Wavelet Thresholding for Unequally Spaced Data

Arne Kovac

A thesis submitted to the University of Bristol
in accordance with the requirements
of the degree of Doctor of Philosophy
in the Faculty of Science

Supervisor: Prof. B. W. Silverman, FRS

March 1998

Abstract

We consider the problem of non-parametric regression. Given are noisy observations $y_1, \ldots, y_n$ at time points $t_1, \ldots, t_n$ which are modelled as $y_i = f(t_i) + \varepsilon_i$, where f is the function to be estimated and $\varepsilon_1, \ldots, \varepsilon_n$ i.i.d. Gaussian white Noise with mean 0 and variance σ^2. One approach to this problem is wavelet shrinkage, but it has the disadvantage that it requires the number of data points to be a power of two and the time points to be regularly spaced, i.e. $t_i = i/n$.

In the first chapter we give a short introduction to the thesis and discuss the problem of non-parametric regression as well as the problem with unequally spaced data within the approach of wavelet shrinkage. We review some material on wavelets, the discrete wavelet transform and non-parametric regression in the second chapter.

The third chapter deals with a fast $O(n)$ algorithm for calculating the variances of wavelet coefficients that are obtained by applying the discrete wavelet transform to correlated normal noise with mean 0 and covariance matrix Σ, where Σ is a band matrix. We use this algorithm to introduce a new method for thresholding unequally spaced data in the fourth chapter. We also review contributions of other authors to this topic, present an implementation of the new procedures in the software package WaveThresh which runs under S-Plus and apply the new method to two real data sets.

In the fifth chapter we analyse different threshold selections by calculating the exact mean square error for four test signals and different choices of parameters like different noise levels, hard and soft thresholding rules etc. In particular we analyse versions of Donoho and Johnstone's *VisuShrink* and *SureShrink*.

The sixth chapter deals with robust wavelet techniques which perform well even if the the noise is not normally distributed or if the data set is disturbed by several consecutive outliers. Two new algorithms are developed which make use of the new methods for thresholding unequally spaced data and calculating the variances of wavelet coefficients.

The last chapter summarises the results of this thesis and gives the prospects for possible future research.

Declaration

All the work in this thesis is my own, unless otherwise referenced. A large part has previously been submitted to the University of Bristol for the degree of Master of Science by Research, but permission was granted to withdraw candidature for the Master's degree and to resubmit the thesis for the degree of Doctor of Philosophy. Apart from this no part of this thesis has already been or is being submitted for any degree diploma or other qualification at any other university. The views expressed in this dissertation are my own and not of the University.

Acknowledgements

I would like to thank Laurie Davies for talking about my career and suggesting that I undertake graduate study in Bristol. Without him I would not have experienced this wonderful last year. I would also like to thank the DAAD for giving me a generous scholarship which facilitated my stay in Bristol enormously.

A large part of this thesis was previously submitted for the degree of Master of Science, and I am deeply indebted to the examiners, Fanis Sapatinas and Emma McCoy, for their generous assessment and detailed comments.

This final version was also heavily influenced by numerous suggestions of Guy Nason and Trevor Bailey who examined this thesis for the degree of Doctor of Philosophy and I am most thankful for that.

Finally, I want to thank Bernard Silverman for all the support I got during this year and everything I learnt from him. One cannot imagine a better adviser.

Contents

1 Introduction **11**

2 Non-parametric regression with wavelets **18**

2.1 Wavelets 19

2.2 Multiresolution Analysis 25

2.3 The Discrete Wavelet Transform 29

2.4 Non-parametric regression 37

2.4.1 Kernel estimators 38

2.4.2 Local polynomial regression 39

2.4.3 Spline smoothing 40

2.4.4 Fourier series estimators 41

2.4.5 Summary 42

2.5 Wavelet regression 45

3 Calculation of the variances of wavelet coefficients **54**

3.1 An algorithm for computing the variances 55

3.2 Computational complexity in the periodic case 58

3.3 Computational complexity in the symmetric case 61

3.4 Summary 65

4 Thresholding unequally time-spaced data **66**

4.1 Wavelet regression with unequally spaced data 67

4.1.1 Retouching wavelet coefficients 68

4.1.2 Kernel-like wavelet regression 69

4.1.3 Polynomial regression 71

4.2 A new algorithm for thresholding unequally spaced data 72

4.3 Computation of the factors for the variances 76

4.4 An adaptation of SureShrink 79

4.5 An implementation in WaveThresh 81

4.6 Application to real data sets 84

5 Thresholding and the Mean Square Error **89**

5.1 Calculating the risk 90

5.2 Regular grids of arbitrary length 93

5.3 Optimal coefficient-dependent thresholds 99

5.4 Optimal noise-proportional thresholds 106

5.5 Comparison between optimal thresholds and VisuShrink 111

5.6 Global thresholds vs. coefficient-dependent thresholds 118

5.7 Thresholds that are proportional to a power of the noise level 121

5.8 How good is hard thresholding? 129

5.9 Final comparison 131

5.10 Summary 134

6 Robust Wavelet Thresholding **136**

6.1 Wavelet Thresholding and robustness 137

6.2 Robust wavelet methods for data on a regular grid 142

6.3 Median filters 145

6.4 Removing outliers of irregularly spaced data 148

6.5 Another approach for removing outliers 155

6.6 The choice of the outlier thresholds 159

6.7 Wavelet thresholding of median-filtered data 170

6.8 Examples 176

6.9 Summary 182

7 Conclusions and suggestions for future research **185**

7.1 Summary 185

7.2 Suggestions for future research 187

Bibliography **189**

List of Figures

2.1 The Haar function 24

2.2 Schematic representation of The Discrete Wavelet Transform 33

2.3 Application of four linear estimators to the Doppler data with a large amount of smoothing. 43

2.4 Another application of the estimators to the Doppler data, but with a small amount of smoothing. 44

2.5 Wavelet Shrinkage applied to the Doppler data 53

4.1 A simple example for the grid transform 82

4.2 Example for the variance factors 83

4.3 Another representation of the variance factors 84

4.4 Waiting time between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park. 86

4.5 Measurement of exhaust from burning ethanol 87

5.1 Donoho and Johnstone's test signals 93

5.2 The risk for the test functions and *VisuShrink* on regular grids 95

5.3 The risk for the Doppler data with *VisuShrink* compared with a fixed threshold choice on regular grids 97

5.4 The true risk function for the Doppler signal and the difference from a simulated version 99

5.5 A simple example for a risk function with two local minima and its first derivative 107

5.6 The risk for the Heavisine data and noise level 0.35 for three threshold choices on a regular grid of size n 112

5.7 The expected MSE for DJ's test signals and noise level 0.35 for three threshold choices on irregular time structures of size n 115

5.8 The differences between the risk for optimal global thresholds and optimal noise-proportional thresholds for the various test data 119

5.9 Wavelet coefficients separated by their risk behaviour 122

5.10 Wavelet coefficients separated by their risk behaviour 123

5.11 Mean Square Error for the Doppler data, irregular spaced time points and thresholds proportional to the s-th power of the local noise level with optimal factor 124

5.12 The MSE functions in a simple example for each coefficient. 125

5.13 The MSE function for a wavelet coefficient with another parameter choice. 126

5.14 Wavelet Decomposition for 2048 regular spaced points of the Doppler data 127

5.15 Noise level of each wavelet coefficient in a wavelet decomposition. 128

6.1 A sine curve is disturbed by Cauchy noise along with two attempts for reconstructing it 139

6.2 Haerdle's crash data with several wavelet estimates 149

6.3 Numbers of detected outliers when applying algorithm 6.1 to the various test signals and to pure noise 153
6.4 Number of detected outliers when applying algorithm 6.3 to pure noise for different parameter choices 169
6.5 Four robust estimates for the crash data 177
6.6 The balloon data set 178
6.7 Four robust estimates for the balloon data 180
6.8 Four robust estimates for a sine curve which is disturbed by Cauchy noise 181

List of Tables

2.1 Mean squared error for the Doppler data, four linear smoothing methods and wavelet thresholding 52

5.1 Calculation of the Mean Square Error with formula and simulation for the Doppler data and several equally spaced time sets 99

5.2 Simulated mean squared error for different sizes of the grid 117

5.3 The minimal expected mean square error for an artificial data set and different threshold models 127

5.4 Comparison of the optimal Mean Square Error 130

5.5 Comparison of the performance of several threshold models 131

5.6 Comparison of the average Mean Squared Error for five different thresholding techniques. 133

6.1 Thresholds for algorithm 6.1 and the simplified version of algorithm 6.3 . 164

6.2 Thresholds for algorithm 6.3 165

6.3 Further outlier threshold values 168

Chapter 1

Introduction

"The beginning is the most important part of the work."

—Plato, *The Republic. Book II. 377B*

One of the classical problems of statistics is non-parametric regression: A signal f is to be estimated, but only noisy observations $y_1, \ldots, y_n$ at time points $t_1, \ldots, t_n$ are accessible. We do not have any knowledge about f (except, perhaps, that we assume that f is a smooth function), and therefore well-known techniques like linear regression are not suitable in this situation.

A lot of work in statistics has been done in this field (see [43] and [56]), and we will see some of the results in the second chapter of this thesis. It is interesting to see the influence that the rapid developments in computer technology during the last 20 years have had on the methods that were suggested in this area. Some of the later developments have been stimulated by the increasing availability and speed of computers.

However, even today it is still very important to construct fast algorithms for estimating a function in non-parametric regression, because not only has the speed of computers increased—the size of the data sets has grown as well, and it is certainly not unrealistic to assume that they will continue to grow in the face of the improving accuracy of measuring instruments as well as the decreasing price of storage media. Moreover, techniques for non-parametric regression are used in other statistical areas like functional data analysis and projection pursuit where the analysis of one data set typically requires several estimates of a function f.

Note that if we say that an algorithm is "fast", then the absolute time that it needs on a specific machine to complete is not the only consideration. We are more interested in the time as a function of the size of the data set n. Optimal in this sense would be a method that needs twice as long if we double the number of data points, so that the computing time is proportional to n. Such a method is called an *algorithm of order* n. Also acceptable would be a procedure of order $n \log(n)$, because the $\log$ factor is negligible for large n. Algorithms of order n^2 or even larger may still be useful for a number of applications, but they are usually not what we want to understand by "fast".

At the beginning of this decade Donoho and Johnstone suggested "Wavelet Shrinkage" as a new method for estimating a function in non-parametric regression (see for example [41], further references are given below). This approach was made possible by developments in the field of functional analysis and engineering (see Daubechies' monograph [27] and other references given below) which we review in the second chapter of this thesis. Wavelet shrinkage has a lot of advantages compared with other techniques: It is of order n, it is easy to perform and it has a lot of good statistical properties (see [41]).

The basic outline of the method is as follows: First the data $(y_i)_{i=1}^n$ are transformed to a different representation with the "Discrete Wavelet Transform" (DWT), which means essentially that they are multiplied by an orthogonal matrix $\mathcal{W}$. The remarkable fact about the DWT is that the multiplication can be accomplished in order n time by using special properties of the matrix $\mathcal{W}$. The elements of the vector that is obtained by the application of the DWT are called *wavelet coefficients*. They are modified in the second step of the algorithm, and afterwards the inverse of $\mathcal{W}$ is applied to the modified coefficients to yield a vector that is regarded as an estimate of f at the given time points ($\hat{f}$ is then estimated by interpolating these values).

A lot of work has been invested in the second step of the procedure. We will review in the second chapter some ways for modifying the wavelet coefficients. The most popular methods are “thresholding” routines. In this case every coefficient that is smaller than a threshold t is set to zero, while the others are kept or shrunk.

There is, however, still a serious disadvantage of wavelet shrinkage that we have concealed so far: The application of the DWT requires that the number of data points is a power of two and that the time points are equally spaced. The first requirement is usually met by extending the given data set to length 2^J using for example periodic extension or symmetric reflection (see [102]). Although it looks very artificial, this simple solution leads usually to good results. Note that the DWT makes use of boundary conditions anyway—this is discussed in the second chapter in some more detail.

The second condition raises more problems. One way could be to ignore the time structure and apply usual wavelet shrinkage to the y_i. This would give us an estimate of f at the time points t_i. An estimate on a regular grid (which is desirable for time data with gaps inside) could be obtained by interpolating the result. Lenarduzzi has noticed (see [75]) that this idea does not lead to “visually pleasant results”, and we will see more disadvantages later. She also suggests an algorithm for this situation that we introduce in Chapter 4 among contributions of other authors to this topic (see [6] and [32]).

We discuss in this thesis a new approach for dealing with unequally spaced data. The idea is very simple: First we construct from the given data a set of 2^J grid observations, where J is the required resolution level.[1] This is done by connecting the data points with straight lines and evaluating the lines at the new grid points. The number of new "observations" is now a power of two, so we can apply the DWT and obtain wavelet coefficients w_{jk}.

Before we continue with the description of our algorithm it is necessary to mention that it seems sensible for a lot of applications to assume that the noise is uncorrelated normal with zero mean and variance σ^2. As we will see in the second chapter, wavelet thresholding takes advantage of the fact that applying the DWT to the noise yields coefficients that are also uncorrelated normal distributed with mean 0 and variance σ^2, since performing the DWT means multiplying with an orthogonal matrix. One of these advantages is that *global* thresholds can be used, so that the same threshold is used for each coefficient.

However, in our situation this is no longer valid because the noise of the gridded observations is no longer uncorrelated, and it is easy to calculate their covariance matrix: If we denote the linear function that maps the original data to the grid observations by R, then the covariance matrix is $\Sigma = \sigma^2 \cdot RR^T$. This matrix is a band matrix because each new observation is computed from only two neighboured original data points, so it is not surprising that Σ can be calculated rapidly, and we do this in Chapter 4. As we are even more interested in the variances of the wavelet coefficients w_{jk}, we want to calculate the values of $\mathcal{W}\Sigma\mathcal{W}^T$, the covariance matrix of the coefficients, or at least its main diagonal.

