular grid but also for spherical data. An important feature of `mrbsizeR` are the methods for identifying useful smoothing levels. Without useful smoothing levels, no meaningful differences of smooths can be created. Methods for signal-independent and signal-dependent smoothing level selection are implemented in `mrbsizeR`. All these methods deliver useful smoothing levels, but for the best results, user fine tuning is usually inevitable.

Holmström *et al.* (2011) made a comparison of MRBSiZer with wavelets and showed that the two methods are competitive. Advantages of MRBSiZer over comparable methods are the flexibility in statistical modeling (other prior distributions could easily be implemented) and the visualization methods for inferences on image differences (Holmström and Pasanen, 2012), which have a simple but comprehensive interpretation.

4.1. Limitations

In reality, it is not always possible to collect data on a regular grid. Measuring the population of all districts of the Canton of Zurich will for example result in an irregular grid where the distance between a location and its neighbors is not fixed and varies depending on the size and the shape of the districts. `mrbsizeR` has its focus on data defined on a regular grid and it is not possible to analyze unstructured grids. If the analyzed object is defined on a regular grid but features missing values, small-scale details can get lost in the analysis. Nevertheless, using mean imputation, it is possible to recover the most important features. Experiments to the analysis of data featuring missing values can be found in Appendix A.3.

Another restriction of `mrbsizeR` is the limited amount of smoothing parameters λ that can be optimized when taking the underlying signal x into account. Although it would theoretically be possible to optimize over more than three λ's, this gets computationally expensive due to the increasing number of dimensions. However, combined with $\lambda_0 = 0$ and $\lambda_L = \infty$, `mrbsizeR` allows for an optimized smoothing parameter sequence of 5 λ's. It is generally recommended to keep the number of smoothing levels small, and 5 λ's turned out to be enough for many applications. When using too many λ's, some detail components z_i will describe similar structures in the object and other meaningful structures might be completely missed.

4.2. Outlook

The MRBSiZer method has successfully been implement in the **R**-package `mrbsizeR`. Even though some extensions of the original MRBSiZer Matlab implementation and further investigations on the **R** program have been made, several other ideas could be further investigated.

Especially the sampling of several hundreds or thousands of t_ν-distribution-samples and the subsequent analysis of the differences of smooths (Sections 2.3–2.5) is resource-intensive. On the author's computer (Section 1.3), the `mrbsizeR` analysis of the NARCCAP data example in the vignette (Chapter 3) with the function `mrbsizeRgrid()` using `ns = 1000` samples took 2.5 minutes. A **C** implementation of computational intensive parts could improve the performance and reduce the execution time. Especially with large analysis objects, it is not always possible to generate the desired amount of samples due to RAM limitations. Even though the vignette presents a workaround for this issue, an automated solution for memory-intensive analyses would be an improvement of `mrbsizeR`.

Further, some research concerning an optimal choice of the smoothing levels λ_i could be made. All methods implemented in `mrbsizeR` require some user interaction for receiving the best results. This fine tuning can be time-consuming and maybe other, more autonomous approaches exist. Moreover, the numerical optimization of useful smoothing levels (see equation (2.8)) could be implemented in a more efficient way. At the moment, the optimization is carried out on the whole grid that is passed to the optimization algorithm. As the objective function is convex, this could be implemented more efficiently. If the optimization algorithm would stop when the objective function values start increasing, minimization would be less resource intensive and faster.

Bibliography

Chaudhuri, P. and Marron, J. (1999). Sizer for exploration of structures in curves. *Journal of the American Statistical Association*, **94**, 807–823. 4

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. 4, 9, 10, 36

Furrer, R. (2016). Modeling dependent data: An excursion. Script for UZH STA330.

Gent, P. R., Danabasoglu, G., Donner, L. J., Holland, M. M., Hunke, E. C., Jayne, S. R., Lawrence, D. M., Neale, R. B., Rasch, P. J., Vertenstein, M., Worley, P. H., Yang, Z.-L., and Zhang, M. (2011). The community climate system model version 4. *Journal of Climate*, **24**, 4973–4991. 11

Heersink, D. (2013). *On the modeling and analysis of sequential observations of spatial processes with application to modern earthwork compaction.* PhD thesis, Universität Zürich. 2

Holmström, L. and Pasanen, L. (2011). MRBSiZer. http://cc.oulu.fi/∼lpasanen/MRBSiZer/. Accessed: 2017-03-04.

Holmström, L. and Pasanen, L. (2012). Bayesian scale space analysis of differences in images. *Technometrics*, **54**, 16–29. 4, 33

Holmström, L. and Pasanen, L. (2016). Statistical scale space methods. *International Statistical Review*. http://dx.doi.org/10.1111/insr.12155. 4

Holmström, L., Pasanen, L., Furrer, R., and Sain, S. R. (2011). Scale space multiresolution analysis of random signals. *Computational Statistics and Data Analysis*, **55**, 2840–2855. 2, 4, 5, 7, 8, 9, 10, 11, 33, 39

Lindeberg, T. (1994). Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, **21**, 225–270. 4

Mearns, L., McGinnis, S., Arritt, R., Biner, S., Duffy, P., Gutowski, W., Held, I., Jones, R., Leung, R., Nunes, A., Snyder, M., Caya, D., Correia, J., Flory, D., Herzmann, D., Laprise, R., Moufouma-Okia, W., Takle, G., Teng, H., Thompson, J., Tucker, S., Wyman, B., Anitha, A., Buja, L., Macintosh, C., McDaniel, L., O'Brien, T., Qian, Y., Sloan, L., Strand, G., and Zoellick, C. (2007, updated 2014). The north american regional climate change assessment program dataset. National Center for Atmospheric Research Earth System Grid data portal, Boulder, CO. 2, 11

Pasanen, L., Launonen, I., and Holmström, L. (2013). A scale space multiresolution method for extraction of time series features. *Stat*, **2**, 273–291. 4, 10

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

Strang, G. (1999). The discrete cosine transform. *Society for Industrial and Applied Mathematics*, **41**, 135–147.

A. Appendix

A.1. Derivation of Marginal Posterior Density $p(\boldsymbol{x}|\boldsymbol{y})$

Appendix A.1 provides a detailed explanation on how the marginal posterior denisty $p(\boldsymbol{x}|\boldsymbol{y})$ introduced in Section 2.2 is derived, see also Erästö and Holmström (2005).

For σ^2, an Inv-$\chi^2(\nu_0, \sigma_0^2)$ prior was used. The prior of the (not observed) object $\boldsymbol{x}$ is of the form

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

and the parameters ν_0, σ_0^2 and λ_0 can be used to reflect prior knowledge about σ^2 and $\boldsymbol{x}$. Combining the two priors results in a $\mathcal{N}$-Inv-χ^2 prior distribution of the form

$$p(\boldsymbol{x}, \sigma^2) \propto \left(\frac{\lambda_0}{\sigma^2}\right)^{(n-1)/2} \exp\left(-\frac{\lambda_0}{2\sigma^2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x}\right) \frac{1}{\Gamma\left(\frac{\nu_0}{2}\right)} \left(\frac{\nu_0\sigma_0^2}{2}\right)^{\frac{\nu_0}{2}} (\sigma^2)^{-\left(\frac{\nu_0}{2}+1\right)} \exp\left(-\frac{\nu_0\sigma_0^2}{2\sigma^2}\right)$$

which, when the hyperparameters are neglected, simplifies to

$$p(\boldsymbol{x}, \sigma^2) \propto \left(\frac{\lambda_0}{\sigma^2}\right)^{(n-1)/2} (\sigma^2)^{-\left(\frac{\nu_0}{2}+1\right)} \exp\left(-\frac{\lambda_0}{2\sigma^2}\left(\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} - \frac{\nu_0\sigma_0^2}{\lambda_0}\right)\right). \tag{A.1}$$

According to the Bayesian model used in `mrbsizeR` (Section 2.2), it is assumed that $\boldsymbol{y}|\boldsymbol{x}, \sigma^2 \sim \mathcal{N}(\boldsymbol{x}, \boldsymbol{I}\sigma^2)$, where $\boldsymbol{I}$ is the $n \times n$ identity matrix. The likelihood of the observed object $\boldsymbol{y}$ is therefore given by

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{x}, \sigma^2) &\propto \prod_{i=1}^{n} \sigma^{-1} \exp\left(-\frac{1}{2\sigma^2}(y_i - x_i)^2\right) \\ &\propto \sigma^{-n} \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{y} - \boldsymbol{x}\|^2\right). \end{aligned} \tag{A.2}$$

For the following steps, λ_0 is assumed to be fixed. With (A.1) and (A.2), it is possible to write down the full posterior density, which then simplifies to

$$\begin{aligned} p(\boldsymbol{x}, \sigma^2|\boldsymbol{y}) &\propto p(\sigma^2)p(\boldsymbol{x}|\sigma^2)p(\boldsymbol{y}|\boldsymbol{x}, \sigma^2) \\ &\propto (\sigma^2)^{-\left(\frac{2n+\nu_0+1}{2}\right)} \exp\left(-\frac{1}{2\sigma^2}(\lambda_0\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \|\boldsymbol{y} - \boldsymbol{x}\|^2 + \nu_0\sigma 0^2)\right). \end{aligned} \tag{A.3}$$