[1] For a lot of situations it seems reasonable to choose 2^J as the smallest power of two that is not smaller than n.

One obvious way would be to actually carry out the matrix multiplications, but this would be too slow. We propose in Chapter 3 a more sophisticated algorithm for doing this. It makes use of a special property of Σ: Remember that each grid observation depends only on the value of its left and right neighbour among the data. Therefore, Σ is a band matrix, and this property makes it possible to compute the variances of the wavelet coefficients in a fast way (see Chapter 3). This is what we do in step two of our algorithm.

Afterwards, we modify the coefficients w_{jk} in a similar way as in the usual setting where the time points are equally spaced, but we determine from the variances *coefficient-dependent* thresholds. Finally we invert the DWT to obtain an estimate of the unknown signal f.

The whole procedure was implemented in the wavelet package *WaveThresh* (available from the Statlib archive) which runs under S-PLUS and R. While the well-known commercial product S-Plus should not need any further explanation, it seems to be necessary to remark that R (see [63]) is a software package that tries to implement most of the functionality from S. But it is developed under a GNU copyright, and is therefore freely available from the Statlib archive. The implementation was used to apply the new methods to two real data sets. The new functions can be downloaded from the web page

`http://feat.mathematik.uni-essen.de/~kovac/thesis.html`

which supplements this thesis.

By calculating the exact formula for the expected mean square error (or *risk*) for four different signals, we carry out an extensive risk analysis for our new thresholding procedure. We compute optimal thresholds that minimise the risk and compare them with different threshold choices. We also compare our method with the "classical" method that was mentioned above where the DWT is applied directly to the given data, ignoring the unequally spaced time points.

An important application of our algorithm (and other methods for dealing with unequally spaced data as well) is robust wavelet regression. Suppose that some of the observations are contaminated by outliers. Then it is seems reasonable to remove the outliers and apply a regression technique on the remainder like our new algorithm. We consider several techniques for recognising outliers (see Chapter 6) and show the result for two specific examples. A different idea for robust wavelet thresholding is a two-stage algorithm where first the data are filtered with running medians, then wavelet thresholding is applied to the result. As usual, the interesting question is which thresholds should be used, and we will again suggest a threshold choice which is based on the variances of the wavelet coefficients so that the algorithm for calculating the variances turns out once more to be very useful.

So the structure of the thesis is as follows: We review some material on wavelets and non-parametric regression in Chapter 2. In Chapter 3 we introduce a fast algorithm for computing the covariance matrix of wavelet coefficients that are obtained by applying the discrete wavelet transform to multivariate normal data with mean 0 and covariance matrix Σ, where Σ is a band matrix.

Chapter 4 is mainly concerned with thresholding unequally spaced data. First, some work of other authors is reviewed and a new method based on the results of Chapter 3 is proposed. Then a software implementation is described and illustrated with examples, and finally the new thresholding technique is applied to two data sets.

In Chapter 5 the new thresholding algorithm is examined with respect to the mean square error. Various thresholding choices are compared by calculating the exact risk and assuming that the noise of the original observations is uncorrelated normal noise. We also present a short simulation study where we compare our estimator with a more classical approach.

Robust wavelet regression is treated in Chapter 6 where two different ideas are introduced: First we discuss several approaches where outliers are removed and the new thresholding algorithm is applied to the cleaned data, then we discuss the combination of median filters with wavelet thresholding. All techniques are applied to two real data sets.

We finish the thesis in Chapter 7 with a short summary and give the prospects for possible further research.

Chapter 2

Non-parametric regression with wavelets

"Mang Shih She also had a method of developing his courage.

He said: 'I regard victory and defeat as the same.

To gauge the enemy and then attack; to plan the victory

and then engage—this is to be afraid of the opposing army.

How can I be sure of winning? I can only be fearless, and that's all."

—Confucius, *Mencius, 2A:2*

We have already mentioned in the introductory chapter the statistical problem of non-parametric regression where one tries to estimate a function from noisy data. Many approaches to the solution of this problem can be found in the statistical literature. Although we mention several of these techniques, we focus on one specific approach called "Wavelet Shrinkage".

In order to give a proper explanation of this method it is necessary to review some basic facts about Hilbert spaces and orthonormal bases. Then we define what a wavelet is, and it turns out that wavelets form orthonormal bases of the function space $L^2(\mathbb{R})$ with some special properties.

We proceed by introducing the notion of a multiresolution analysis, and we will see that this concept leads to a better understanding of wavelets in general and another nice property of wavelet bases which is used in the subsequent subsection. There we are concerned with the discrete wavelet transform which serves as the link between wavelets in Mathematics on the one hand and applications of wavelets on the other hand, because in practice, one usually deals with discrete data sets and not functions.

Before we explain how the discrete wavelet transform can be applied to noisy data, we first introduce again the problem of non-parametric regression. We also discuss briefly, three methods: kernel estimators, spline smoothing and Fourier estimators. We demonstrate with an example a typical problem that these estimators share: They cannot at the same time reduce the noise and retain local features.

The last subsection deals with wavelet shrinkage which is another approach for estimating a function from noisy data and based on the discrete wavelet transform. We introduce the basic idea and review some ideas for choosing the threshold which is a parameter that determines the smoothness of the regression function.

2.1 Wavelets

Wavelets are nowadays used in a variety of scientific areas. They are involved in mathematical fields like functional analysis, ordinary and partial differential equations, numerical analysis and statistics, in natural sciences such as physics and chemistry, and further applications can even be found in economics.

Among the applications are image compression (which is used by the American FBI for digitally storing of finger prints, see [125] and [13]), image processing (recognising edges), predicting of earthquakes (see [109]), voice recognition and synthetic generation of sound (see [124]) and signal detection in underwater sounds (see [7] and [90]).

In the following we try to explain what wavelets are and how they are defined. Nevertheless, this is not meant as an introductory text to wavelets. More detailed representations can be found in a nice paper by Strang (see [106]) and a book by Hubbard (see [61]). Mathematically more sophisticated are the three monographs of Chui (see [22]), Daubechies (see [27]) and Meyer (see [81]) which serve as standard references throughout the wavelet literature.

Before introducing wavelets we recall some basic facts about bases in normed vector spaces. In a finite-dimensional vector space V, a *basis* is a set $\{e_j\}_{1\leq j\leq n}$ such that each element $v \in V$ can be written in one and only one way as a sum $v = \sum_{j=1}^{n} s_j e_j$ for some numbers s_j. One of the advantages of having a basis is that a linear transformation A from V into itself corresponds to a matrix $\{A_{j,k}\}$ with $A(e_j) = \sum_{1\leq k\leq n} A_{j,k} e_k$ for each j. In general, bases in finite-dimensional spaces are important. Therefore, it is natural to try to extend the idea of a basis to infinite-dimensional spaces.

But it turns out to be hard to keep all the good properties of bases of finite-dimensional spaces. In any vector space V, a *Hamel basis* is a set $\{e_\alpha\}_{\alpha\in I}$ such that every $v \in V$ can be written in one and only one way as $v = \sum_{\alpha\in I} s_\alpha e_\alpha$ where only finitely many of the numbers s_α are not 0. For a finite-dimensional space, a Hamel basis is just a basis, but in an infinite-dimensional Banach space, there are no obvious Hamel bases, although they can be shown to exist with the axiom of choice. Another problem is that in analysis one usually wants to deal with converging infinite sums, so the notion of Hamel basis, which is purely algebraic, is usually not appropriate.

In a Banach space $(V, ||\cdot||)$, an *unconditional basis* is a collection $\{e_\alpha\}_{\alpha\in I}$ such that for each $v \in V$ there is a unique set of numbers $\{s_\alpha\}_{\alpha\in I}$ such that $\sum_{\alpha\in I} s_\alpha e_\alpha = v$ (converging in $||\cdot||$). In general, it may be hard to find unconditional bases. However, everything becomes much easier if V is a Hilbert space.

It can be shown (see [95] and [42]) that every Hilbert space $(V, (\cdot,\cdot))$ has an *orthonormal basis*. This is an orthonormal set $\{e_\alpha\}_{\alpha\in I}$ such that each $v \in V$ can be represented as $v = \sum_{\alpha\in I}(v, e_\alpha)e_\alpha$.

One example of a Hilbert space is $L^2([0, 2\pi))$. Roughly speaking [1] this is the space of all functions f such that the integral

$$\int_0^{2\pi} f(t)^2 dt$$

is finite. One famous orthonormal basis of $L^2([0, 2\pi))$ is the well known *Fourier basis* $\{\psi_n\}$, where the ψ_n are the Fourier functions

$$\frac{1}{2\pi}, \frac{1}{\pi}\cos(\cdot), \frac{1}{\pi}\sin(\cdot), \frac{1}{\pi}\cos(2\cdot), \frac{1}{\pi}\sin(2\cdot), \frac{1}{\pi}\cos(3\cdot), \ldots.$$

So each $f \in L^2([0, 2\pi))$ can be written as

$$f = \sum_{n=0}^{\infty} c_n\psi_n,$$

where the Fourier coefficients c_n are calculated as

$$c_n = \int_0^{2\pi} f(t)\psi_n(t)dt.$$

[1] More precisely, this defines the $\mathcal{L}^2([0, 2\pi))$ which is not complete. The Hilbert space $L^2([0, 2\pi))$ is obtained by considering equivalence classes: If Z is defined by

$$Z = \{f| \int_0^{2\pi} f^2(t)dt = 0, \}$$

then $L^2([0, 2\pi))$ is defined as the quotient space $\mathcal{L}^2([0, 2\pi))/Z$.

The convergence takes place in $L^2([0, 2\pi))$, so that

$$\lim_{N\to\infty} \int_0^{2\pi} (f(t) - \sum_{n=0}^{N} c_n \psi_n(t))^2 dt = 0.$$

Suppose now, we want to approximate a given function f in terms of its Fourier series. Naturally, we hope that only a small number of coefficients is clearly different from zero. There are certain smooth functions that have such an "economical" Fourier expansion. For example, it is possible to prove (see [10]) that the coefficients c_n of an infinitely differentiable function decay faster as $n \to \infty$ than $1/n^r$ for each $r \in \mathbb{N}$.

However, in general Fourier expansions are not economical. One reason is that there is no localisation in time—each coefficient depends on values of the function across the whole interval $[0, 2\pi)$. Consequently, a discontinuity in f affects all the coefficients. Typically, discontinuities lead to coefficients of order $1/n$ for $n \to \infty$. Therefore, a large number of terms are required to approximate discontinuous functions accurately.

We will see later that wavelets have substantial advantages, so let us introduce them now. Everything is based on a function ψ called the *mother wavelet.* For each $m \geq 0$ we define [2]

Definition 2.1. *A function $\psi : \mathbb{R} \to \mathbb{R}$ is called a mother wavelet of order m if the following five properties are satisfied:*

$(\psi 1)$ *If $m > 1$ then ψ is $(m-1)$-times differentiable.*

$(\psi 2)$ *$\psi \in L^\infty(\mathbb{R})$. If $m > 1$, for each $j \in \{1, \ldots, m-1\}$ $\psi^{(j)} \in L^\infty(\mathbb{R})$.*

$(\psi 3)$ *ψ and all its derivatives up to order $m-1$ decay rapidly: For each $r > 0$ there is*

[2] This definition is due to Y. Meyer (see [81]). Other authors often have slightly different definitions.

a $\gamma > 0$ such that

$$|\psi^{(j)}(t)| < 1/t^r \quad j \in \{0, 1, \ldots, m-1\}$$

for each $|t| > \gamma$.

($\psi 4$) *For each $j \in \{0, \ldots, m\}$ we have $\int t^j \psi(t)dt = 0$.*

($\psi 5$) *The set $\{\psi_{j,k}\}_{j,k \in \mathbb{Z}}$ is an orthonormal basis of $L^2(\mathbb{R})$, where the $\psi_{j,k}$ are derived from the mother wavelet by the relationship*

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k).$$

As a simple example of a wavelet we consider the *Haar basis* which is generated from the mother wavelet of order 0 (see Figure 2.1).

$$\psi(x) = \begin{cases} 1 & 0 \le x < \frac{1}{2}, \\ -1 & \frac{1}{2} \le x < 1 \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

Obviously, the properties ($\psi 1$), ($\psi 2$), ($\psi 3$) and ($\psi 4$) are satisfied. Property ($\psi 5$) requires somewhat more effort and will be shown in the next section. However, it is not too difficult to see that the $\{\psi_{j,k}\}$ are mutually orthonormal.

1. Two different wavelets on the same scale do not have intersecting supports. Therefore

   $$< \psi_{j,k_1}, \psi_{j,k_2} >= 0$$

   for all $j \in \mathbb{Z}$ and $k_1, k_2 \in \mathbb{Z}$ with $k_1 \neq k_2$.

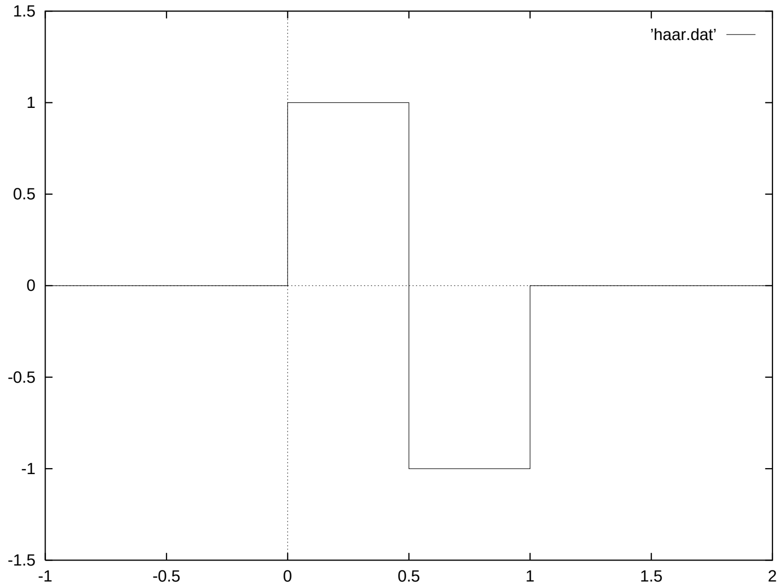

Figure 2.1: The Haar function

2. Wavelets on different scales either have non-intersecting support or if they do, then the support of one wavelet lies in a set where the other wavelet is constant. Therefore

$$< \psi_{j_1,k_1}, \psi_{j_2,k_2} >= 0$$

for all $j_1, j_2 \in \mathbb{Z}$ and $k_1, k_2 \in \mathbb{Z}$ with $j_1 \neq j_2$.