For finding the marginal posterior density $p(\boldsymbol{x}|\boldsymbol{y})$, σ^2 needs to be integrated out of (A.3). According to Erästö and Holmström (2005), the resulting marginal posterior is

$$p(\boldsymbol{x}|\boldsymbol{y}) \propto \left(\lambda_0\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \|\boldsymbol{y} - \boldsymbol{x}\|^2 + \nu_0\sigma_0^2\right)^{-\left(\frac{2n+\nu_0-1}{2}\right)}$$

and follows a multivariate t-distribution $t_\nu(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. For being able to sample from this distribution, the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ have to be known. They can be found by completing the square for $(\lambda_0 \boldsymbol{x}^T \boldsymbol{Q}\boldsymbol{x} + \|\boldsymbol{y} - \boldsymbol{x}\|^2 + \nu_0\sigma_0^2)$. With $S_{\lambda_0} = (\boldsymbol{I} + \lambda_0 \boldsymbol{Q})^{-1} \; \forall \; (\lambda_0 \geq 0)$,

$$
\begin{aligned}
&\lambda_0 \boldsymbol{x}^T \boldsymbol{Q}\boldsymbol{x} + \|\boldsymbol{y} - \boldsymbol{x}\|^2 + \nu_0\sigma_0^2 \\
&= \lambda_0 \boldsymbol{x}^T \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{x}^T\boldsymbol{y} + \boldsymbol{x}^T\boldsymbol{x} + \nu_0\sigma_0^2 \\
&= \boldsymbol{x}^T(\boldsymbol{I} + \lambda_0 \boldsymbol{Q})\boldsymbol{x} + \boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{x}^T\boldsymbol{y} + \nu_0\sigma_0^2 \\
&= \boldsymbol{x}^T \boldsymbol{S}_{\lambda_0}^{-1}\boldsymbol{x} + \boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{x}^T\boldsymbol{y} + \nu_0\sigma_0^2 \\
&= \boldsymbol{x}^T \boldsymbol{S}_{\lambda_0}^{-1}\boldsymbol{x} - 2\boldsymbol{x}^T \boldsymbol{S}_{\lambda_0}^{-1}\boldsymbol{S}_{\lambda_0}\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{S}_{\lambda_0}\boldsymbol{S}_{\lambda_0}^{-1}\boldsymbol{S}_{\lambda_0}\boldsymbol{y} - \boldsymbol{y}^T\boldsymbol{S}_{\lambda_0}\boldsymbol{S}_{\lambda_0}^{-1}\boldsymbol{S}_{\lambda_0}\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{y} + \nu_0\sigma_0^2 \\
&= (\boldsymbol{x} - \boldsymbol{S}_{\lambda_0}\boldsymbol{y})^T \boldsymbol{S}_{\lambda_0}^{-1}(\boldsymbol{x} - \boldsymbol{S}_{\lambda_0}\boldsymbol{y}) - \boldsymbol{y}^T\boldsymbol{S}_{\lambda_0}\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{y} + \nu_0\sigma_0^2 \\
&= \frac{1}{-\boldsymbol{y}^T\boldsymbol{S}_{\lambda_0}\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{y} + \nu_0\sigma_0^2} (\boldsymbol{x} - \boldsymbol{S}_{\lambda_0}\boldsymbol{y})^T \boldsymbol{S}_{\lambda_0}^{-1} (\boldsymbol{x} - \boldsymbol{S}_{\lambda_0}\boldsymbol{y}) + 1. \qquad \text{(A.4)}
\end{aligned}
$$

The relevant part of the t_ν density function is of the form $1 + \frac{1}{\nu}(\boldsymbol{x} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})$. Equation (A.4) already looks quite similar, but it is necessary to add an additional parameter ν. This results in

$$
= 1 + (\boldsymbol{x} - \boldsymbol{S}_{\lambda_0}\boldsymbol{y})^T \boldsymbol{S}_{\lambda_0}^{-1} \left(\frac{\nu}{\boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{y}^T\boldsymbol{S}_{\lambda_0}\boldsymbol{y} + \nu_0\sigma_0^2} \right) (\boldsymbol{x} - \boldsymbol{S}_{\lambda_0}\boldsymbol{y})
$$

and the parameters of $t_\nu(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ are

$$
\begin{aligned}
\boldsymbol{\mu} &= \boldsymbol{S}_{\lambda_0}\boldsymbol{y} \\
\boldsymbol{\Sigma} &= \boldsymbol{S}_{\lambda 0} \left(\frac{\boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{y}^T\boldsymbol{\mu} + \nu_0\sigma_0^2}{\nu} \right) \\
\nu &= \nu_0 + n - 1.
\end{aligned}
$$

A.2. The Roughness Penalty Smoother S_λ

The operator

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

is a roughness penalty smoother that minimizes the penalized loss defined by $\boldsymbol{Q}$,

$$\boldsymbol{S}_\lambda \boldsymbol{x} = \underset{\boldsymbol{u}}{\operatorname{argmin}} \left\{ \|\boldsymbol{x} - \boldsymbol{u}\|^2 + \lambda \boldsymbol{u}^T \boldsymbol{Q} \boldsymbol{u} \right\}. \tag{A.5}$$

For deriving this, the argmin argument of (A.5) needs further investigation. The derivative of this expression with respect to $\boldsymbol{u}$

$$\begin{aligned} &\frac{d}{d\boldsymbol{u}} \left((\boldsymbol{x} - \boldsymbol{u})^T (\boldsymbol{x} - \boldsymbol{u}) + \lambda \boldsymbol{u}^T \boldsymbol{Q} \boldsymbol{u} \right) \\ &= \frac{d}{d\boldsymbol{u}} \left(\boldsymbol{x}^T \boldsymbol{x} - 2\boldsymbol{x}^T \boldsymbol{u} + \boldsymbol{u}^T \boldsymbol{u} + \lambda \boldsymbol{u}^T \boldsymbol{Q} \boldsymbol{u} \right) \\ &= -2\boldsymbol{x} + 2\boldsymbol{u} + 2\lambda \boldsymbol{Q} \boldsymbol{u} \end{aligned}$$

has to be set to zero:

$$\begin{aligned} & -2\boldsymbol{x} + 2\boldsymbol{u} + 2\lambda \boldsymbol{Q} \boldsymbol{u} = 0 \\ \Leftrightarrow \quad & -\boldsymbol{x} + \boldsymbol{u} + \lambda \boldsymbol{Q} \boldsymbol{u} = 0 \\ \Leftrightarrow \quad & \boldsymbol{u} + \lambda \boldsymbol{Q} \boldsymbol{u} = \boldsymbol{x} \\ \Leftrightarrow \quad & (\boldsymbol{I} + \lambda \boldsymbol{Q}) \boldsymbol{u} = \boldsymbol{x} \\ \Leftrightarrow \quad & (\boldsymbol{I} + \lambda \boldsymbol{Q})^{-1} (\boldsymbol{I} + \lambda \boldsymbol{Q}) \boldsymbol{u} = (\boldsymbol{I} + \lambda \boldsymbol{Q})^{-1} \boldsymbol{x} \\ \Leftrightarrow \quad & \boldsymbol{u} = \boldsymbol{S}_\lambda \boldsymbol{x}. \end{aligned}$$

Based on the eigenvalue/eigenvector decompostion of $\boldsymbol{Q} = \sum_{j=1}^{n} \gamma_j \boldsymbol{v}_j \boldsymbol{v}_j^T$, we have

$$\begin{aligned} & \boldsymbol{I} + \lambda \boldsymbol{Q} = \sum_{j=1}^{n} \boldsymbol{v}_j \boldsymbol{v}_j^T + \lambda \sum_{j=1}^{n} \gamma_j \boldsymbol{v}_j \boldsymbol{v}_j^T = \sum_{j=1}^{n} (1 + \lambda \gamma_j) \boldsymbol{v}_j \boldsymbol{v}_j^T \\ \Leftrightarrow \quad & \boldsymbol{u} = \boldsymbol{S}_\lambda \boldsymbol{x} = \sum_{j=1}^{n} (1 + \lambda \gamma_j)^{-1} (\boldsymbol{v}_j^T \boldsymbol{x}) \boldsymbol{v}_j, \end{aligned} \tag{A.6}$$

and (A.6) is equal to $\boldsymbol{S}_\lambda \boldsymbol{x}$ in equation (2.7).

A.3. Data Sets With Missing Values

To investigate the performance of `mrbsizeR` on objects that feature missing values, an artificial image with details on different scales is analyzed. Figure A.1 shows the image that originally has been used in Holmström *et al.* (2011) to illustrate the MRBSiZer method.