3. Finally we have for all $j, k \in \mathbb{Z}$ that

$$< \psi_{j,k}, \psi_{j,k} >= 1.$$

It remains to show that $\overline{\mathrm{span}\{\psi_{j,k}\}} = L^2(\mathbb{R})$. This is done in the next section.

The Haar basis has been known for a long time. Haar mentioned it in 1910 in an appendix to an article (see [49]). Levy used it in the 1930's when he examined Brownian motion. But essential for the wavelet theory was the observation that there are many other wavelets, generated from smoother mother wavelet functions. In 1982, Strömberg was the first to construct orthonormal bases of arbitrary order (see [108]). Meyer discovered in 1985 a basis with infinitely differentiable functions (see [80]) and in 1988, Daubechies (see [26]) constructed her famous Daubechies wavelets which have finite support (and of course, arbitrarily high order). We will see later that wavelets with compact support are particularly useful for dealing with discrete data.

Let us now take a further look at the wavelet coefficients: Every coefficient $f_{j,k}$ depends only locally on f, because

$$f_{j,k} = \int_{-\infty}^{\infty} f(t)\psi_{j,k}(t)dt$$

and the wavelet $\psi_{j,k}$ has finite (or at least essentially finite) support. Consequently, $f_{j,k}$ gives information on "scale" 2^j near position $2^{-j}k$. This is a difference from the Fourier transform where every Fourier coefficient was calculated from the values of the interval $[0, 2\pi)$, and leads to more economical representations for much wider classes of smooth functions, including those that are smooth between discontinuities.

2.2 Multiresolution Analysis

In this section we introduce the concept of *Multiresolution Analysis.* There are two reasons for doing this. The first one is that it has a very important application, the Discrete Wavelet Transform, which we will define in the next section. The other is that the concept leads to a better understanding of wavelets. In particular we will complete the proof that the Haar wavelet is a mother wavelet. We only sketch the most important results and refer to Daubechies (see [27]) and Jawerth and Sweldens (see [66]) for more details.

The following definition of Multiresolution Analysis was introduced by Mallat and Meyer in 1986 (see [77] and [27], p. 129).

Definition 2.2. *A family $(V_j)_{j\in\mathbb{Z}}$ of closed subspaces of $L^2(\mathbb{R})$ is called a Multiresolution Analysis, if the following five properties are satisfied:*

$(M1)$ $V_j \subset V_{j+1}$

$(M2)$ $v(x) \in V_j \Leftrightarrow v(2x) \in V_{j+1}$

$(M3)$ $v(x) \in V_0 \Leftrightarrow v(x+1) \in V_0$

$(M4)$ $\bigcup_{j\in\mathbb{Z}} V_j$ *is dense in* $L^2(\mathbb{R})$ *and* $\bigcap_{j\in\mathbb{Z}} V_j = \{0\}$

$(M5)$ *There is a function $\phi \in V_0$ such that the family $\{\phi(x-k) : k \in \mathbb{Z}\}$ is an orthonormal basis of V_0.*

The subspaces V_j are called levels. One level is said to be *coarser* (or *finer*) than another level, if its index is less (or greater) than the other index. The function ϕ from $(M5)$ is called the *scaling function* or *father wavelet.*

We proceed with a simple example which will demonstrate how a wavelet basis can be constructed from a multiresolution analysis. In particular, the example finishes the proof of the statement that the Haar wavelet is a mother wavelet.

Consider the family $\{V_j\}_{j\in\mathbb{Z}}$

$$V_j = \{f \in L^2(\mathbb{R}) : f \text{ is constant on } [2^{-j}k, 2^{-j}(k+1)) \text{ for all } k \in \mathbb{Z}\}. \tag{2.2}$$

Obviously, the V_j are closed and $(M1), (M2)$ and $(M3)$ are satisfied. The second part of property $(M4)$ is also obvious and the first part follows because every function $f \in L^2(\mathbb{R})$ can be arbitrarily well approximated by a function with compact support which is piecewise constant on the $[2^{-j}k, 2^{-j}(k+1))$ (it suffices to take the support and j large enough). The orthogonal projection from $L^2(\mathbb{R})$ to V_j is given by

$$P_j(f) = 2^j \sum_{k\in\mathbb{Z}} \int_{2^{-j}k}^{2^{-j}(k+1)} f(x)dx \cdot \chi_{[2^{-j}k, 2^{-j}(k+1))}.$$

Finally, we define ϕ to be the indicator function on $[0,1)$ which satisfies property $(M5)$.

$$\phi(x) = \begin{cases} 1 & 0 \leq x < 1 \\ \\ 0 & \text{otherwise.} \end{cases}$$

Now we define: (similarly to the definition of the mother wavelet translations and dilations of the scaling function)

$$\phi_{j,k}(t) = 2^{j/2}\phi(2^j t - k). \tag{2.3}$$

Then we have

$$< f, \phi_{j,k} >= 2^{j/2} \int_{2^{-j}k}^{2^{-j}(k+1)} f(x)dx$$

and consequently

$$P_j f = \sum_{k\in\mathbb{Z}} < f, \phi_{j,k} > \phi_{j,k}.$$

Let ψ be the Haar function as defined in (2.1). Then it follows at once that

$$\psi(x) = \phi(2x) - \phi(2x-1).$$

Moreover, after some straightforward calculations (see [27] for details) we obtain that the difference $Q_j f$ between two successive approximations $P_j f$ and $P_{j+1} f$ is given by

$$Q_j f = P_{j+1} f - P_j f = \sum_{k\in\mathbb{Z}} < f, \psi_{j,k} > \psi_{j,k}. \tag{2.4}$$

We have already seen that the $\psi_{j,k}$ are mutually orthogonal. Consequently, the preceding equations show that $Q_j f$ can be expanded in terms of an orthonormal system. So, $Q_j f$ is the orthogonal projection from $L^2(\mathbb{R})$ to $W_j = P_{j+1}L^2 - P_jL^2$ which is the orthogonal complement of V_j in V_{j+1}:

$$V_{j+1} = V_j \oplus W_j, \qquad W_j \perp V_j.$$

Therefore, the W_j are mutually orthogonal and their direct sum is $L^2(\mathbb{R})$:

$$L^2(\mathbb{R}) = \oplus_{j\in\mathbb{Z}} W_j.$$

Finally, the family $\{\psi_{j,k}\}_{k\in\mathbb{Z}}$ is an orthonormal basis of W_j for each j. Thus, the whole family $\{\psi_{j,k}\}_{k\in\mathbb{Z}}$ forms an orthonormal basis of $L^2(\mathbb{R})$.

In the preceding example, a wavelet basis was constructed from a multiresolution analysis. It turns out that the example can be generalised to generate other wavelet bases, in particular of a higher order. Suppose that $(V_j)_{j\in\mathbb{Z}}$ is a multiresolution analysis. Again, we define W_j to be the orthonormal complement of V_j in V_{j+1}:

$$V_{j+1} = V_j \oplus W_j, \qquad W_j \perp V_j.$$

It follows immediately[3] that the spaces W_j are scaled versions of W_0:

$$f \in W_j \Leftrightarrow f(2^{-j}\cdot) \in W_0,$$

and that the W_j are mutually orthogonal, so that their direct sum is $L^2(\mathbb{R})$,

$$L^2(\mathbb{R}) = \oplus_{j\in\mathbb{Z}} W_j.$$

To generate a wavelet basis from the given multiresolution analysis we have to find a function $\psi \in W_0$ whose integer translations represent an orthonormal basis of W_0. Since

[3]It is clear from the definition of W_j that a function f lies in W_j if and only if $f \in V_{j+1}$ and if $f \perp V_j$. Property (M2) of a multiresolution analysis implies that this is equivalent to the condition that $f(\cdot/2) \in V_j$ and $f(\cdot/2) \perp V_{j-1}$ which simply means $f(\cdot/2) \in W_{j-1}$. Using induction we see that $f \in W_j \leftrightarrow f(2^{-j}\cdot) \in W_0$.

$\phi \in V_0 \subset V_1 = \overline{\text{span}\{\phi(2 \cdot -k)\}}$, there is a sequence $\{c_k\} \in l^2(\mathbb{Z})$ such that

$$\phi(x) = \sum_{k \in \mathbb{Z}} c_k \phi(2x - k). \tag{2.5}$$

If we now define

$$\psi(x) = \sum_{k \in \mathbb{Z}} (-1)^k c_{2-k} \phi(2x - k), \tag{2.6}$$

then it is possible to show (see [27], [22] or [81]) that the corresponding $\psi_{0,k}$ form an orthonormal basis of W_0. In other words, whenever we have a father wavelet ϕ that generates a multiresolution analysis we can easily construct a mother wavelet ψ from it. Yves Meyer is right when he remarks (see [81]) that "the mother wavelet is the daughter of the father wavelet ... "

2.3 The Discrete Wavelet Transform

The last sections were mainly concerned with function spaces, sequences of function spaces and orthonormal bases. However, most applications of wavelets in statistics deal with discrete data, so we need a discrete version of the wavelet transform. We will define the *Discrete Wavelet Transform* (DWT) which is a tool that operates on finite data sets, and it turns out that it is *based* on the usual wavelet transform and *linked* with the framework of function spaces through the concept of multiresolution analysis.

So let us consider now the case where the function f is not known in its entirety, but some values of f on a regular grid are given. The DWT can be written as a linear transformation where an orthogonal matrix $\mathcal{W}$ is used which is derived from the mother wavelet. The DWT of data $y_1, \ldots, y_n$ is defined as

$$w = \mathcal{W}y$$

and the wavelet coefficients that are obtained by this formula are often called *empirical coefficients*. On first sight, the necessary calculations are carried out in $O(n^2)$ operations, but in the case where n is a power of two, Mallat (see [78]) provided an algorithm which is of order n.

Computer implementations are easily available. The source code for the wavelet functions in "Numerical Recipes" (see [92]) can be downloaded as public domain software. [4] The DWT is part of two statistical wavelet packages, WaveThresh [5] and WaveLab, [6] that can be obtained freely as well. Finally, there is a commercial product S+Wavelets. [7]

We assume now that $n = 2^J$ for some integer J. Furthermore, we assume that our wavelet basis is derived from a multiresolution analysis, so that there is a father wavelet ϕ and $\{h_i\} \in l^2(\mathbb{Z})$ such that

$$\phi(x) = \sum_k h_k \phi(2x - k). \tag{2.7}$$

As we have seen in the last section, the corresponding mother wavelet is then given by

$$\psi(x) = \sum_k g_k \phi(2x - k), \tag{2.8}$$

where

$$g_k = (-1)^k h_{1-k}.$$

The idea behind the DWT can be summarised as follows: Let $\phi_{j,k}$ be defined as in the previous section and $f \in L^2(\mathbb{R})$. Then we have

$$< f, \phi_{j,k} >= \sum_k h_k < f, \phi_{j+1,k} >$$

[4] The URL is `http://nr.harvard.edu/nr/public-domain.html`

[5] WaveThresh is developed by Guy Nason as an add-on package for S-Plus (see [85]) and available from the Statlib archive.

[6] WaveLab is developed at the University of Stanford (see [17]), can be downloaded from `http://playfair.stanford.edu/~wavelab`, and runs under MatLab.

[7] S+Wavelets is a statistical package from StatSci Division, Mathsoft (see [16] for further details).

and

$$< f, \psi_{j,k} >= \sum_k g_k < f, \phi_{j+1,k} > .$$

Thus, once we have calculated all the coefficients $< f, \phi_{J,k} >$ for a level $J \in \mathbb{Z}$, we can compute the coefficients $< f, \phi_{j,k} >$ and $< f, \psi_{j,k} >$ for all $j < J$ and $k \in \mathbb{Z}$ without evaluating the integrals. On the other hand, the coefficients $< f, \phi_{J,k} >$ are weighted and scaled averages of f on a neighbourhood of $2^{-J}k$, and the neighbourhood is small, if J is large. Consequently, it makes sense to replace these coefficients by the (scaled) value of the function. [8] Note that the empirical coefficients that are obtained by applying this method to samples of a function f are usually different from the wavelet coefficients of the wavelet transform of f. In general, the more smooth and regular the function f, the better the agreement. However, Donoho (see [33]) has constructed a "hybrid wavelet transform" which is very similar to the wavelet transform but with the property that the empirical wavelet coefficients are exactly the first n theoretical coefficients of f in the new transform.

We now go into some more detail and define a *linear filter:*

Definition 2.3. *A linear filter $\mathcal{F}$ is a continuous mapping $\mathcal{F} : l^2(\mathbb{Z}) \to l^2(\mathbb{Z})$ whose action on a doubly infinite sequence $(x_i)_{i\in\mathbb{Z}} \in l^2(\mathbb{R})$ is defined by*

$$(\mathcal{F}x)_k = \sum_i f_{i-k} x_i, \tag{2.9}$$

where $(f_i)_{i\in\mathbb{Z}} \in l^2(\mathbb{Z})$.

If x is a vector of finite length, then the definition of $\mathcal{F}x$ depends on a choice of treatment at the boundaries. Common are periodic and symmetric boundary conditions.

[8] There are other suggestions for choosing the start values of the discrete wavelet transform. Sweldens and Piessens have discussed and compared several techniques, in particular quadrature formulae (see [110],[111] and [112]).

The DWT is based on filters $\mathcal{G}$ and $\mathcal{H}$ and on a "binary decimation" operator $\mathcal{D}_0$. The filters $\mathcal{G}$ and $\mathcal{H}$ are defined by the coefficients $(g_k)_{k\in\mathbb{Z}}$ and $(h_k)_{k\in\mathbb{Z}}$ that were derived from the multiresolution analysis. We now see the advantage of Daubechies' wavelets: In this case, both the mother and the father wavelet have compact support, so that (2.9) is a finite sum.

It is easy to see from the definitions of mother and father wavelet that the filters $\mathcal{G}$ and $\mathcal{H}$ satisfy the internal orthogonality relations

$$\sum_k g_k g_{k+2l} = \sum_k h_k h_{k+2l} = 0 \text{ for all } l \neq 0 \tag{2.10}$$

and

$$\sum_k g_k^2 = \sum_k h_k^2 = 1. \tag{2.11}$$

Moreover, they obey the mutual orthogonality relation

$$\sum_k g_k h_{k+2l} = 0 \tag{2.12}$$

for all $l \in \mathbb{Z}$. Such filters are known as *conjugate quadrature filters* (see [27] and [101]).

Finally, the *binary decimation operator* $\mathcal{D}_0$ chooses every even member of a sequence

$$(\mathcal{D}_0 x)_j = x_{2j}.$$

To make the notation easier, we will assume now that periodic boundary conditions are used (for further details on the symmetric case, see [85]). Since $\mathcal{G}$ and $\mathcal{H}$ are conjugate quadrature filters, the mapping $\{\mathcal{D}_0\mathcal{G}, \mathcal{D}_0\mathcal{H}\}$ is an orthogonal transformation. Furthermore, if x is a vector of length 2^m, then using periodic boundary conditions, each of $\mathcal{D}_0\mathcal{G}x$ and $\mathcal{D}_0\mathcal{H}x$ will be a vector of length 2^{m-1}.

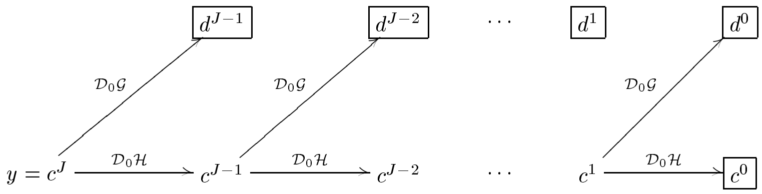

Figure 2.2: Schematic representation of The Discrete Wavelet Transform

Furthermore, denote $c_J = y$. We define now recursively for $j = J-1, J-2, \ldots, 0$ the smooth component c_j at level j and the detailed component d_j at level j by

$$c_j = \mathcal{D}_0\mathcal{H}c_{j+1} \quad \text{and} \quad d_j = \mathcal{D}_0\mathcal{G}c_{j+1}. \tag{2.13}$$

Finally, the DWT of the data y is defined to be the coefficients $d_{J-1}, d_{J-2}, \ldots, d_0, c_0$ (see Figure 2.3). Note that c_j and d_j are vectors of length 2^j.

In our situation the operators $\mathcal{D}_0\mathcal{G}$ and $\mathcal{D}_0\mathcal{H}$ can be identified with matrices. Therefore we denote by G_j and H_j the linear functions that correspond to the application of $\mathcal{D}_0\mathcal{G}$ and $\mathcal{D}_0\mathcal{H}$ to c_j. If the filter sequence $\{h_n\}$ has N non-zero members, then the $2^{j-1} \times 2^j$-matrices G_j and H_j can have at most 2^jN non-zero elements. So we need no more than 2^jN multiplications to compute c_j and d_j from c_{j+1}. Therefore, summing over j shows that the DWT is an algorithm of order n.

Furthermore, the DWT is orthogonal because the mapping $\{\mathcal{D}_0\mathcal{G}, \mathcal{D}_0\mathcal{H}\}$ is orthogonal. Therefore, the DWT is easily invertible and its inverse is performed by an algorithm of order n (see [85], [78] and [91] for further details).

It will ease the understanding of Chapter 3 to have an idea of the shapes of the matrices G_j and H_j, so we sketch now their shape for the case $N = 4$ where only $h_0, \ldots, h_3$ are different from zero. Using periodic boundary conditions we obtain (for $j > 1$)

$$
H_j^p = \begin{pmatrix}
h_0 & h_1 & h_2 & h_3 & & & & & & & & \\
& & h_0 & h_1 & h_2 & h_3 & & & & & & \\
& & & & h_0 & h_1 & h_2 & h_3 & & & & \\
& & & & & & & & & & & \\
& & & & & & & & h_0 & h_1 & h_2 & h_3 \\
h_2 & h_3 & & & & & & & & & h_0 & h_1
\end{pmatrix}
$$

and

$$
G_j^p = \begin{pmatrix}
g_0 & g_1 & & & & & & & & & g_{-2} & g_{-1} \\
g_{-2} & g_{-1} & g_0 & g_1 & & & & & & & & \\
& & g_{-2} & g_{-1} & g_0 & g_1 & & & & & & \\
& & & & g_{-2} & g_{-1} & g_0 & g_1 & & & & \\
& & & & & & & & & & & \\
& & & & & & & & g_{-2} & g_{-1} & g_0 & g_1
\end{pmatrix}.
$$

If symmetric boundary corrections are applied, everything gets somewhat more complicated because width and height may be even or odd. Moreover, a number of "extra values" has to be taken into account and has influence on the shape of the matrices (see [85] for details). For example, for even width and even number of extra values H_j^s looks like this:

$$
H_j^s = \begin{pmatrix}
h_2+h_1 & h_3+h_0 & & & & & & & & & & \\
h_0 & h_1 & h_2 & h_3 & & & & & & & & \\
 & & h_0 & h_1 & h_2 & h_3 & & & & & & \\
 & & & & h_0 & h_1 & h_2 & h_3 & & & & \\
 & & & & & & & & & & & \\
 & & & & & & & & h_0 & h_1 & h_2 & h_3 \\
 & & & & & & & & & & h_0+h_3 & h_1+h_2
\end{pmatrix}.
$$

If the latter number is odd and the width still even we obtain

$$
H_j^s = \begin{pmatrix}
h_3+h_2 & h_1 & h_0 & & & & & & & \\
h_1+h_0 & h_2 & h_3 & & & & & & & \\
 & h_0 & h_1 & h_2 & h_3 & & & & & \\
 & & & h_0 & h_1 & h_2 & h_3 & & & \\
 & & & & & & & & & \\
 & & & & & & & h_0 & h_1 & h_2+h_3 \\
 & & & & & & & & h_2+h_3 & h_0+h_1
\end{pmatrix}.
$$

It is not difficult to imagine the shape of H_j^s for odd width. Finally, we remark that unlike the periodic case G_j^s has non-zero entries at the same places as H_j^s.

We conclude this section by discussing another very important property of the DWT. When we defined the mother wavelet above, we mentioned that the property of vanishing moments $(\psi 3)$ is very useful because it leads to economical representations of functions. We will show now that there is an analogous statement for the discrete case.

Assume that f is a polynomial of order $l < m$

$$f(x) = \sum_{\kappa=0}^{l} \alpha_\kappa x^\kappa \tag{2.14}$$

and that $(g_k)_{k\in\mathbb{Z}}$ are the filter coefficients of a wavelet basis of order m. Then every wavelet coefficient which is not influenced by boundary conditions and obtained by applying the DWT on an equally spaced sample of f is equal to zero.

Proof. Without restrictions we can assume that the sample to which the DWT is applied is simply $f(0), f(1), \ldots, f(k)$. First we prove the statement for the coefficients at the finest level. Every such coefficient not close to the boundaries can be written as

$$\sum_k f(k) g_{k-2m} = \sum_k f(k+2m) g_k$$

for some integer m. The function $\tilde{f}(x) = f(x+2m)$ is again a polynomial of order l. Equation (2.14) is now obtained by noting that

$$\sum_k k^\mu g_k = 0$$

for all $\mu \in \{0, \ldots, m\}$. This result is proved for example by Kaiser (see [70]) and Kovac (see [71]).

To prove the statement for other levels we show that applying the filter $\mathcal{D}_0\mathcal{H}$ to an equally spaced sample from a polynomial of order l yields again an equally spaced sample from a polynomial of order $\tilde{l} \leq l$. This is fairly easy because all of these smooth coefficients can be written as

$$\sum_k f(k) h_{k-2m} = \sum_k f(k+2m) h_k = \sum_k \left(\sum_{j=0}^{l} \alpha_j (k+2m)^j \right) h_k.$$

Changing the order of the summations and reordering the summands in this expression yields an polynomial of order $\tilde{l} \leq l$.

In other words, when the DWT is applied to f, then the smooth components at each level is a polynomial of order l (except near the boundaries). On the other hand, the details at level J are obtained by applying the filter $\mathcal{G}$ to the smooth coefficients at level $J+1$. Therefore, the first part of the proof shows that they vanish. This completes the proof. $\square$

Note that there are levels where every coefficient is influenced by the boundaries. It is possible to show with induction that every coefficient $d_{j,k}$ depends on $\gamma(j,k) = N \cdot 2^{J-j} + (J-j)(N-2)$ values of the double infinite sequence, where N is the number of non-zero filter coefficients. If $N > 2$ and j is small enough, then $\gamma(j,k)$ is greater than $n = 2^J$, and therefore the computation of each coefficient in level j requires that some values of the finite data are used several times. We define j_0 by

$$j_0 = \min\{j \in \mathbb{N} : 2^{J-1-j} + (J-1-j)(N-2) \leq 2^J\}. \tag{2.15}$$

Hence, j_0 is the coarsest level such that there is a coefficient which is not influenced by the boundaries. This level is sometimes called *Low-resolution cutoff*.

2.4 Non-parametric regression

One classical problem in statistics is *non-parametric regression.* Noisy observations $y_1, \dots, y_n$ at time points $t_1, \dots, t_n$ are modelled as

$$y_i = f(t_i) + \varepsilon_i,$$

where f is the function to be estimated and $\varepsilon_1, \dots, \varepsilon_n$ is noise. It is often assumed that the noise is white noise with mean 0 and variance σ^2.

The statistical literature is full of different techniques that have been proposed to tackle this problem and we review in the following subsections four classical approaches. Our aim is not to give a comprehensive overview of all possible approaches and refinements. We rather want to show the variety of ideas that can lead to smoothing methods.

2.4.1 Kernel estimators

The idea with *kernel estimators* is to estimate $\hat{f}(t)$ by an weighted average of the given data in a small window centred around t. Classically (see [82] and [121]), they are defined by the formula

$$\hat{f}(t) = \frac{\sum_{i=1}^{n} y_i K(\frac{t-t_i}{\lambda})}{\sum_{i=1}^{n} K(\frac{t-t_i}{\lambda})},$$

where the kernel $K : \mathbb{R} \to \mathbb{R}$ is required to satisfy the following properties:

1. $\text{supp}K \subset [-1, 1]$,
2. $\max_{t\in[-1,1]} K(t) = K(0)$,
3. $\int_{-1}^{1} K(u)du = 1$,
4. $\int_{-1}^{1} uK(u)du = 0$,
5. $\int_{-1}^{1} u^2 K(u)du \neq 0$,
6. $\int_{-1}^{1} K^2(u)du < \infty$.

A popular choice for K is a triangular kernel of the form

$$K(t) = 1 - (|t|/C) \cdot 1_{\{|t|\leq \frac{1}{C}\}}$$

because because it is simple and fast to calculate. Note that there are also other ways of defining kernel estimators (compare [43]). For example, the first condition is often relaxed by the requirement that the kernel decays rapidly.

Hastie and Tibshirani (see [58]) point out that the choice of the *bandwidth* $\lambda > 0$ is much more important than the choice of kernel. With λ small, $\hat{f}(t)$ strongly depends on the data near t, and the kernel estimator produces a wiggly curve. On the other hand, if λ is large, then more data points are averaged and the estimator is smooth. In the extreme case where λ diverges to ∞, $\hat{f}(t)$ converges to the mean of $y_1, \ldots, y_n$.

The choices of λ and K have been discussed widely, and many different ideas can be found in the statistical literature (see for example [43] and [119]).

2.4.2 Local polynomial regression

Local polynomial regression currently enjoys considerable popularity. It was introduced by Stone in 1979 (see [105]) and important recent work includes the articles and the monograph of Fan and Gijbels (see [44], [45] and [46]).

By using Taylor's expansion we can approximate the regression function f in a neighbourhood of t by

$$f(t) \doteq \sum_{j=0}^{p} \frac{f^{(j)}(x)}{j!}(t-x)^j$$

where $\doteq$ denotes approximate equality. Consequently it is reasonable to model $f(t)$ locally by a simple polynomial model and to perform a locally weighted polynomial regression

$$\sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{p} \beta_j (t_i - x)^j \right)^2 K(\frac{t_i - x}{h}). \tag{2.16}$$

Here K is a kernel function and h a bandwidth. Denote the minimiser of (2.16) by $\hat{\beta}$. Then an estimator for $f^{(j)}(t)$ is given by

$$\hat{f}_j(t) = j!\hat{\beta}_j.$$

This procedure is repeated for several values of t to obtain an estimator $\hat{f}$ for the signal f. On first sight, this appears to be very complicated and costly, but in fact it is possible to implement local polynomial regression in an algorithm or order n. Updating ideas where averages are computed recursively by updating averages previously computed have made this possible (see [46]).

Several suggestions have been made for the choice of bandwidth, kernel and order of the polynomial fit, but we do not elaborate this here and refer to the literature for details.

2.4.3 Spline smoothing

The next approach that we consider is spline smoothing. This approach makes use of two conflicting aims when estimating the regression function. The first one is a goodness-of-fit measure like the l^2-distance between the data and $\hat{f}$. Minimising this criterion would yield a function that interpolated the data, so a smoothness measure is considered as well, for example the integral of the squared second derivative of $\hat{f}$. The estimator $\hat{f}$ is then chosen as the twice differentiable function that minimises the sum

$$S(\hat{f}) = \frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{f}(t_j))^2 + \lambda \int \hat{f}''(x)^2 dx.$$

It can be shown that the solution of the minimisation problem is a cubic spline with knots $(t_i)_{i=1}^n$, so that $\hat{f}$ is a cubic polynomial between any two neighbouring time points t_i and t_{i+1}.