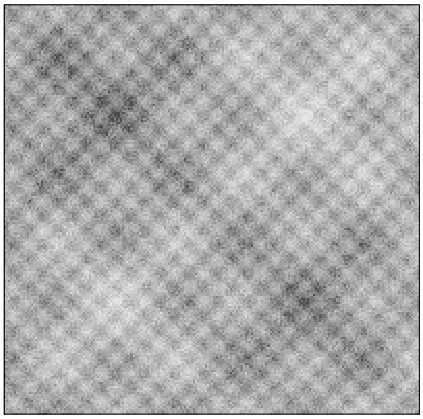

Figure A.1.: Artificial image with details on different scales.

The `mrbsizeR` analysis results of Figure A.1 using the smoothing parameter sequence $[0, 0.0001, 80, 320'000, \infty]$, 500 samples and the parameters $\lambda = 1$, $\sigma_0 = 19.6$, $\nu_0 = 50$ are visible in Figures A.2 and A.3. Especially the larger-scale features found are mostly credible. For finding out how `mrbsizeR` performs on objects featuring missing data, a part of the pixels in Figure A.1 was removed and mean imputation was used to replace the missing values. A `mrbsizeR` analysis has then been carried out and the resulting HPW-maps with 10% and 20% missing values are shown in Figures A.4 and A.5, respectively. The corresponding differences of smooths are omitted. With increasing number of missing values, the credible areas found in Figures A.4 and A.5 get smaller. Especially small-scale details are lost, see for example detail component z_2. Nevertheless, even with 20% missing values, the results using mean imputation are still meaningful and apart from the smallest-scale details, all relevant features are found. This emphasizes that, when the values are missing at random, simple methods like mean imputation are sufficient for generating meaningful differences of smooths and credibility maps.

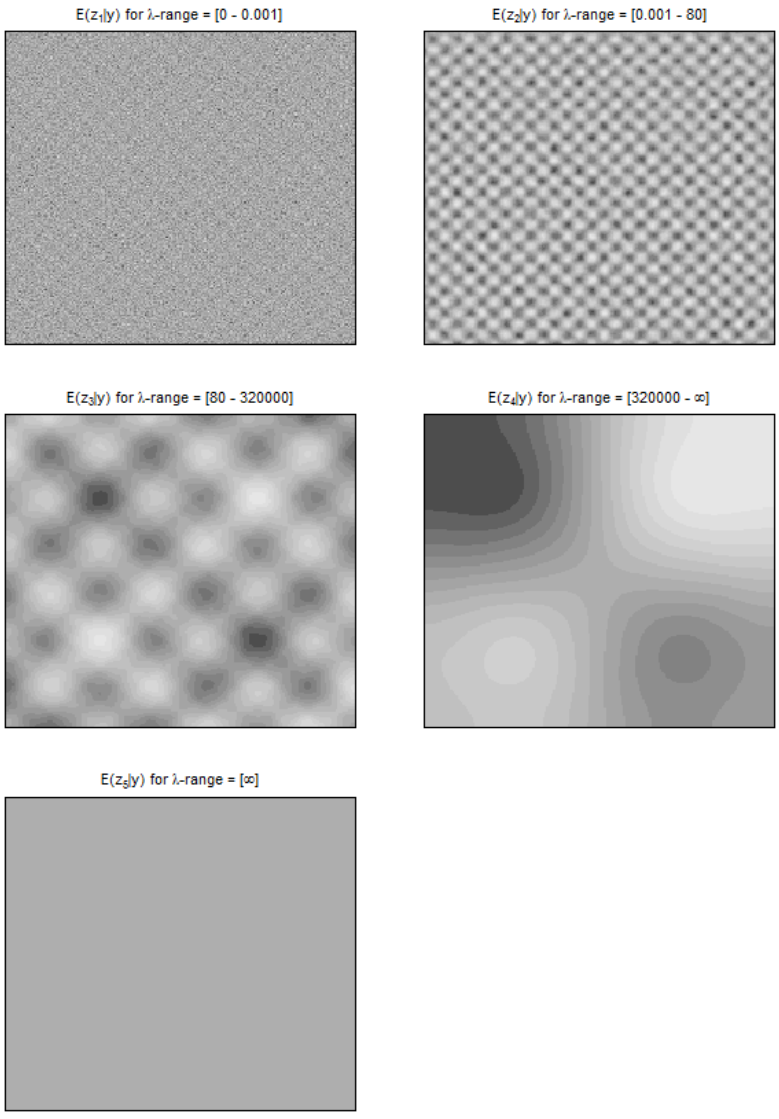

Figure A.2.: Decomposition of Figure A.1 into differences of smooths at neighboring scales. The details $z_1, \ldots, z_4$ are summarized by their posterior means.

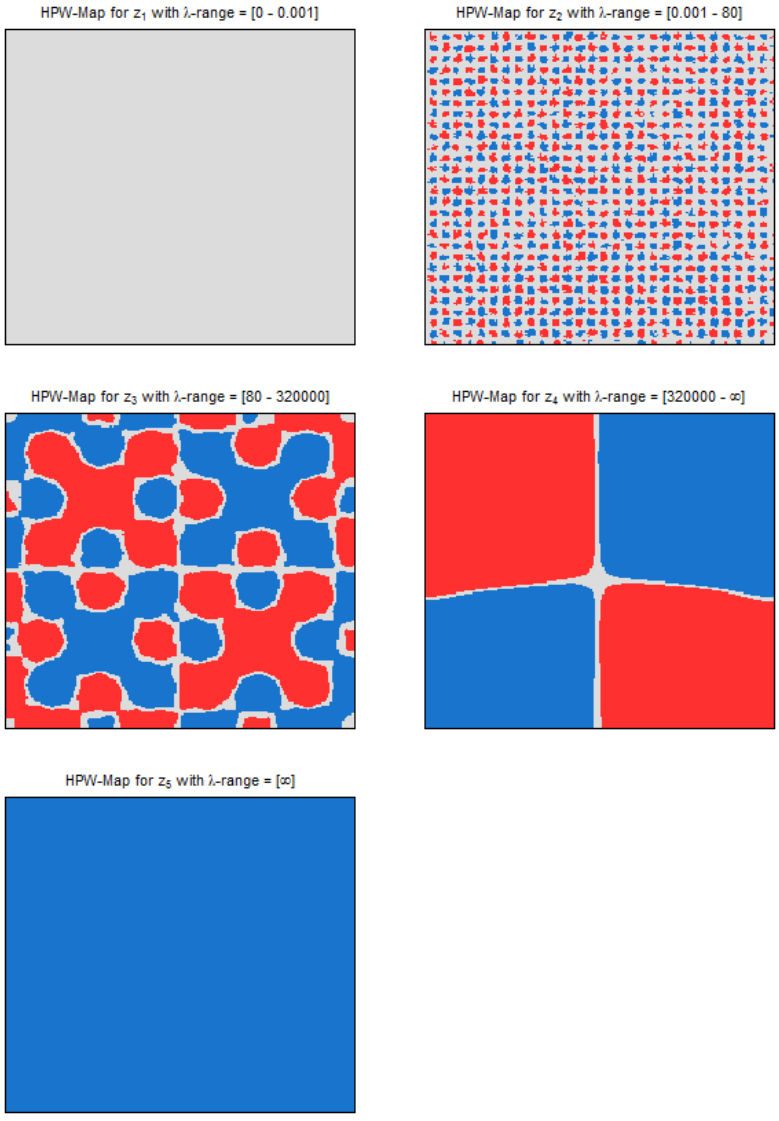

Figure A.3.: Highest pointwise probability (HPW) maps for Figure A.1. Credibly brighter areas are colored blue, credibly darker areas red. Areas without credibilty are gray.

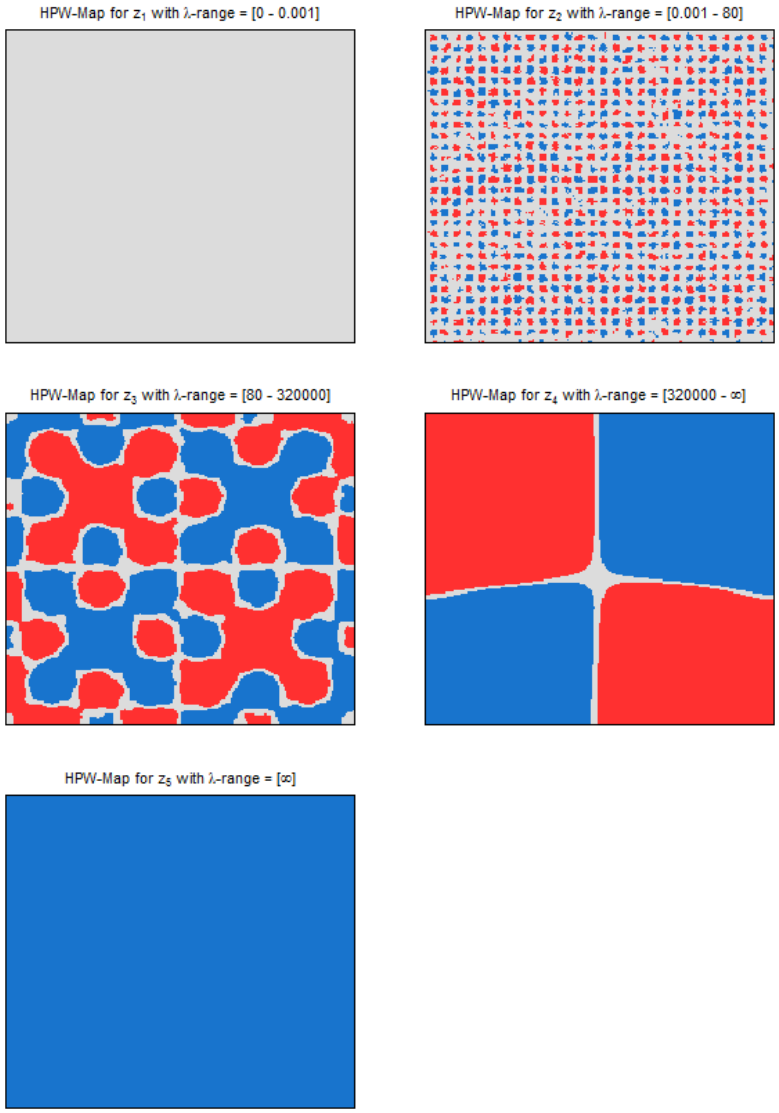

Figure A.4.: HPW maps for the image in Figure A.1 with 10% missing values, replaced by mean imputation. Credibly brighter areas are colored blue, credibly darker areas red. Areas without credibilty are gray.

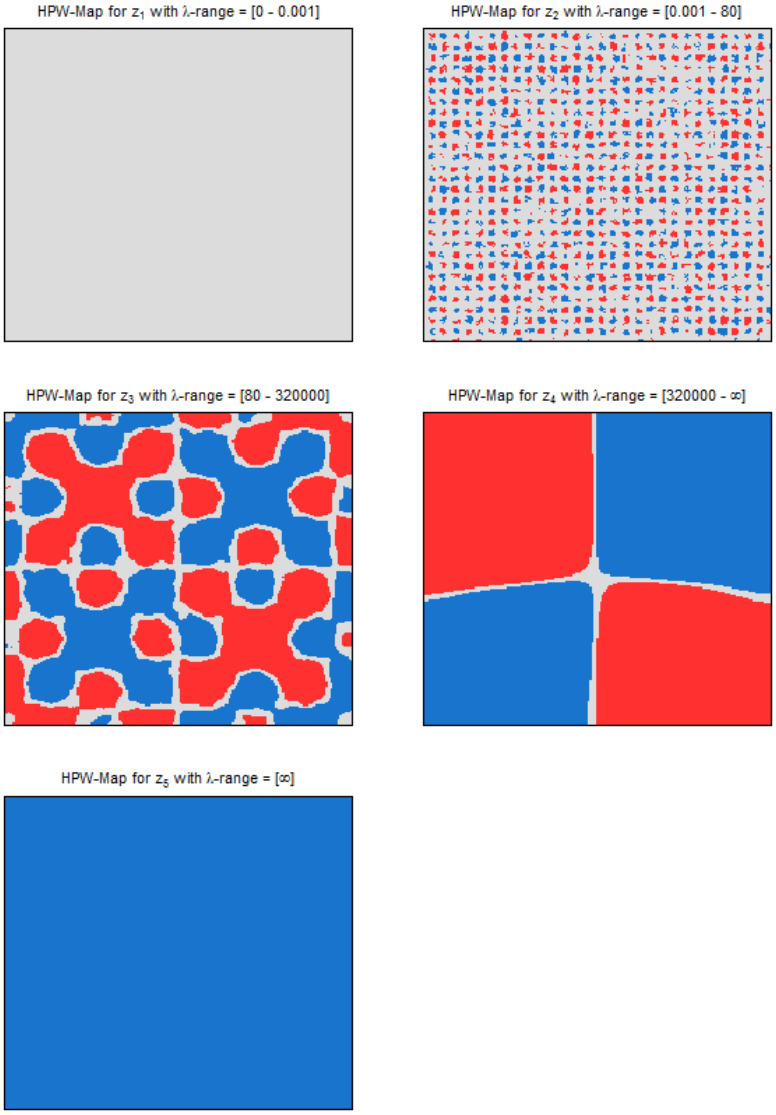

Figure A.5.: HPW maps for the image in Figure A.1 with 20% missing values, replaced by mean imputation. Credibly brighter areas are colored blue, credibly darker areas red. Areas without credibilty are gray.

A.4. Manual of `mrbsizeR` Package

Appendix A.4 contains the manual of `mrbsizeR`. The manual is a collection of all object documentations in the package. This includes a description of each object, the arguments, the output values and some examples. In contrast to the vignette in Chapter 3, a manual doesn't help finding the object one needs to solve a problem. It is helpful if the relevant object is already known: In this case it can be used as a reference manual which includes all relevant details for each object.

Package 'mrbsizeR'

April 2, 2017

Type Package

Title Scale Space Multiresolution Analysis of Random Signals

Version 1.0.1

Author Thimo Schuster [cre, aut],
Leena Pasanen [ctb],
Reinhard Furrer [ctb]

Maintainer Thimo Schuster <thimo.schuster@gmail.com>

Description A method for the multiresolution analysis of spatial fields and images to capture scale-dependent features. mrbsizeR is based on scale space smoothing and uses differences of smooths at neighbouring scales for finding features on different scales. To infer which of the captured features are credible, Bayesian analysis is used.
The scale space multiresolution analysis has three steps: (1) Bayesian signal reconstruction. (2) Using differences of smooths, scale-dependent features of the reconstructed signal can be found. (3) Posterior credibility analysis of the differences of smooths created. The method has first been proposed by Holmstrom, Pasanen, Furrer, Sain (2011) <DOI:10.1016/j.csda.2011.04.011>.

License GPL-2

URL http://cc.oulu.fi/~lpasanen/MRBSiZer/

LazyData TRUE

Depends R (>= 3.0.0), maps(>= 3.1.1)

Imports fields (>= 8.10), stats (>= 3.0.0), grDevices(>= 3.0.0), graphics(>= 3.0.0), methods(>= 3.0.0)

RoxygenNote 6.0.1

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

`CImap` *Computation of simultaneous credible intervals.*

Description

Simultaneous credible intervals for all differences of smooths at neighboring scales z_i are computed.

Usage

```
CImap(smoothVec, mm, nn, prob = 0.95)
```

Arguments

`smoothVec`	Differences of smooths at neighboring scales.
`mm`	Number of rows of the original input object.
`nn`	Number of columns of the original input object.
`prob`	Credibility level for the posterior credibility analysis. By default `prob = 0.95`.

Details

`CImap` is an internal function of `mrbsizeRgrid` and is usually not used independently. The output can be analyzed with the plotting function `plot.CImapGrid`.

Value

An array with simultaneous credible intervals `VmapCI` and the dimensions of the original input object, `mm` and `nn`.

Examples

```
# Artificial sample data: 10 observations (5-by-2 object), 10 samples
set.seed(987)
sampleData  <- matrix(stats::rnorm(100), nrow = 10)
sampleData [4:6, ] <- sampleData [4:6, ] + 5

# Calculation of the simultaneous credible intervals
CImap(smoothVec = sampleData , mm = 5, nn = 2, prob = 0.95)
```

`dctMatrix` *Create a n-by-n discrete cosine transform matrix.*

Description

The discrete cosine transform (DCT) matrix for a given dimension n is calculated.

Usage

```
dctMatrix(n)
```

Arguments

`n`	Dimension for the DCT matrix.

Details

The function can be used for 1D- or 2D-DCT transforms of data.

- **1D:** Let `Q` be a m-by-n matrix with some data. `D` is a m-by-m DCT matrix created by `dctMatrix(m)`. Then `D %*% Q` returns the discrete cosine transform of the columns of Q. `t(D) %*% Q` returns the inverse DCT of the columns of Q. As D is orthogonal, `solve(D) = t(D)`.
- **2D:** Let `Q` be a m-by-n matrix with some data. `D_m` is a m-by-m DCT matrix created by `dctMatrix(m)`, `D_n` a n-by-n DCT matrix created by `dctMatrix(n)`. `D_m %*% Q %*% t(D_n)` computes the 2D-DCT of Q. The inverse 2D-DCT of Q can be computed via `t(D_mm) %*% DCT_Q %*% D_n`. D_m transforms along columns, D_n along rows. Since D is orthogonal, `solve(D) = t(D)`.

It can be faster to use `dctMatrix` than using a direct transformation, especially when calculating several DCT's.

Value

The n-by-n DCT matrix.

Examples

```
D <- dctMatrix(5)
```

`dftMatrix` *Create a n-by-n discrete Fourier transform matrix.*

Description

The discrete Fourier transform (DFT) matrix for a given dimension n is calculated.

Usage

```
dftMatrix(n)
```

Arguments

`n`	Dimension for the DFT matrix.