The parameter λ controls again the smoothness of the regression function. If λ equals 0, then $\hat{f}$ is the spline interpolant of the data. On the other hand, as λ diverges to ∞, then $\hat{f}$ converges to the linear “Least Squares” estimator. There are several suggestions how λ can be chosen, and we refer again to the vast amount of literature on the use of splines in statistics (for example [43], [48] and [97]).

2.4.4 Fourier series estimators

Finally, we have a short look at *Fourier Series Estimators*. They are very similar to the wavelet estimators that we examine in the next section. We assume that the time points are equally spread on the unit interval:

$$t_k = k/n.$$

The idea is that every function $f \in L^2([0,1])$ can be written as a Fourier Series:

$$f(t) = \sum_{j=1}^{\infty} \beta_j \exp(2\pi i j t).$$

Since $(\beta_j)_{j\in\mathbb{N}} \in l^2$, there is a $\lambda \in \mathbb{N}$ such that the finite sum

$$f_\lambda(t) = \sum_{j=1}^{\lambda} \beta_j \exp(2\pi i j t)$$

is a good approximation to f, and hence

$$y_k \doteq \sum_{j=1}^{\lambda} \beta_j \exp(2\pi i j t_k) + \varepsilon_k. \tag{2.17}$$

Therefore, we obtain for each λ a linear model and can apply methods of linear regression to choose the coefficients β_j. Straightforward calculations show (see [43]) that the least squares estimator leads to the explicit formula

$$\hat{f}_\lambda(t) = \frac{1}{n}\sum_{k=1}^{n} y_k K_\lambda(t - t_k),$$

where K_λ is the *Dirichlet Kernel*

$$K_\lambda(s) = \begin{cases} \frac{\sin(\pi(2\lambda+1)s)}{\sin(\pi s)}, & s \notin \mathbb{Z} \\ 2\lambda + 1, & \text{otherwise.} \end{cases}$$

2.4.5 Summary

We have considered four different concepts that produce a regression function $\hat{f}$. Each time a smoothing parameter λ is involved to adjust the smoothness of the regression function. However, the smoothness is characterised in a manner that does not allow $\hat{f}$ to adapt to local features. Either $\hat{f}$ is still noisy or sharp features have been blurred out.

We demonstrate this behaviour by an example. The "Doppler" function, which was also used by Donoho et al. (see [39] and [34]) to demonstrate differences between wavelet estimators and other linear techniques, was perturbed by Gaussian noise with $\sigma = 0.15$. Figures 2.3 and 2.4 show the regression functions that were obtained under S-Plus using the four smoothing methods and two different values for λ. [9] In Figure 2.3 an attempt was made to choose the smoothing parameters in a way such that the resulting curves look very smooth. While this aim was nearly achieved, the high frequency components of the signal are oversmoothed by each of the methods.

In Figure 2.4 we tried to retain the high curvature on the left-hand side of the signal, but this time the estimators produce regression functions that are very wiggly on the right-hand side. Automatic procedures for the choice of the smoothing parameter will typically produce curves which are more rough than in Figure 2.3 and oversmooth the data more than in Figure 2.4.

The problem with kernel estimators and local polynomial regression is that the bandwidth is chosen globally. Several authors have suggested methods for variable selection

[9] A triangular kernel was used for the kernel estimator.For the application of local poynomial regression the default kernel of S-Plus was used which is

$$K(x) = (1 - x^3)^3 \cdot 1_{\{0 \leq x < 1\}}.$$

Moreover locally-quadratic fitting was employed.

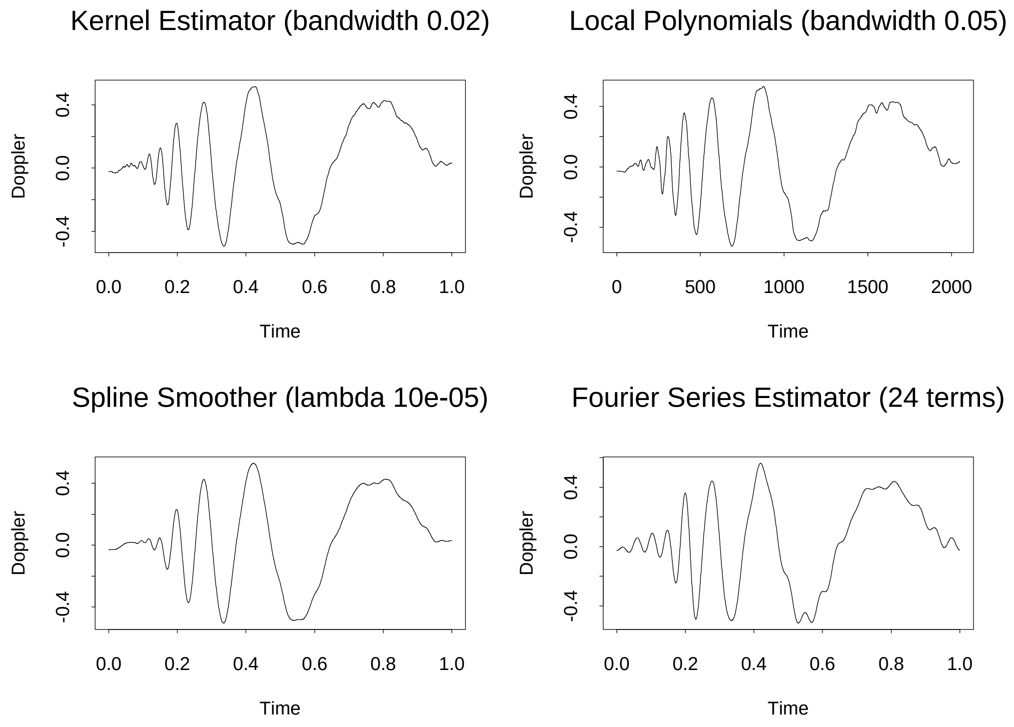

Figure 2.3: Application of four linear estimators to the Doppler data with a large amount of smoothing.

where the bandwidth adapts to the curvature of the signal (see [45] and [46]): A small bandwidth is used in regions where the curvature is high for example near peaks. On the other hand a larger amount of smoothing is introduced where the variation of the signal is small. One way of doing that is to derive theoretically an expression for the optimal bandwidth and to substitute afterwards the unknown quantities by pilot estimators. Another possibility is to split up the interval $[t_1, t_n]$ into subintervals and to calculate for each subinterval a bandwidth using some method for global bandwidth selection. This yields a bandwidth step function which is often smoothed before it is used for the reconstruction of the regression signal.

The spline estimators as described above suffer the problem that smoothness is defined globally by definition. Davies introduced recently another smoothness measure: the

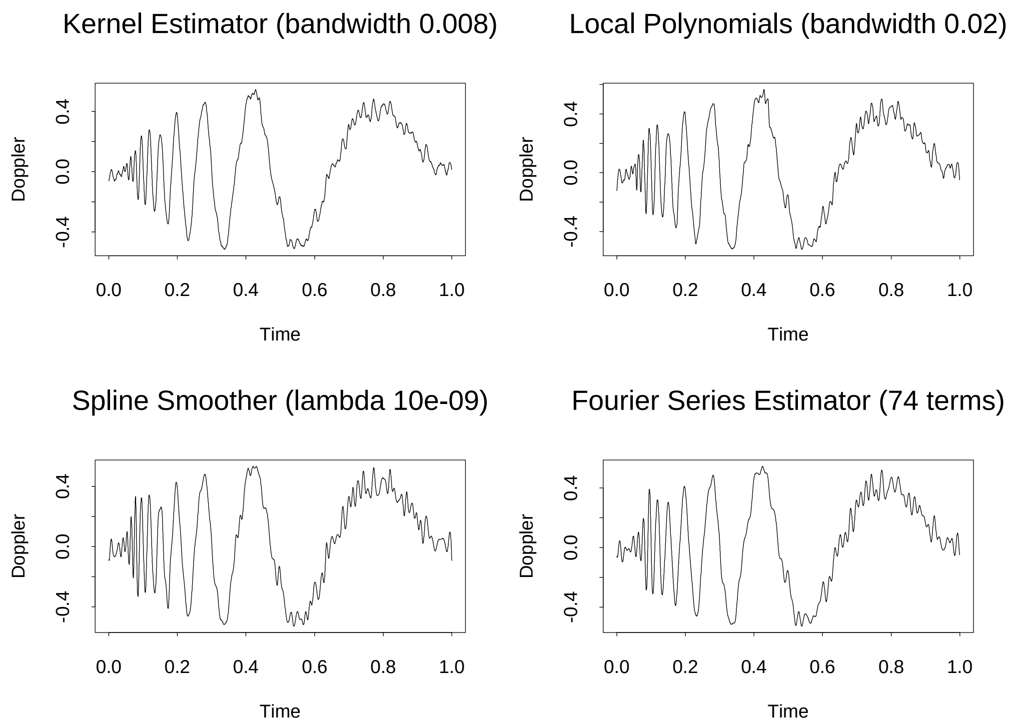

Figure 2.4: Another application of the estimators to the Doppler data, but with a small amount of smoothing.

number of local extrema (see [31] and [79]). This criterion is minimised among all functions that satisfy a special goodness-of-fit condition [10] and yields for every time point t_i an upper and lower boundary for the regression function.

Finally, Fourier series estimators are based on the Fourier basis that is only localised in frequency but not in time, so again the smoothness of the regression function can only be adjusted globally. We will see in the next section that wavelets offer a localisation in both frequency *and* time, and that this method leads to more satisfying results.

2.5 Wavelet regression

Wavelet Regression Estimators are a more recent approach to curve estimation, introduced by Donoho and Johnstone in 1991. The outline of the procedure is fairly easy. It is usually assumed that the time points $t_1, \ldots, t_n$ are equally spread over the unit interval

$$t_i = i/n,$$

where n is a power of two

$$n = 2^J$$

for some positive integer J.

Then *Wavelet Regression Estimators* are defined as follows:

Algorithm 2.1.

(1) Apply the discrete wavelet transform to the data $(y_1, \ldots, y_n)^T$

$$w = \mathcal{W}y.$$

[10] This condition is that the maximum run length of the residuals does not exceed a theoretical constant.

(2) Modify the wavelet coefficients and obtain modified coefficients w^*.

(3) Estimate f by performing the inverse DWT:

$$\hat{f}_n(t_i)_1^n = \mathcal{W}^T w^*.$$

The part of Algorithm 2.1 which needs more explanation is, of course, the second part. How should the wavelet coefficients w be modified to form w^*? A very popular way of doing this was suggested by Donoho (see [34],[35] and [36]), Donoho and Johnstone (see [39] and [40]), and Donoho et al. (see [41]). They propose *thresholding* the coefficients.

The idea is that smooth functions have economical wavelet representations, so that most of their coefficients can be set to zero without causing a large error. Moreover, white noise is transformed by the DWT to white noise with same variance, because the DWT is an orthogonal transformation. Therefore, in step (2) of Algorithm 2.1 we set small coefficients equal to zero and keep or shrink the others. More precisely, we introduce two thresholding rules, given by the soft thresholding function

$$\eta_S(w_{j,k}, \tau) = \operatorname{sgn}(w_{j,k})(|w_{j,k}| - \tau)_+,$$

where $(x)_+ = \max(0, x)$, and the hard thresholding function

$$\eta_H(w_{j,k}, \tau) = w_{j,k} \cdot I\{|w_{j,k}| \geq \tau\}.$$

One of these two functions is now applied to the wavelet coefficients $w_{j,k}$ to produce modified coefficients $w^*_{j,k}$. To be exact, they are usually not applied to *all* coefficients, but to every coefficient $w_{j,k}$ where $j \geq j_0$. Originally, Donoho proposed (see [39]) to choose j_0 as the Low-resolution cutoff (as defined in (2.15)) because the property of vanishing moments is completely missing in the levels that are smaller than j_0 (see the remark at the end of Section 2.3). However, in later papers, Donoho does not go into the choice of j_0. Hall and Patil suggest in the discussion of [41] to increase j_0 with increasing n and further details can be found in [51]. Hall and Nason (see [50]) show by an expansion of the mean integrated squared error that the Low-resolution cutoff in wavelet thresholding plays the same role as the bandwidth in kernel estimators with the notable exception that the usual formulation of wavelet thresholding allows only a very small number of possible values for j_0. The authors suggest adapting classical procedures for bandwidth selection to wavelet regression like cross-validation. Finally, the influence of choosing j_0 on the wavelet estimator is also discussed in an article of Abramovich and Benjamini (see [1]). In the remainder of this thesis we will use the default value of the WaveThresh package $j_0 = 3$ in all examples if not explicitly given.

A lot more of work is available concerning the choice of a suitable *threshold* τ. Note that the choice of τ is crucial: if its value is too small or too large, then the wavelet shrinkage estimator will tend to overfit or underfit the data respectively.

Very popular is Donoho's *VisuShrink* estimator (see [36]). Its goal is to produce a noise-free reconstruction and this is achieved by a relatively high threshold choice which is based on the following fact (see [74]): Let (x_n) be i.i.d. $\mathcal{N}(0,1)$. Then

$$\mathbb{P}\left(\left\{\max_i |x_i| \leq \sqrt{2\log(n)}\right\}\right) \to 1, \quad n \to \infty.$$

This motivates the threshold choice

$$\tau^{VS} = \sqrt{2\log(n)}\,\sigma.$$

When the noise level σ is unknown, Donoho and Johnstone (see [39]) suggest to estimate it from the data by taking the median absolute deviation (MAD) of the wavelet coefficients at the finest level, and dividing by 0.6745 because the median of the distibution of X for a standard normal variable X is 0.6745. distribution, i.e.

$$\hat{\sigma} = \mathrm{MAD}((w_{J-1,k})_k)/0.6745. \tag{2.18}$$

Although *VisuShrink* is a very simple method, Donoho et al. have proved asymptotic results about the behaviour of this estimator. They show (see [36], [39], [38] and [41]) that the estimates are with high probability “as smooth as the truth” and that they are near-optimal for spatial adaptation, estimating a function at a point and estimating the object in global loss. [11] Furthermore, Johnstone and Silverman (see [67]) have generalised the estimator for the case where the noise is correlated and stationary. They consider *level-dependent* thresholds because the variance of the noise of a wavelet coefficient $w_{j,k}$ depends on the level j but not on k.