Details

The DFT matrix can be used for computing the discrete Fourier transform of a matrix or vector. `dftMatrix(n) %*% testMatrix` is the same as `apply(testMatrix, MARGIN = 2, FUN = fft)`.

Value

The n-by-n DFT matrix.

Examples

```
set.seed(987)
testMatrix <- matrix(sample(1:10, size = 25, replace = TRUE), nrow = 5)
D <- dftMatrix(5)

# Discrete Fourier transform with matrix multiplication:
D %*% testMatrix

# Discrete Fourier transform with function fft:
apply(testMatrix, MARGIN = 2, FUN = fft)
```

`eigenLaplace` *Generate eigenvalues of discrete Laplace matrix.*

Description

The eigenvalues of a discrete Laplace matrix with dimension (`mm`, `nn`) are calculated.

Usage

```
eigenLaplace(mm, nn)
```

Arguments

`mm`	Number of rows of the discrete Laplace matrix.
`nn`	Number of columns of the discrete Laplace matrix.

Value

A row vector containing the eigenvalues of the discrete laplace matrix.

Examples

```
eigval <- eigenLaplace(5, 5)
```

---

eigenQsphere *Generate eigenvalues of precision matrix Q on the surface of a sphere.*

Description

The eigenvalues of the precision matrix Q with dimension (`mm`, `nn`) and polar angle limits `phimin`, `phimax` are calculated.

Usage

```
eigenQsphere(phimin, phimax, mm, nn)
```

Arguments

`phimin`	Polar angle minimum.
`phimax`	Polar angle maximum.
`mm`	Number of rows of precision matrix Q.
`nn`	Number of columns of precision matrix Q.

Details

The corresponding function for data on a grid is `eigenLaplace`.

Value

A list containing 2 elements:

- eigval Row vector containing the eigenvalues of Q.
- eigvec Matrix containing the eigenvectors of Q as columns.

Examples

```
eig_out <- eigenQsphere(phimin = 180/10, phimax = 180 - 180/10, mm = 10, nn = 20)
```

`fftshift` *Swap the quadrants or halves of a 2d matrix.*

Description

`fftshift` is an R equivalent to the Matlab function `fftshift` applied on matrices. For more information about `fftshift` see the Matlab documentation.

Usage

```
fftshift(inputMatrix, dimension = -1)
```

Arguments

`inputMatrix`	Matrix to be swapped.
`dimension`	Which swap should be performed? • `1`: swap halves along the rows. • `2`: swap halves along the columns. • `-1`: swap first quadrant with third and second quadrant with fourth.

Details

It is possible to swap the halves or the quadrants of the input matrix. Halves can be swapped along the rows (`dimension = 1`) or along the columns (`dimension = 2`). When swapping the quadrants, `fftshift` swaps the first quadrant with the third and the second quadrant with the fourth (`dimension = -1`).

Value

Swapped matrix.

Examples

```
set.seed(987)
sampleMat <- matrix(sample(1:10, size = 25, replace = TRUE), nrow = 5)

# Swap halves along the rows:
fftshift(sampleMat, dimension = 1)

# Swap halves along the columns:
fftshift(sampleMat, dimension = 2)

# Swap first quadrant with third and second quadrant with fourth:
fftshift(sampleMat, dimension = -1)
```

HPWmap *Computation of pointwise and highest pointwise probabilities.*

Description

Pointwise (PW) probabilities and highest pointwise (HPW) probabilities of all differences of smooths at neighboring scales are computed.

Usage

```
HPWmap(smoothVec, mm, nn, prob = 0.95)
```

Arguments

smoothVec	Differences of smooths at neighboring scales.
mm	Number of rows of the original input image.
nn	Number of columns of the original input image.
prob	Credibility level for the posterior credibility analysis

Details

HPWmap is an internal function of mrbsizeRgrid and is usually not used independently. The output can be analyzed with the plotting function plot.HPWmapGrid.

Value

List with two arrays:

- pw: Pointwise probabilities (VmapPW) including the dimensions of the original input image, mm and nn.
- hpw: Highest pointwise probabilities (VmapHPW) including the dimensions of the original input image, mm and nn.

Examples

```
# Artificial sample data: 10 observations (5-by-2 object), 10 samples
set.seed(987)
sampleData <- matrix(stats::rnorm(100), nrow = 10)
sampleData[4:6, ] <- sampleData[4:6, ] + 5

# Calculation of the simultaneous credible intervals
HPWmap(smoothVec = sampleData, mm = 5, nn = 2, prob = 0.95)
```

`ifftshift` *Inverse FFT shift of a 2d matrix.*

Description

`ifftshift` is an R equivalent to the Matlab function `ifftshift` applied on matrices. For more information about `ifftshift` see the Matlab documentation.

Usage

```
ifftshift(inputMatrix, dimension = -1)
```

Arguments

`inputMatrix`	Matrix to be swapped.
`dimension`	Which swap should be performed? • `1`: swap halves along the rows. • `2`: swap halves along the columns. • `-1`: swap first quadrant with third and second quadrant with fourth.

Details

`ifftshift` is the inverse function to `fftshift`. For more information see the details of `fftshift`

Value

Swapped matrix.

Examples

```
set.seed(987)
sampleMat <- matrix(sample(1:10, size = 25, replace = TRUE), nrow = 5)

# Swap halves along the rows:
ifftshift(sampleMat, dimension = 1)

# Swap halves along the columns:
ifftshift(sampleMat, dimension = 2)

# Swap first quadrant with third and second quadrant with fourth:
ifftshift(sampleMat, dimension = -1)
```

`MinLambda` *Numerical optimization for finding appropriate smoothing levels.*

Description

Numerical optimization of an objective function G is carried out to find appropriate signal-dependent smoothing levels (λ's). This is easier than visual inspection via the signal-dependent tapering function in `TaperingPlot`.

Usage

```
MinLambda(Xmu, mm, nn, nGrid, nLambda = 2, lambda, sphere = FALSE)
```

Arguments

`Xmu`	Posterior mean of the input object as a vector.
`mm`	Number of rows of the original input object.
`nn`	Number of columns of the original input object.
`nGrid`	Size of grid where objective function is evaluated (nGrid-by-nGrid).
`nLambda`	Number of lambdas to minimize over. Possible arguments: 2 (default) or 3.
`lambda`	λ-sequence which is used for optimization. If nothing is provided, `lambda <- 10^seq(-3, 10, len = nGrid)` is used for data on a grid and `lambda <- 10^seq(-6, 1, len = nGrid)` is used for spherical data.
`sphere`	`TRUE` or `FALSE`: Is the input object defined on a sphere?

Details

As signal-dependent tapering functions are quiet irregular, it is hard to find appropriate smoothing values only by visual inspection of the tapering function plot. A more formal approach is the numerical optimization of an objective function.

Optimization can be carried out with 2 or 3 smoothing parameters. As the smoothing parameters 0 and ∞ are always added, this results in a mrbsizeR analysis with 4 or 5 smoothing parameters.

Sometimes, not all features of the input object can be extracted using the smoothing levels proposed by `MinLambda`. It might then be necessary to include additional smoothing levels.

`plot.minLambda` creates a plot of the objective function G on a grid. The minimum is indicated with a white point. The minimum values of the λ's can be extracted from the output of `MinLambda`, see examples.

Value

A list with 3 objects:

`G` Value of objective function G.

`lambda` Evaluated smoothing parameters λ.

`minind` Index of minimal λ's. `lambda[minind]` gives the minimal values.

Examples

```
# Artificial sample data
set.seed(987)
sampleData <- matrix(stats::rnorm(100), nrow = 10)
sampleData[4:6, 6:8] <- sampleData[4:6, 6:8] + 5

# Minimization of two lambdas on a 20-by-20-grid
minlamOut <- MinLambda(Xmu = c(sampleData), mm = 10, nn = 10,
                       nGrid = 20, nLambda = 2)

# Minimal lambda values
minlamOut$lambda[minlamOut$minind]
```

`mrbsizeR` *mrbsizeR: Scale space multiresolution analysis in R.*

Description

`mrbsizeR` contains a method for the scale space multiresolution analysis of spatial fields and images to capture scale-dependent features. The name is an abbreviation for **M**ulti**R**esolution **B**ayesian **SI**gnificant **ZE**ro crossings of derivatives in **R** and the method combines the concept of statistical scale space analysis with a Bayesian SiZer method.

Details

The `mrbsizeR` analysis can be applied to data on a regular grid and to spherical data. For data on a grid, the scale space multiresolution analysis has three steps:

1. Bayesian signal reconstruction.
2. Using differences of smooths, scale-dependent features of the reconstructed signal are found.
3. Posterior credibility analysis of the differences of smooths created.

In a first step, Bayesian signal reconstruction is used to extract an underlying signal from a potentially noisy observation. Samples of the resulting posterior can be generated and used for the analysis. For finding features on different scales, differences of smooths at neighboring scales are used. This is an important distinction to other scale space methods (which usually use a wide range of smoothing levels without taking differences) and tries to separate the features into distinct scale categories more aggressively. After a successful extraction of the scale-different features, posterior credibility analysis is necessary to assess whether the features found are "really there" or if they are artifacts of random sampling.