We will see another version of *VisuShrink* later in this thesis as well as a variant on the *SureShrink* estimator. This latter method was originally proposed by Donoho and Johnstone (see [40]) for the i.i.d. Gaussian error setting, and also adapted for correlated noise by Johnstone and Silverman (see [67]). If the signal f was known then it would be possible to calculate an “optimal” threshold τ that minimises the risk which is simply the expected mean square error

$$\mathbb{E}\left(\frac{1}{n}\|f - \hat{f}\|_2^2\right).$$

Note that by the orthogonality of the wavelet transform and Parseval’s identity this is equal to

$$\mathbb{E}\left(\frac{1}{n}\|w - \hat{w}\|_2^2\right),$$

[11] Here loss can be chosen from a large scale of loss measures including L^p loss for $p > 1$.

where w and $\hat{w}$ are the wavelet coefficients of f and $\hat{f}$. However, in practice we do not know the signal or its wavelet coefficients, but we can nevertheless estimate the risk in an unbiased fashion.

This is possible by a result due to Stein (see [104]): Suppose that $X \sim \mathcal{N}_n(w, \Sigma)$ and that $g : \mathbb{R}^n \to \mathbb{R}^n$ is a weakly differentiable function. Stein showed that the mean square error of an estimator $\hat{w} = X + g(X)$ may be written

$$\mathbb{E}\left(||\hat{w} - w||^2\right) = \mathbb{E}\left(\mathrm{tr}(\Sigma) + ||g(X)||^2 + 2\,\mathrm{tr}\left(\Sigma \cdot Dg(x)\right)\right), \tag{2.19}$$

where $Dg(X)$ denotes the $n \times n$ matrix with entries $\partial g_i / \partial x_{j,k}(X)$. Suppose now that X are the wavelet coefficients that we obtain by applying the DWT to a noisy sample y. In the case of soft thresholding with threshold τ, g has components

$$g_{j,k}(x) = \begin{cases} -\tau, & x_{j,k} > \tau, \\ -x_{j,k}, & |x_{j,k}| \leq \tau, \\ \tau, & x_{j,k} < -\tau. \end{cases}$$

Therefore, $g_{j,k}$ is a function of $x_{j,k}$ alone, and the matrix Dg in (2.19) is diagonal. We get from (2.19) that the quantity

$$S(\tau;\ x) = \hat{\sigma}^2 n + \sum_{j,k} \left(\min(x_{j,k}^2, \tau^2) - 2\hat{\sigma}^2 \cdot \{|x_{j,k}| \leq \tau\}\right) \tag{2.20}$$

is an unbiased estimate of the risk in (2.19):

$$\mathbb{E}\left(S(\tau;\ x)\right) = \mathbb{E}\left(||\hat{w} - w||^2\right).$$

Therefore, it is reasonable to choose the threshold τ by minimising this expression. So, the *SureShrink* [12] estimator is obtained by taking

$$\tau = \mathrm{argmin}_{0 \leq \tau \leq \sqrt{2\log(n)}}\, S(\tau;\ y).$$

[12] Sure=Stein's unbiased risk estimator

Donoho and Johnstone have shown (see [40]) that this minimisation can be accomplished in $O(n \log(n))$ time.

Nason (see [84] and [83]) presents other threshold selections that aim to minimise the mean square error. His approaches try to imitate the cross-validation method which is widely used as an automatic procedure to choose a smoothing parameter in many statistical settings (see [43], [98], [48]). Usual cross-validation is carried out by systematically expelling a data point from the construction of an estimate, predicting what the removed value would have been and comparing the prediction to the value of the expelled point. However, this simple leave-one-out procedure cannot be directly applied to wavelet shrinkage because the DWT only operates on data sets that contain a power of two number of elements.

Therefore, Nason mainly discusses the *two-fold cross-validation.* This procedure works by leaving out half of the data points. This leaves 2^{J-1} data points that are then used to form a wavelet shrinkage estimator using a particular threshold. The values of the expelled points can then be compared with the shrinkage estimator to form an estimate of prediction error at a particular threshold. This quantity can be then numerically minimised over values of the threshold.

Nason shows in simulation studies that his threshold selection works very well in a lot of different situations. Furthermore, there is a generalisation due to Wang (see [120]) who divides the data in 2^m groups, where the first group consists of the data points $y_1, y_{1+2^m}, y_{1+2\cdot 2^m}, \ldots$, the second group has points $y_2, y_{2+2^m}, y_{2+2\cdot 2^m}, \ldots$ and so on. For every group he calculates a reconstruction and compares the reconstructed values with an interpolation obtained from the rest of the values. Summing up over all differences gives a new criterion which is to be minimised among several threshold values to yield the threshold choice.

There are other cross validation procedures studied by Nason (see [84]), Weyrich and Warhola (see [123]), Jansen and Bultheel (see [64]) and Jansen et al. (see [65]), but we do not go into more detail.

Moreover, there are lots of other ideas to shrink wavelet coefficients, coming from different areas like multiple hypothesis testing (see Abramovich and Benjamini in [1] and [2] and Ogden and Parzen in [88] and [87]), Bayesian statistics (see Vidakovic in [118], Clyde et al. in [25], Chipman et al. in [21] and Abramovich et al. in [3]) or Wiener filtering (see Bertrand et al. in [9]). Ogden and Parzen have developed further ideas in [88] and [87].

Let us finally return to the Doppler data that we have examined in the last section with several linear estimators. Remember that these methods could not cope with the curvature on the left-hand side while producing a smooth output on the right-hand side at the same time. We used hard thresholding with the VisuShrink threshold, Daubechies' filter with extremal phase of order 4 and cut-off-level $j_0 = 3$. The result can be seen in Figure 2.5. Obviously, it looks totally different from the curves that we obtained in the last section in Figures 2.3 and 2.4. The right half is very smooth (even smoother than the linear estimators in Figure 2.3 where the amount of smoothing was relatively high) and looks very similar to the original Doppler function (see Figure 5.1). At the same time, the left half gives a good impression of the oscillation of the Doppler function and shows more peaks than the linear estimators.

Another way of comparing the linear estimators with wavelet thresholding is to exam-

Method	Mean Squared Error
Kernel estimator	0.0026
Spline smoothing	0.0029
Fourier series	0.0033
Local polynomials	0.0028
Wavelet Thresholding	0.0023

Table 2.1: Mean squared error for the Doppler data, four linear smoothing methods and wavelet thresholding. The smoothing parameters were determined such that the mean squared error was minimal for the linear methods, and the SURE threshold selection was employed for wavelet thresholding.

ine the mean squared error

$$\sum_{i=1}^{n}(f(t_i) - \hat{f}(t_i))^2.$$

Table 2.1 shows the mean squared error for the Doppler data and the various estimators. While the SURE thresholding mechanism was used for the wavelet estimator, the smothing parameters of the linear estimators were chosen such that the error was minimal. In other words, wavelet shrinkage with an automatic thresholding procedure is still better in terms of the mean squared error than any of the linear estimators with an optimal choice of the smoothing parameter. However, it is likely that for example local polynomial regression with variable choice of bandwidth attains similar results.

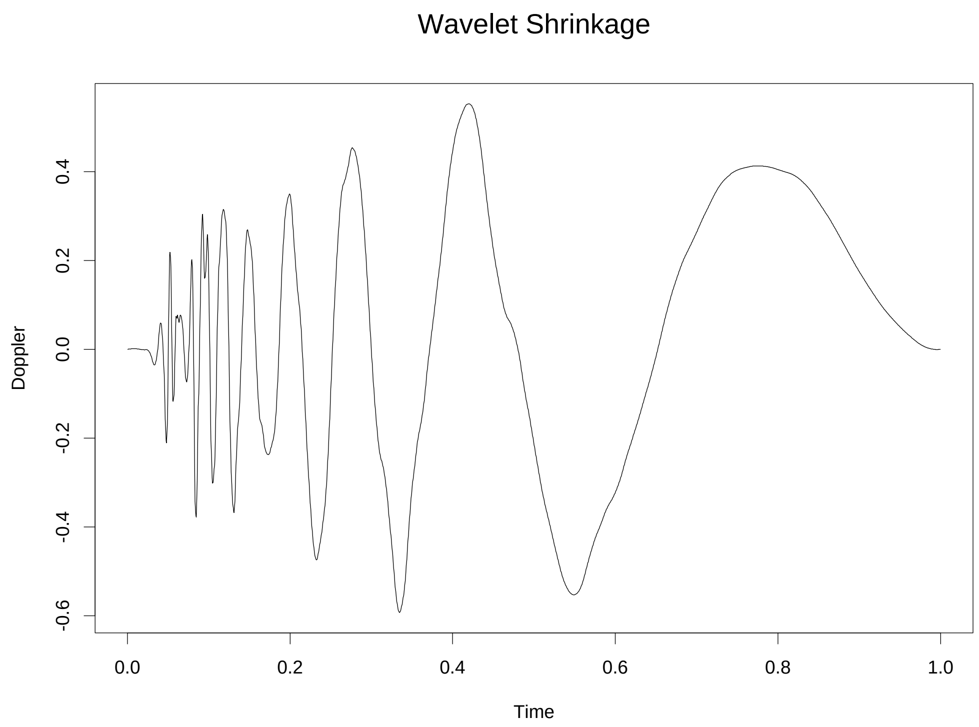

Figure 2.5: Wavelet Shrinkage applied to the Doppler data

Chapter 3

Calculation of the variances of wavelet coefficients

This operator did his office after a different manner from those of his trade in Europe.
He first took my altitude by a quadrant, and then, with rule and compasses,
described the dimensions and outlines of my whole body,
all which he entered upon paper, and in six days brought my clothes very ill made,
and quite out of shape, by happening to mistake a figure in the calculation.

—Jonathan Swift, *Gulliver's Travels*

Classical thresholding as described in the previous chapter is based on two properties of the wavelet transform: Smooth functions have economical representations, so that most of the coefficients are nearly zero, and white noise is transformed to white noise. Therefore it is reasonable to assume that small coefficients are noise and can be set to zero, while the signal is stored in a few large coefficients which should be retained.

If the noise is correlated then the second property no longer holds. Johnstone and Silverman (see [67]) have recently extended wavelet threshold estimators for data with stationary correlated noise. In this situation the variances of the wavelet coefficients at one level are identical, so that they threshold the coefficients level by level.

In the next chapter we will apply the DWT to data which were preprocessed by a linear transformation. In general the noise of the transformed data is no longer stationary. We will assume that it is multivariate normal with mean 0 and covariance matrix Σ. In our formulation, however, Σ will be a band matrix. We will be interested to know the variances of the wavelet coefficients, and therefore we need an algorithm that calculates these variances in a fast way. We will see in this chapter that it is possible to use the band matrix structure of Σ to derive an algorithm of order $n \cdot b_J \cdot N^2$, where b_J is the bandwidth of Σ (the number of non-zero diagonals) and N the number of filter coefficients.

3.1 An algorithm for computing the variances

Suppose that $\tilde{\varepsilon}_0, \ldots, \tilde{\varepsilon}_{2^J-1}$ have a multivariate normal distribution with mean 0 and covariance matrix Σ^J and let us perform the DWT on the $\tilde{\varepsilon}_i$, getting wavelet coefficients $d_{j,k}$. We will develop in this chapter an algorithm for finding the variances of all the wavelet coefficients $d_{j,k}$. In the case where Σ is a band matrix, the number of operations will be linear in the length of $\tilde{\varepsilon}$, while even in the case where we only require that Σ is symmetric and positive-definite, the algorithm is quadratic.

Using the notation that we introduced in Chapter 2 we can write each wavelet coefficient as

$$d_{j,k} = (G_{j+1} \cdot H_{j+2} \cdots H_J \tilde{\varepsilon})_k. \tag{3.1}$$

Therefore, the variance of each wavelet coefficient $d_{j,k}$ is given by

$$\operatorname{var} d_{j,k} = (G_{j+1} \cdot H_{j+2} \cdots H_J \cdot \Sigma^J \cdot (G_{j+1} \cdot H_{j+2} \cdots H_J)^T)_{k,k}. \tag{3.2}$$

Equation (3.2) suggests that the variances $\sigma^2_{j,k}$ can be calculated by computing the covariance matrices of the smooth component at level j

$$\Sigma^j = H_{j+1}\Sigma^{j+1}(H_{j+1})^T \tag{3.3}$$

and the detail at level j

$$\tilde{\Sigma}^j = G_{j+1}\Sigma^{j+1}(G_{j+1})^T,$$

where $j = J-1, J-2, \ldots, 0$.

Naturally, this not only gives us the $\sigma^2_{j,k} = \tilde{\Sigma}^j_{k,k}$, but also the covariance structure of the wavelet coefficients at each level. If one is only interested in the values of the $\sigma^2_{j,k}$, then it is of course not necessary to calculate all entries of the matrices $\tilde{\Sigma}^j$. In this case it suffices to calculate only the main diagonal elements of $\tilde{\Sigma}^j$.

The order of Σ^j and $\tilde{\Sigma}^j$ will be denoted by n_j. If periodic boundary corrections are applied, we simply get $n_j = 2^j$. However, the symmetric case is somewhat more difficult, see [85] for details and also Section 3.3. If we combine the definition of matrix multiplication with equation (3.3) we obtain for each entry

$$\begin{aligned}\Sigma^j_{i,m} &= \sum_{k,l=0}^{n_{j+1}-1} (H_{j+1})_{i,k}\Sigma^{j+1}_{k,l}(H_{j+1})^T_{l,m} \\ &= \sum_{k,l=0}^{n_{j+1}-1} (H_{j+1})_{i,k}(H_{j+1})_{m,l}\Sigma^{j+1}_{k,l}.\end{aligned} \tag{3.4}$$

Remember now the shape of H_{j+1}: Every row contains a shifted version of the N filter coefficients, so that the sum in (3.3) can be reduced to

$$\Sigma^j_{i,m} = \sum_{k,l=0}^{N-1} h_k h_l \Sigma^{j+1}_{2i+k,2m+l} \tag{3.5}$$

using periodic boundary conditions. The indices $2i+k$ and $2m+l$ are of course to be treated modulo n_j. Similarly, we get

$$\tilde{\Sigma}^j_{i,m} = \sum_{k,l=0}^{N-1} g_k g_l \Sigma^{j+1}_{2i+k-N+2,2m+l-N+2} \tag{3.6}$$

for the detail at level j. Finally, we remark that we obtain for the symmetric case

$$\Sigma^j_{i,m} = \sum_{k,l=0}^{N-1} h_k h_l \Sigma^{j+1}_{\rho(2i+k-N+2-(o_{j+1} \bmod 2)),\rho(2m+l-N+2-(o_{j+1} \bmod 2))}, \tag{3.7}$$

where ρ gives the boundary correction with

$$\rho(p) = \begin{cases} -1-p, & \text{if } p < 0, \\ p, & \text{if } 0 \le p < n_{j+1}, \\ 2n_{j+1} - p - 1, & \text{if } n_{j+1} \le p \end{cases}$$

and $o_{j+1} \in \{0, 1\}$ is the number of extra values modulo 2 at level $j+1$ (see [85]).

Note that this reduces the number of multiplications from $(n_{j+1})^2$ to N^2 where the number of filter coefficients N is independent of the number of (grid) observations 2^J. This shows that the complexity of the computation of Σ^j is $O((2^j)^2 \cdot N^2)$ because the equations (3.5) and (3.6) are calculated $(n_j)^2$ times. Hence, the overall complexity of the algorithm for general covariance matrix Σ is $O((2^J)^2 \cdot N^2)$.

Let us now turn to the case where Σ is a band matrix. In this case the covariance matrices Σ^j and $\tilde{\Sigma}^j$ have also a band structure because the wavelet filter matrices G_j and H_j are band matrices.[1] We will discuss now how to use the sparsity structures of all these matrices to obtain a fast algorithm.

One way of getting a fast algorithm is to determine the non-empty diagonals of Σ^j (see next section) and to calculate the equations (3.5) and (3.6) only for these diagonals. This leads to an algorithm of order $2^J \cdot b_J \cdot N^2$, as we will see below. Here b_J denotes the number of non-zero diagonals in Σ^J above the main diagonal. A faster alternative, rather than calculating each entry of Σ^j individually as a linear combination of the $\Sigma^{j+1}_{2i+k,2m+l}$,

[1]This is not completely true when periodic boundary conditions are used. In this case, Σ^j and $\tilde{\Sigma}^j$ have additional entries in the upper right and the lower left corner. See below in the text.

is to compute them simultaneously: Set all $\Sigma^j_{i,m}$ to zero, determine for each non-zero entry of Σ^{j+1} in which sums it is involved and perform the necessary multiplications and additions. In practice the latter method can be about twice as fast as the first one and was used in an implementation of the algorithm in C. The source code is part of the extension package for WaveThresh which we describe in Chapter 4 and can be downloaded from the web page

`http://feat.mathematik.uni-essen.de/~kovac/thesis.html.`

See function `irregwd` for more details.

3.2 Computational complexity in the periodic case

We will now examine how many diagonals in each Σ^J are not empty. This is useful for the algorithm itself when determining which entries of Σ^j are to be computed. Another consequence is that the proposed method is of order $2^J \cdot b_J \cdot N^2$. In other words, as long as we assume the bandwidth of Σ^j to be bounded by a constant, the computation time of the algorithm increases only linearly with the number of points.

In this section we concentrate on periodic boundary conditions, while the next section deals with the symmetric case. We define the *bandwidth* b_j as the smallest integer such that for all $i, m \in \{0, \ldots, n_j - 1\}$ with

$$b_j < |i - m| < n_j - b_j$$

it follows that $\Sigma^j_{i,m} = 0$. Note that the bandwidth of the identity matrix is 0. Moreover, with this definition the “generalised” band matrix Σ^j has no more than $(2b_j+1)\cdot n_j$ entries that are different from zero.

Let us first prove a small lemma:

Lemma 3.1. *With periodic boundary conditions the inequality*

$$b_j \leq \frac{b_{j+1} + N}{2} \tag{3.8}$$

holds for all levels $j = J-1, J-2, \ldots, 0$.

Proof. For levels j which satisfy $n_{j+1} < b_{j+1} + N$ this is easily seen because

$$b_j \leq n_j = \frac{n_{j+1}}{2} < \frac{b_{j+1} + N}{2}.$$

So let us assume now that

$$n_{j+1} \geq b_{j+1} + N. \tag{3.9}$$

In order to prove inequality (3.8) we have to show for all $i, m \in \{0, \ldots, n_j - 1\}$ with

$$\frac{b_{j+1} + N}{2} < |i - m| < n_j - \frac{b_{j+1} + N}{2}$$

and for all $k, l \in \{0, \ldots, N-1\}$ that we have

$$b_{j+1} < |((2i + k) \bmod n_{j+1}) - ((2m + l) \bmod n_{j+1})| < n_{j+1} - b_{j+1}. \tag{3.10}$$

Then every term in (3.5) vanishes, and consequently we get $\Sigma^j_{i,m} = 0$.

We distinguish several cases depending on the values of $2i+k$ and $2m+l$, in particular whether each of these lies above or below n_{j+1}.

- If $2i + k, 2m + l \in \{0, \ldots, n_{j+1} - 1\}$, then the inequality is obtained just by using the triangular inequality. For example, the right inequality is proved by

$$\begin{aligned}
|((2i+k) \bmod n_{j+1}) - ((2m+l) \bmod n_{j+1})| &= |2(i-m)+k-l| \\
&\leq 2|i-m| + |k-l| \\
&< 2(n_j - \frac{b_{j+1}+N}{2}) + N \\
&= n_{j+1} - b_{j+1}.
\end{aligned}$$

- Now turn to the case where $2i+k \geq n_{j+1} > 2m+l$. Then considering the possible values for i and k and assumption (3.9) give us

  $$2i+k \leq n_{j+1} + N - 3 \leq n_{j+1} + (n_{j+1} - b_{j+1}) - 3 < 2n_{j+1}.$$

  Therefore we have

  $$|((2i+k) \bmod n_{j+1}) - ((2m+l) \bmod n_{j+1})| = |(2i+k-n_{j+1}) - (2m+l)|. \tag{3.11}$$

  Note further that

  $$\begin{aligned}
  2i+k-(2m+l)-2n_j &< 2i+k-(2m+l)-(2|i-m|+b_{j+1}+N) \\
  &\leq k-l-b_{j+1}-N < 0,
  \end{aligned}$$

  so that formula (3.11) becomes

  $$\begin{aligned}
  |((2i+k) \bmod n_{j+1}) - ((2m+l) \bmod n_{j+1})| &= n_{j+1} + 2m + l - (2i+k) \\
  &= n_{j+1} - |2i+k-(2m+l)|.
  \end{aligned}$$

  Now we can apply the result from the first case and get inequality (3.10).

- The remaining two cases where $2m+l \geq n_{j+1}$ can be reduced to the first very similarly.

Therefore, inequality (3.8) is true for all level $j \in \{0, \ldots, J-1\}$. $\square$

The result of the preceding lemma can be used in an implementation of the algorithm to reduce the number of multiplications in the periodic case. Before calculating the covariance structures of the j-th level Σ^j and $\tilde{\Sigma}^j$, the actual bandwidth b_{j+1} of Σ^{j+1} is determined. Afterwards, only the main diagonal, the $\frac{b_{j+1}+N}{2}$ diagonals next to it and the corners of Σ^j are computed.

We complete the periodic case with a statement about the order of the algorithm.

Theorem 3.1. *With periodic boundary conditions the algorithm is of order* $b_J \cdot 2^J \cdot N^2$.

Proof. A direct consequence from Lemma 3.1 is that we have a uniform bound

$$b_j \leq \max\{b_J, N\}.$$

As we have pointed out above, the number of non-zero coefficients in level j is not greater than $(b_j + 1) \cdot n_j \leq C \cdot 2^j$, where $C = \max\{b_J, n\} + 1$. By summing up over $j \in \{0, \ldots, N-1\}$ we see that only $C \cdot 2^J$ new entries have to be computed. Moreover, the calculation of each coefficient requires N^2 multiplications. Therefore, the number of multiplications in the whole procedure is bounded by $C \cdot N^2 \cdot 2^J$ where C does not depend on J and our algorithm is of order $b_J \cdot 2^J \cdot N^2$. $\square$

3.3 Computational complexity in the symmetric case

We consider now the symmetric case in more detail. First, we introduce similarly to the periodic case the *bandwidth* of Σ^j. This time, Σ^j is a band matrix in the classical sense, so we define b_j as the smallest integer such that for all $i, j \in \{0, \ldots, n_j\}$ with

$$b_j < |i - j|$$

it follows that $\Sigma^j_{i,j} = 0$.

Let us continue with a remark on the order n_j of Σ^j: Although we do not have an exact formula for n_j, it is fairly easy to see from the usual filter definitions that

$$n_j \leq (n_{j+1} + N)/2. \tag{3.12}$$

We will now prove a statement which is equivalent to Lemma 3.1 in the periodic case:

Lemma 3.2. *With symmetric boundary conditions the inequality*

$$b_j \leq \frac{b_{j+1}}{2} + N \tag{3.13}$$

holds for all levels $j = J-1, J-2, \ldots, 0$.

Proof. To avoid some problems with boundary corrections, we distinguish two cases. The easier one is where $b_{j+1} + 2N > n_{j+1}$ which occurs essentially at low levels. In this situation we immediately get

$$b_j \leq n_j \leq \frac{n_{j+1} + N}{2} < \frac{b_{j+1}}{2} + N.$$

So let us assume now that

$$b_{j+1} + 2N \leq n_{j+1}. \tag{3.14}$$

We have to prove that the inequality

$$|\rho(2i + k - N + 2 - o_j) - \rho(2m + l - N + 2 - o_j)| > b_{j+1} \tag{3.15}$$

follows for all $i, m \in \{0, \ldots, n_j - 1\}$ with

$$|i - m| > \frac{b_{j+1}}{2} + N$$

and for all $k, l \in \{0, \ldots, N-1\}$. Let us fix such i, m, k, l and define κ and μ by

$$\kappa = 2i + k - N + 2 - o_j$$

and

$$\mu = 2m + l - N + 2 - o_j.$$

Without any restriction we assume that $\kappa \leq \mu$. We must again distinguish several cases depending on the values of κ and μ, in particular whether each of these is negative, lies between 0 and $n_{j+1} - 1$ or is greater than n_{j+1}.

- Let us first assume that $\kappa \leq \mu < 0$ or $0 \leq \kappa \leq \mu < n_{j+1}$ or $n_{j+1} \leq \kappa \leq \mu$. Then it follows

$$\begin{aligned}|\rho(\kappa) - \rho(\mu)| = |\kappa - \mu| = |2(i-m) + k - l| &\geq 2|i-m| - |k-l| \\ &> b_{j+1} + 2N - N > b_{j+1}.\end{aligned}$$

- The second case is where $\kappa < 0 \leq \mu < n_{j+1}$. Here we have

$$\begin{aligned}0 \leq 2(i + m - |i-m|) &< 2(i + m - (\frac{b_{j+1}}{2} + N)) \\ &< 2(i + m - (\frac{b_{j+1}}{2} + N)) + 5 - 2o_j + k + l.\end{aligned}$$

  and consequently

$$|\rho(\kappa) - \rho(\mu)| = |1 + \kappa + \mu| = |5 + 2i + 2m + k + l - 2N - 2o_j| > b_{j+1}$$

- The case where $0 \leq \kappa < n_{j+1} \leq \mu$ is easily reduced to the second case by regarding

$$\tilde{\kappa} = n_{j+1} - 1 - \mu < 0$$

  and

$$\tilde{\mu} = n_{j+1} - 1 - \kappa \in \{0, \ldots, n_{j+1} - 1\}.$$

  It follows from above that

$$b_{j+1} < |\rho(\tilde{\kappa}) - \rho(\tilde{\mu})| = |-1 - \tilde{\kappa} - \tilde{\mu}| = |1 - 2n_{j+1} + \kappa + \mu| = |\rho(\kappa) - \rho(\mu)|.$$

- The last case to consider is where $\kappa < 0 < n_{j+1} \leq \mu$. Here we derive inequality (3.15) by noting that

$$\kappa = 2i + k - N + 2 - o_j \geq -N$$

and

$$\mu = 2m+l-N+2-o_j \leq 2(n_j-1)+(N-1)-N+2-o_j = 2n_j-1-o_j \leq n_{j+1}+N.$$

From these two inequalities we conclude that

$$\kappa - \mu + 2n_{j+1} \geq -N - (n_{j+1} + N) + 2n_{j+1} \geq n_{j+1} - 2N \geq b_{j+1}$$

and consequently

$$|\rho(\kappa) - \rho(\mu)| = |-1 - \kappa - (2n_{j+1} - \mu - 1)| = |\kappa - \mu + 2n_{j+1}| \geq b_{j+1}.$$

All in all, we have shown that for all levels $j \in \{0, \ldots, J-1\}$ the inequality

$$b_j \leq \frac{b_{j+1}}{2} + N,$$

holds. □

As in the periodic case, this lemma can be used to reduce the number of multiplications that are carried out in the implementation of the algorithm, and with regard to the order of the algorithm, we obtain the same statement:

Theorem 3.2. *With symmetric boundary conditions the algorithm is of order* $b_J \cdot 2^J \cdot N^2$.