For spherical data, no Bayesian signal reconstruction is implemented in `mrbsizer`. Data samples therefore need to be available beforehand. The analysis procedure can therefore be summarized in two steps:

1. Using differences of smooths, scale-dependent features of the reconstructed signal are found.
2. Posterior credibility analysis of the differences of smooths created.

This method has first been proposed by Holmstrom, Pasanen, Furrer, Sain (2011), see also http://cc.oulu.fi/~lpasanen/MRBSiZer/.

Major Functions

- `TaperingPlot` Graphical estimation of useful smoothing levels. Can be used signal-independent and signal-dependent.
- `MinLambda` Numerical estimation of useful smoothing levels. Takes the underlying signal into account. `plot.minLambda` can be used for plotting the result.
- `rmvtDCT` Creates samples on a regular grid from a multivariate t_ν-distribution using a discrete cosine transform (DCT).
- `mrbsizeRgrid` Interface of the mrbsizeR method for data on a regular grid. Differences of smooths at neighboring scales are created and posterior credibility analysis is conducted. The results can be visualized using `plot.smMeanGrid`,`plot.HPWmapGrid` and `plot.CImapGrid`.
- `mrbsizeRsphere` Interface of the mrbsizeR method for data on a sphere. Differences of smooths at neighboring scales are created and posterior credibility analysis is conducted. The results can be visualized using `plot.smMeanSphere`,`plot.HPWmapSphere` and `plot.CImapSphere`. For data on a sphere, no Bayesian signal reconstruction is implemented. Samples have to be provided instead.

Getting Started

The vignette for this package offers an extensive overview of the functionality and the usage of `mrbsizeR`.

References

- Holmstrom, L. and Pasanen, L. (2011). MRBSiZer. http://cc.oulu.fi/~lpasanen/MRBSiZer/. Accessed: 2017-03-04.
- Holmstrom, L., Pasanen, L., Furrer, R., and Sain, S. R. (2011). Scale space multiresolution analysis of random signals. Computational Statistics and Data Analysis, 55, 2840-2855. <DOI:10.1016/j.csda.2011.04.011>.
- Holmstrom, L. and Pasanen, L. (2016). Statistical scale space methods. International Statistical Review. <DOI:10.1111/insr.12155>.

DISCLAIMER: The author can not guarantee the correctness of any function or program in this package.

Examples

```
# Artificial sample data
set.seed(987)
sampleData <- matrix(stats::rnorm(100), nrow = 10)
sampleData[4:6, 6:8] <- sampleData[4:6, 6:8] + 5

# Generate samples from multivariate t-distribution
tSamp <- rmvtDCT(object = sampleData, lambda = 0.2, sigma = 6, nu0 = 15,
                 ns = 1000)

# mrbsizeRgrid analysis
mrbOut <- mrbsizeRgrid(posteriorFile = tSamp$sample, mm = 10, nn = 10,
                       lambdaSmoother = c(1, 1000), prob = 0.95)

# Posterior mean of the differences of smooths
plot(x = mrbOut$smMean, turn_out = TRUE)

# Credibility analysis using simultaneous credible intervals
plot(x = mrbOut$ciout, turn_out = TRUE)
```

`mrbsizeRgrid` *Multiresolution analysis of random signals.*

Description

`mrbsizeRgrid` is the interface of the scale space multiresolution method for data on a regular grid. Here, the differences of smooths as well as the posterior credibility analysis are computed. The output can be analyzed with the plotting functions `plot.smMeanGrid`, `plot.CImapGrid` and `plot.HPWmapGrid`.

Usage

```
mrbsizeRgrid(posteriorFile, mm, nn, lambdaSmoother, prob = 0.95,
  smoothOut = FALSE)
```

Arguments

`posteriorFile`	Matrix with posterior samples as column vectors.
`mm`	Number of rows of the original object.
`nn`	Number of columns of the original object.
`lambdaSmoother`	Vector consisting of the smoothing levels to be used.
`prob`	Credibility level for the posterior credibility analysis.
`smoothOut`	Should the differences of smooths at neighboring scales be returned as output (FALSE by default)?

Details

`mrbsizeRgrid` conducts two steps of the scale space multiresolution analysis:

1. Extraction of scale-dependent features from the reconstructed signal. This is done by smoothing at different smoothing levels and taking the difference of smooths at neighboring scales.
2. Posterior credibility analysis of the differences of smooths created. Three different methods are applied: Pointwise probabilities (see `HPWmap`), highest pointwise probabilities (see `HPWmap`) and simultaneous credible intervals (see `CImap`).

The signal can be reconstructed using the build-in multivariate t-distribution sampling `rmvtDCT`. It is also possible to provide samples generated with other methods, see the parameter `posteriorFile` and the examples.

For further information and examples, see the vignette.

Value

A list containing the following sublists:

`smMean` Posterior mean of all differences of smooths created.

`hpout` Pointwise (PW) and highest pointwise (HPW) probabilities of all differences of smooths created.

`ciout` Simultaneous credible intervals (CI) of all differences of smooths created.

`smoothSamples` Samples of differences of smooths at neighboring scales, as column vectors.

Examples

```
# Artificial sample data
set.seed(987)
sampleData <- matrix(stats::rnorm(100), nrow = 10)
sampleData[4:6, 6:8] <- sampleData[4:6, 6:8] + 5

# Generate samples from multivariate t-distribution
tSamp <- rmvtDCT(object = sampleData, lambda = 0.2, sigma = 6, nu0 = 15,
                 ns = 1000)

# mrbsizeRgrid analysis
mrbOut <- mrbsizeRgrid(posteriorFile = tSamp$sample, mm = 10, nn = 10,
                       lambdaSmoother = c(1, 1000), prob = 0.95)
```

---

mrbsizeRsphere *Multiresolution analysis of random signals for spherical data.*

Description

mrbsizeRSphere is the interface of the scale space multiresolution method for spherical data. Here, the differences of smooths as well as the posterior credibility analysis are computed. The output can be analyzed with the plotting functions plot.smMeanSphere, plot.CImapSphere and plot.HPWmapSphere.

Usage

```
mrbsizeRsphere(posteriorFile, mm, nn, lambdaSmoother, prob = 0.95,
  smoothOut = FALSE)
```

Arguments

posteriorFile	Matrix with posterior samples as column vectors.
mm	Number of rows of the original object.
nn	Number of columns of the original object.
lambdaSmoother	Vector consisting of the smoothing levels to be used.
prob	Credibility level for the posterior credibility analysis.
smoothOut	Should the differences of smooths at neighboring scales be returned as output (FALSE by default)?

Details

In contrast to mrbsizeRgrid, mrbsizeRsphere does not conduct Bayesian signal reconstruction via sampling from a posterior distribution. Samples of the posterior distribution have to be provided instead.

For further information and examples, see mrbsizeRgrid and the vignette.

Value

A list containing the following sublists:

`smMean` Posterior mean of all differences of smooths created.

`hpout` Pointwise (PW) and highest pointwise (HPW) probabilities of all differences of smooths created.

`ciout` Simultaneous credible intervals (CI) of all differences of smooths created.

`smoothSamples` Samples of differences of smooths at neighboring scales, as column vectors.

Examples

```
# Artificial spherical sample data
set.seed(987)
sampleData <- matrix(stats::rnorm(2000), nrow = 200)
sampleData[50:65, ] <- sampleData[50:65, ] + 5

# mrbsizeRsphere analysis
mrbOut <- mrbsizeRsphere(posteriorFile = sampleData, mm = 10, nn = 20,
                         lambdaSmoother = c(1, 1000), prob = 0.95)
```

`plot.CImapGrid` *Plot of simultaneous credible intervals.*

Description

Maps with simultaneous credible intervals for all differences of smooths at neighboring scales z_i are plotted.

Usage

```
## S3 method for class 'CImapGrid'
plot(x, color = c("firebrick1", "gainsboro",
  "dodgerblue3"), turnOut = TRUE, title, aspRatio = 1, ...)
```

Arguments

`x`	List containing the simultaneous credible intervals for all differences of smooths.
`color`	Vector of length 3 containing the colors to be used in the credibility maps. The first color represents the credibly negative pixels, the second color the pixels that are not credibly different from zero and the third color the credibly positive pixels.
`turnOut`	Logical. Should the output images be turned 90 degrees counter-clockwise?
`title`	Vector containing one string per plot. The required number of titles is equal to `length(mrbOut$ciout)`. If no `title` is passed, defaults are used.
`aspRatio`	Adjust the aspect ratio of the plots. The default `aspRatio = 1` produces square plots.
`...`	Further graphical parameters can be passed.

Details

The default colors of the maps have the following meaning:

- **Blue**: Credibly positive pixels.
- **Red**: Credibly negative pixels.
- **Grey**: Pixels that are not credibly different from zero.

`x` corresponds to the `ciout`-part of the output of `mrbsizeRgrid`.

Value

Plots of simultaneous credible intervals for all differences of smooths are created.

Examples

```
# Artificial sample data
set.seed(987)
sampleData <- matrix(stats::rnorm(100), nrow = 10)
sampleData[4:6, 6:8] <- sampleData[4:6, 6:8] + 5

# Generate samples from multivariate t-distribution
tSamp <- rmvtDCT(object = sampleData, lambda = 0.2, sigma = 6, nu0 = 15,
                  ns = 1000)

# mrbsizeRgrid analysis
mrbOut <- mrbsizeRgrid(posteriorFile = tSamp$sample, mm = 10, nn = 10,
                       lambdaSmoother = c(1, 1000), prob = 0.95)

# Posterior mean of the differences of smooths
plot(x = mrbOut$smMean, turnOut = TRUE)

# Credibility analysis using simultaneous credible intervals
plot(x = mrbOut$ciout, turnOut = TRUE)
```

`plot.CImapSphere` *Plotting of simultaneous credible intervals on a sphere.*

Description

Maps with simultaneous credible intervals for all differences of smooths at neighboring scales z_i are plotted. Continental lines are added.

Usage

```
## S3 method for class 'CImapSphere'
plot(x, lon, lat, color = c("firebrick1", "gainsboro",
  "dodgerblue3"), turnOut = FALSE, title, ...)
```

Arguments

x	List containing the simultaneous credible intervals of all differences of smooths.
lon	Vector containing the longitudes of the data points.
lat	Vector containing the latitudes of the data points.
color	Vector of length 3 containing the colors to be used in the credibility maps. The first color represents the credibly negative pixels, the second color the pixels that are not credibly different from zero and the third color the credibly positive pixels.
turnOut	Logical. Should the output images be turned 90 degrees counter-clockwise?
title	Vector containing one string per plot. The required number of titles is equal to `length(mrbOut$ciout)`. If no `title` is passed, defaults are used.
...	Further graphical parameters can be passed.

Details

The default colors of the maps have the following meaning:

- **Blue**: Credibly positive pixels.
- **Red**: Credibly negative pixels.
- **Grey**: Pixels that are not credibly different from zero.

`x` corresponds to the `ciout`-part of the output of `mrbsizeRsphere`.

Value

Plots of simultaneous credible intervals for all differences of smooths are created.

Examples

```
# Artificial spherical sample data
set.seed(987)
sampleData <- matrix(stats::rnorm(2000), nrow = 200)
sampleData[50:65, ] <- sampleData[50:65, ] + 5
lon <- seq(-180, 180, length.out = 20)
lat <- seq(-90, 90, length.out = 10)

# mrbsizeRsphere analysis
mrbOut <- mrbsizeRsphere(posteriorFile = sampleData, mm = 20, nn = 10,
                         lambdaSmoother = c(0.1, 1), prob = 0.95)

# Posterior mean of the differences of smooths
plot(x = mrbOut$smMean, lon = lon, lat = lat,
     color = fields::tim.colors())

# Credibility analysis using simultaneous credible intervals
plot(x = mrbOut$ciout, lon = lon, lat = lat)
```

`plot.HPWmapGrid` *Plotting of pointwise and highest pointwise probabilities.*

Description

Maps with pointwise (PW) probabilities and/or highest pointwise (HPW) probabilities of all differences of smooths at neighboring scales are plotted.

Usage

```
## S3 method for class 'HPWmapGrid'
plot(x, plotWhich = "Both", color = c("firebrick1",
  "gainsboro", "dodgerblue3"), turnOut = TRUE, title, aspRatio = 1, ...)
```

Arguments

`x`	List containing the pointwise (PW) and highest pointwise (HPW) probabilities of all differences of smooths.
`plotWhich`	Which probabilities shall be plotted? `HPW`, `PW` or `Both`?
`color`	Vector of length 3 containing the colors to be used in the credibility maps. The first color represents the credibly negative pixels, the second color the pixels that are not credibly different from zero and the third color the credibly positive pixels.
`turnOut`	Logical. Should the output images be turned 90 degrees counter-clockwise?
`title`	Vector containing one string per plot. The required number of titles is equal to `length(mrbOut$hpout)`. If no `title` is passed, defaults are used.
`aspRatio`	Adjust the aspect ratio of the plots. The default `aspRatio = 1` produces square plots.
`...`	Further graphical parameters can be passed.

Details

The default colors of the maps have the following meaning:

- **Blue**: Credibly positive pixels.
- **Red**: Credibly negative pixels.
- **Grey**: Pixels that are not credibly different from zero.

`x` corresponds to the `hpout`-part of the output of `mrbsizeRgrid`.

Value

Plots of pointwise and/or highest pointwise probabilities for all differences of smooths are created.

Examples

```
# Artificial sample data
set.seed(987)
sampleData <- matrix(stats::rnorm(100), nrow = 10)
sampleData[4:6, 6:8] <- sampleData[4:6, 6:8] + 5

# Generate samples from multivariate t-distribution
tSamp <- rmvtDCT(object = sampleData, lambda = 0.2, sigma = 6, nu0 = 15,
                 ns = 1000)

# mrbsizeRgrid analysis
mrbOut <- mrbsizeRgrid(posteriorFile = tSamp$sample, mm = 10, nn = 10,
                       lambdaSmoother = c(1, 1000), prob = 0.95)

# Posterior mean of the differences of smooths
plot(x = mrbOut$smMean, turnOut = TRUE)

# Credibility analysis using pointwise (PW) maps
plot(x = mrbOut$hpout, plotWhich = "PW", turnOut = TRUE)

# Credibility analysis using highest pointwise probability (HPW) maps
plot(x = mrbOut$hpout, plotWhich = "HPW", turnOut = TRUE)
```

---

plot.HPWmapSphere *Plotting of pointwise and highest pointwise probabilities on a sphere.*

Description

Maps with pointwise (PW) probabilities and/or highest pointwise (HPW) probabilities of all differences of smooths at neighboring scales are plotted. Continental lines are added.

Usage

```
## S3 method for class 'HPWmapSphere'
plot(x, lon, lat, plotWhich = "Both",
  color = c("firebrick1", "gainsboro", "dodgerblue3"), turnOut = FALSE,
  title, ...)
```

Arguments

x	List containing the pointwise (PW) and highest pointwise (HPW) probabilities of all differences of smooths.
lon	Vector containing the longitudes of the data points.
lat	Vector containing the latitudes of the data points.
plotWhich	Which probabilities shall be plotted? HPW, PW or Both?
color	Vector of length 3 containing the colors to be used in the credibility maps. The first color represents the credibly negative pixels, the second color the pixels that are not credibly different from zero and the third color the credibly positive pixels.
turnOut	Logical. Should the output images be turned 90 degrees counter-clockwise?

`title`	Vector containing one string per plot. The required number of titles is equal to `length(mrbOut$hpout)`. If no `title` is passed, defaults are used.
`...`	Further graphical parameters can be passed.

Details

The default colors of the maps have the following meaning:

- **Blue**: Credibly positive pixels.
- **Red**: Credibly negative pixels.
- **Grey**: Pixels that are not credibly different from zero.

`x` corresponds to the `hpout`-part of the output of `mrbsizeRsphere`.

Value

Plots of pointwise and/or highest pointwise probabilities for all differences of smooths are created.

Examples

```
# Artificial spherical sample data
set.seed(987)
sampleData <- matrix(stats::rnorm(2000), nrow = 200)
sampleData[50:65, ] <- sampleData[50:65, ] + 5
lon <- seq(-180, 180, length.out = 20)
lat <- seq(-90, 90, length.out = 10)

# mrbsizeRsphere analysis
mrbOut <- mrbsizeRsphere(posteriorFile = sampleData, mm = 20, nn = 10,
                         lambdaSmoother = c(0.1, 1), prob = 0.95)

# Posterior mean of the differences of smooths
plot(x = mrbOut$smMean, lon = lon, lat = lat,
     color = fields::tim.colors())

# Credibility analysis using pointwise (PW) maps
plot(x = mrbOut$hpout, lon = lon, lat = lat, plotWhich = "PW")

# Credibility analysis using highest pointwise probability (HPW) maps
plot(x = mrbOut$hpout, lon = lon, lat = lat, plotWhich = "HPW")
```

`plot.minLambda` *Plot of objective function for finding appropriate smoothing parameters.*

Description

The objective function G is plotted on a grid. The minimum is indicated with a white point.

Usage

```
## S3 method for class 'minLambda'
plot(x, ...)
```

Arguments

`x`	Output of function `MinLambda`.
`...`	Further graphical parameters can be passed.

Details

When minimizing over 2 λ's, one plot is generated: (λ_2 vs λ_3). With 3 λ's, 3 plots are generated: λ_2 vs. λ_3, λ_2 vs. λ_4 and λ_3 vs. λ_4.

Value

Plot of G on a grid.

Examples

```
set.seed(987)
sampleData <- matrix(stats::rnorm(100), nrow = 10)
sampleData[4:6, 6:8] <- sampleData[4:6, 6:8] + 5

# Minimization of two lambdas on a 20-by-20-grid
minLamOut <- MinLambda(Xmu = c(sampleData), mm = 10, nn = 10,
                       nGrid = 20, nLambda = 3)

# Plot of the objective function
plot(x = minLamOut)
```

`plot.smMeanGrid` *Plotting of scale-dependent features.*

Description

Scale-dependent features are plotted using differences of smooths at neighboring scales. The features are summarized by their posterior mean.

Usage

```
## S3 method for class 'smMeanGrid'
plot(x, color.pallet = fields::tim.colors(),
  turnOut = TRUE, title, aspRatio = 1, ...)
```

Arguments

`x`	List containing the posterior mean of all differences of smooths.
`color.pallet`	The color pallet to be used for plotting scale-dependent features.
`turnOut`	Logical. Should the output images be turned 90 degrees counter-clockwise?
`title`	Vector containing one string per plot. The required number of titles is equal to `length(mrbOut$smMean)`. If no `title` is passed, defaults are used.
`aspRatio`	Adjust the aspect ratio of the plots. The default `aspRatio = 1` produces square plots.
`...`	Further graphical parameters can be passed.

Details

x corresponds to the smmean-part of the output of mrbsizeRgrid.

Value

Plots of the differences of smooths are created.

Examples

```
# Artificial sample data
set.seed(987)
sampleData <- matrix(stats::rnorm(100), nrow = 10)
sampleData[4:6, 6:8] <- sampleData[4:6, 6:8] + 5

# Generate samples from multivariate t-distribution
tSamp <- rmvtDCT(object = sampleData, lambda = 0.2, sigma = 6, nu0 = 15,
                 ns = 1000)

# mrbsizeRgrid analysis
mrbOut <- mrbsizeRgrid(posteriorFile = tSamp$sample, mm = 10, nn = 10,
                       lambdaSmoother = c(1, 1000), prob = 0.95)

# Posterior mean of the differences of smooths
plot(x = mrbOut$smMean, turnOut = TRUE)
```

---

plot.smMeanSphere *Plotting of scale-dependent features on a sphere.*

Description

Scale-dependent features are plotted using differences of smooths at neighboring scales. The features are summarized by their posterior mean. Continental lines are added to the plots.

Usage

```
## S3 method for class 'smMeanSphere'
plot(x, lon, lat, color.pallet = fields::tim.colors(),
  turnOut = TRUE, title, ...)
```

Arguments

x	List containing the posterior mean of all differences of smooths.
lon	Vector containing the longitudes of the data points.
lat	Vector containing the latitudes of the data points.
color.pallet	The color pallet to be used for plotting scale-dependent features.
turnOut	Logical. Should the output images be turned 90 degrees counter-clockwise?
title	Vector containing one string per plot. The required number of titles is equal to length(mrbOut$smMean). If no title is passed, defaults are used.
...	Further graphical parameters can be passed.

Details

x corresponds to the smmean-part of the output of mrbsizeRsphere.

Value

Plots of the differences of smooths are created.

Examples

```
# Artificial spherical sample data
set.seed(987)
sampleData <- matrix(stats::rnorm(2000), nrow = 200)
sampleData[50:65, ] <- sampleData[50:65, ] + 5
lon <- seq(-180, 180, length.out = 20)
lat <- seq(-90, 90, length.out = 10)

# mrbsizeRsphere analysis
mrbOut <- mrbsizeRsphere(posteriorFile = sampleData, mm = 20, nn = 10,
                         lambdaSmoother = c(0.1, 1), prob = 0.95)

# Posterior mean of the differences of smooths
plot(x = mrbOut$smMean, lon = lon, lat = lat,
     color = fields::tim.colors(), turnOut = FALSE)
```

---

rmvtDCT *Sampling from marginal posterior multivariate t-distribution.*

Description

Samples from a marginal posterior multivariate t-distribution with normal-inverse-chi-squared-prior are generated.

Usage

```
rmvtDCT(object, lambda, sigma, nu0, ns)
```

Arguments

object	Observed object, as matrix.
lambda	Scaling parameter (λ) of the normal-inverse-chi-squared-prior.
sigma	Square root of the σ_0^2 parameter of the normal-inverse-chi-squared-prior.
nu0	Degrees of freedom (ν_0) of the normal-inverse-chi-square-prior.
ns	Number of samples that should be generated.

Details

An eigenvalue decomposition is used for sampling. To speed up computations, a 2D discrete cosine transform (DCT) has been implemented, see dctMatrix. The output is a list containing

1. Samples of the marginal posterior of the input as column vectors.
2. The mean of the marginal posterior of the input as a vector.

Value

A list containing the following elements:

`sample` Samples of the marginal posterior of the input.

`mu` Mean of the marginal posterior of the input.

Examples

```
# Artificial sample data
set.seed(987)
sampleData <- matrix(stats::rnorm(100), nrow = 10)
sampleData[4:6, 6:8] <- sampleData[4:6, 6:8] + 5

# Sampling from a multivariate t-distribution
t_dist_samp <- rmvtDCT(object = sampleData, lambda = 1, sigma = 10,
                       nu0 = 50, ns = 1000)
```

`TaperingPlot` *Plot of tapering functions.*

Description

Tapering functions corresponding to the smoothing levels in `lambdaSmoother` are drawn. This plot helps to assess if the chosen smoothing levels are appropriate.

Usage

```
TaperingPlot(lambdaSmoother, mm, nn, Xmu, ...)
```

Arguments

`lambdaSmoother`	Vector consisting of the smoothing levels to be used.
`mm`	Number of rows of the original input object.
`nn`	Number of columns of the original input object.
`Xmu`	If availabe, posterior mean of the input object.
`...`	Further graphical parameters can be passed.

Details

The tapering functions of the smoothing levels chosen should be generally approximately disjoint. This will produce features which are somewhat orthogonal. With orthogonal features, it is likely that each difference of smooths corresponds to a different pattern in the input image.

Sometimes, not all patterns of the input image can be extracted using smoothing levels whose tapering functions are disjoint. It might then be necessary to include additional smoothing levels, and the disjointedness might not be satisfied anymore. The selection of appropriate smoothing levels with this method therefore requires some user interaction. Still, choosing disjoint tapering functions for finding appropriate smoothing levels is a good starting point.

Better results could be obtained if the structure of the posterior mean of the input is also taken into account. If the posterior mean is available, it can be added with the argument `Xmu` and moving averages of the absolute values of the signal-dependent tapering functions are drawn. `MinLambda` offers a more formal approach of optimizing the disjointedness of the tapering functions and can help finding appropriate smoothing levels when the input signal is taken into account.

Value

Plots of the tapering functions for all differences of smooths at neighboring scales are created.

Examples

```
# Signal-independent tapering function plot for a 30-by-10 object with
# the smoothing parameter sequence [0, 1, 10, 1000, inf]:

TaperingPlot(lambdaSmoother = c(1, 10, 1000), mm = 30, nn = 10)


# Signal-dependent tapering function plot for a 30-by-10 object with
# the smoothing parameter sequence [0, 1, 10, 1000, inf]:

set.seed(987)
xmuExample <- c(stats::rnorm(300))
TaperingPlot(lambdaSmoother = c(1, 10, 1000), mm = 30, nn = 10,
             Xmu = xmuExample)
```

---

tridiag *Generate a tridiagonal matrix.*

Description

Generate a tridiagonal matrix with `upperDiag` as superdiagonal, `lowerDiag` as subdiagonal and `mainDiag` as diagonal.

Usage

```
tridiag(mainDiag, upperDiag, lowerDiag)
```

Arguments

`mainDiag`	Diagonal of tridiagonal matrix.
`upperDiag`	Superdiagonal of tridiagonal matrix. Must have length `length(mainDiag) - 1`.
`lowerDiag`	Subdiagonal of tridiagonal matrix. Must have length `length(mainDiag) - 1`.

Value

Tridiagonal matrix.

Examples

```
set.seed(987)
mainDiag <- sample(100:110, size = 6, replace = TRUE)
upperDiag <- sample(10:20, size = 5, replace = TRUE)
lowerDiag <- sample(1:10, size = 5, replace = TRUE)

tridiag(mainDiag, upperDiag, lowerDiag)
```

`turnmat` *Turn matrix 90 degrees counter-clockwise.*

Description

Help function to turn matrix 90 degrees counter-clockwise. `turnmat` is used as an internal function in some of the plotting functions. Depending on how the data is stored, it can be necessary to turn the matrices 90 degrees counter-clockwise for being able to plot them correctly.

Usage

```
turnmat(x)
```

Arguments

`x`	Matrix to be turned.

Value

Matrix `x`, turned by 90 degrees counter-clockwise.

Examples

```
set.seed(987)
sampleMat <- matrix(stats::rnorm(100), nrow = 10)
sampleMatTurn <- turnmat(x = sampleMat)
```

Index