Proof. We conclude from Lemma 3.2 that the number of non-zero diagonals in level j is bounded by

$$C = \max\{b_J, 2N\} + 1.$$

This means that the total number of new entries to be calculated does not exceed

$$\sum_{j=0}^{J-1} Cn_j \leq C2^J + CJN$$

using inequality (3.12). The calculation of each entry requires again N^2 multiplications, so like the periodic case it follows that our algorithm is of order $b_J \cdot 2^J \cdot N^2$. $\square$

3.4 Summary

In this chapter we introduced an algorithm which can be used to calculate the variances of wavelet coefficients when the covariance matrix Σ of the data points is known. The central idea was to calculate the covariance matrices of the smooth component and wavelet coefficients at each level separately rather than calculating the variances of the wavelet coefficient immediately. The computational complexity was analysed and it turned out that the algorithm performs in order n if Σ is a band matrix.

Chapter 4

Thresholding unequally time-spaced data

"The gates made fast! Brother, I like not this;

For many men that stumble at the threshold

Are well foretold that danger lurks within."

—William Shakespeare, *King Henry VI, Act IV, Scene vii*

We return now to our original problem. Remember that a function f on the interval $[0, 1]$ has to be estimated from noisy samples $y_1, \ldots, y_n$ at time points $t_1, \ldots, t_n$. We assume that $y_i = f(t_i) + \varepsilon_i$, where $\{\varepsilon_i\}$ is i.i.d. Gaussian white noise with mean 0 and variance σ^2. We have already seen that usual wavelet thresholding which requires 2^J equally spaced time points has considerable advantages, therefore, we would like to extend thresholding to the situation where the time points are unequally spaced.

In the first section of this chapter we review the contributions of other authors to this topic and discuss briefly some of their advantages and disadvantages. Then we introduce a new method for thresholding unequally spaced data: Calculate data on an equally spaced grid by connecting the data points with straight lines and evaluating them at the grid points. Then perform thresholding on the new "observations", but use *coefficient-*

dependent thresholds instead of a global threshold. This is reasonable because the noise of the gridded observations is now correlated, and therefore the noise of the wavelet coefficients is as well and does not have equal variances.

We have already seen in the last chapter how the variances of the wavelet coefficients can be calculated efficiently from the covariance matrix Σ of the observations. We still have to calculate Σ, but this is much easier, as we see in the third section. In the fourth section we describe an adaptation of SureShrink for our new situation. The new methods for thresholding unequally spaced data have been implemented in the wavelet package WaveThresh, and we describe the implementation in Section 4.5 together with some examples. Finally, we apply in Section 4.6 the new method to two real data sets.

4.1 Wavelet regression with unequally spaced data

We have seen that wavelet thresholding is a very promising approach in non-parametric regression. However, it requires that the number of data points is a power of two, and the time points must be spaced equally $t_i = i/n$. The first requirement is often met by extending the given data set to length 2^J using for example periodic extension or symmetric reflection (see [102]). At first sight, this looks a little bit artificial, but we have seen that the DWT itself makes use of boundary conditions. In fact, the results that are obtained with this simple techniques are usually very good.

There are other ways for dealing with data sets that are equally spaced in time but do not contain a power of two elements. Kwong and Tang (see [73]) as well as Taswell and McGill (see [113]) propose alternatives to the fast DWT that deal with sets of arbitrary length. These techniques are of particular interest in image compression.

The second condition raises more problems. One way could be to ignore the time structure by applying the usual threshold routines to the y_i. This would give us an estimate of f at the time points t_i. An estimate on a grid (which is desirable for time data with gaps inside) could be obtained by interpolating the result. Unfortunately, wavelet representations of irregularly spaced data are not as economical as they are if the time structure is regular, since the property of vanishing moments does in general not hold any more. As a consequence, the mean square error is relatively high, as we will see at the end of the next chapter.

We describe now several wavelet techniques that are designed to relax those conditions.

4.1.1 Retouching wavelet coefficients

Lenarduzzi (see [75]) analyses the approach, where thresholding is applied to an unequally spaced data set of size 2^J. She notices that the regression functions do not look smooth even if the signal is smooth, and therefore she suggests an algorithm which changes the already thresholded coefficients in order to get "visually more pleasant" results.

First a usual thresholding technique is applied to the data, ignoring the time structure. Afterwards, the wavelet coefficients $w_{j,k}^{thr}$ of the threshold estimator that are equal to zero are again modified in the following way: For a small subset of time points $(\tilde{t}_j)_1^K \subset (t_j)_1^n$ the cubic spline is calculated that interpolates the thresholding regression function at $\tilde{t}_j$. Several subsets are tried out, before one of these subsets is chosen, so that the corresponding spline s looks roughly like the thresholding regression function. Then the discrete wavelet transform is applied to the values of the spline s in the complete time set $(s(t_j))_1^n$ obtaining wavelet coefficients $w_{j,k}^{spl}$. Finally, coefficients $w_{j,k}^{new}$ are obtained by

$$w_{j,k}^{new} = \begin{cases} w_{j,k}^{spl}, & \text{if } w_{j,k}^{thr} = 0 \\ w_{j,k}^{thr}, & \text{otherwise} \end{cases}$$

and the inverse DWT is applied to $(w_{j,k}^{new})$. The resulting curves look visually better in her examples than those obtained by usual thresholding. However the author does not discuss the question of which subsets are tried out, and she does not give details of which of the splines is finally chosen. Finally, it is not likely that the refinements will decrease the mean square error, and even from a graphical viewpoint it is questionable if this method is really the best way for obtaining “graphically satisfying” curves.

4.1.2 Kernel-like wavelet regression

Another two algorithms are supplied by Antoniadis and Pham. They assume (see [6]) that either the time points are random with a density function g_1 or that they are fixed, but that their empirical distribution converges to a distribution with density g_1 as $n \to \infty$. Instead of estimating f directly they compute estimates of $g_2 = f \cdot g_1$ and g_1. To estimate g_1 and g_2 they use in one of the two algorithms the orthogonal projections that arise from a multiresolution analysis

$$P(g_i) = 2^j \sum_{k \in \mathbb{Z}} < \phi_{j,k}, g_i > \phi_{j,k}.$$

The function ϕ is, of course, the scaling function, and $\phi_{j,k}$ is derived from it by translating and scaling as described in Chapter 2. Then they replace the inner products by their empirical counterparts to obtain

$$\hat{g}_1 = 2^j \sum_{k \in \mathbb{Z}} \left(\frac{1}{n} \sum_{i=1}^{n} \phi_{j,k}(t_i) \right) \phi_{j,k}$$

and

$$\hat{g}_2 = 2^j \sum_{k \in \mathbb{Z}} \left(\frac{1}{n} \sum_{i=1}^{n} y_i \phi_{j,k}(t_i) \right) \phi_{j,k}.$$

Note that this algorithm is very fast. On the other hand, it is linear, and we are in the same situation as in Chapter 2. Essentially, the algorithm is a kernel estimator where instead of the bandwidth λ the smoothness is adjusted by the resolution level j. Large j yield rough estimates, small values for j lead to smooth curves with less sharp features.

The second algorithm which they provide is similar to the first, but is non-linear: They use the sums

$$\frac{1}{n} \sum_{i=1}^{n} \phi_{j,k}(t_i)$$

and

$$\frac{1}{n} \sum_{i=1}^{n} y_i \phi_{j,k}(t_i)$$

as starting values for the discrete wavelet transform. As usual they threshold the coefficients and invert the procedure. This approach is very similar to the method that we will introduce in the next chapter. However, Antoniadis and Pham do not give any details about their shrinkage method. It seems that they ignore completely the differences from the equally spaced setting, in particular that the variances of the wavelet coefficients are no longer constant.

The authors compare their method with spline smoothing and variable bandwidth local polynomial smoothing in two simulations and an application to a real data set. Unfortunately, they do not mention which of their algorithms was used. Their results show that the computational speed measured in CPU time is quite impressive (which is not surprising). On the other hand their estimator has in one of the two simulations the largest mean square error, and is second in the other. The figures provided show that the method produces curves that are more wiggly than the functions which result from the other methods. Moreover, it flattens two sharp features in one of the simulated signals.

Although Antoniadis and Pham's algorithms make use of wavelets, their algorithms have not the properties which are known from wavelet shrinkage in the equally spaced case. They look very similar to the linear estimators that we have described in Chapter 2.

4.1.3 Polynomial regression

Delyon and Juditsky [32] suggest an approach where values on a coarse grid are computed with polynomial regression:

1. Divide $[0,1]$ into 2^J subintervals $I_k = [2^{-J}k, 2^{-J}(k+1))$ where $k \in \{0,\ldots,2^J-1\}$. The resolution J is chosen as the biggest number such that every interval I_k contains at least $C \cdot \ln(n)$ time points, and C is a previously calculated constant.

2. Perform in each "cell" I_k a polynomial regression of order M, getting a polynomial p_k of degree $\leq M$.

3. Calculate the values of the polynomials at the centres of the intervals

$$\tilde{y}_k = p_k(2^{-J}(k+\frac{1}{2})).$$

4. Apply the discrete wavelet transform to the grid data $\tilde{y}_k$ using a wavelet with M vanishing moments.

Note that the new data set is coarser than the original data set. In particular, if d_{max} is the maximum distance between each two neighbouring time points

$$d_{max} = \max_{i \in \{1,\ldots,n\}} (t_i - t_{i-1})$$

(we assume that the t_i are ordered according to their size) then it follows that $2^{-J+1} > d_{max}$ because none of the subintervals I_k may be empty. Therefore, the length of the grid data set is smaller than $2/d_{max}$, no matter of how many observations the original data set consists. Finally, it must be remarked that the authors use their algorithm primarily for estimating wavelet coefficients and not smoothing noisy data. In particular, they do not discuss smoothing techniques, although the first three steps may be regarded as a (linear) algorithm for non-parametric regression.

4.2 A new algorithm for thresholding unequally spaced data

We have seen in the last section some approaches for dealing with unequally spaced data based on wavelets, but none of them shows satisfying results. Therefore, we suggest in this section a new method.

For a given resolution level J (suggestions on the choice of J are discussed later in this section) we create the new time points

$$\tilde{t}_k = (k + \frac{1}{2})2^{-J},$$

where $k \in \{0, \ldots, 2^J - 1\}$. We calculate a new "observation" $\tilde{y}_k$ by evaluating the straight line that interpolates the nearest data point on the left and on the right at $\tilde{t}_k$. More precisely,

$$
\tilde{y}_k = \begin{cases} y_0 & \text{if } \tilde{t}_k < t_0, \\ \\ y_{n-1} & \text{if } \tilde{t}_k \geq t_{n-1} \\ \\ y_i + (\tilde{t}_k - t_i) \cdot \frac{y_{i+1}-y_i}{t_{i+1}-t_i} & \text{if } t_i \leq \tilde{t}_k \leq t_{i+1}. \end{cases} \tag{4.1}
$$

If there are several integers i which satisfy $t_i \leq \tilde{t}_k \leq t_{i+1}$ in the third case, then the smallest of these values is chosen.

The linear function that maps for given $t_0, \ldots, t_{n-1}$ and resolution level J the original data to the grid data will be denoted by R, so that

$$
\tilde{y} = Ry.
$$

The remaining part of the algorithm does not make use of the special form of R. Therefore, we can easily employ other grid transformations instead of the linear interpolation scheme as long as they define a linear transformation between the original observations and the new data set. One possible improvement might be to use some sort of local binning over the grid interval and only use interpolation where the data are sparse. Another possible direction could be to interpolate the data by higher order polynomials: For a given integer μ and a grid point $\tilde{t}_k$ one could determine the nearest μ data points $y_i, y_{i+1}, \ldots, y_{i+\mu-1}$ and interpolate these points by a polynomial of order $\mu - 1$. The new grid point $\tilde{y}_k$ would be obtained by an evaluation of the polynomial at $\tilde{t}_k$. However, a detailed comparison between different interpolation schemes is left as a subject for future research.

A related problem is the choice of the number of grid points 2^J. Clearly the grid transform (4.1) does not make use of all data points unless the distance between two grid points is smaller than than the minimum distance between each two time points of the original sequence. This is demonstrated in Section 5.5 by an analysis of the mean squared error in a particular example where the mean squared error decreases with increasing J as long as there are unused data points and remains constant for larger values of J. On the other hand the computing time could be very large if the size of the grid was chosen in this manner. Moreover it is not clear what one should do if there were two observations at the same time point. For the remainder of this thesis we shall make a compromise by choosing J such that 2^J is the smallest power of two that is not smaller than n. However, a more sophisticated choice of the grid transform would probably raise this problem again since the methods that were sketched in the previous paragraph would make use of all data points even for small values of J. Therefore the choice of J has to be analysed in connection with the initial grid transform in future work.

We now carry on with the description of our new algorithm for thresholding unequally spaced data. The transformation of the original data to a grid yields a new data vector of size 2^J. Therefore we can apply the discrete wavelet transform to $\tilde{y}$ getting wavelet coefficients $d_{j,k}$

$$d = \mathcal{W}\tilde{y}.$$

It is clear that d is multivariate normally distributed with mean $\mathcal{W}Rf$ and covariance matrix Σ, where $f = (f_1, \ldots, f_n)$ and

$$\Sigma = \sigma^2 \cdot \mathcal{W}RR^T\mathcal{W}^T.$$

In particular, we can compute constants $\gamma_{j,k}$ depending only on the time points, the wavelet filter and the length of the grid such that

$$\sigma_{j,k}^2 = \operatorname{var} d_{j,k} = \sigma^2 \cdot \gamma_{j,k}. \tag{4.2}$$

This is essentially done by applying the algorithm that we studied in Chapter 3 to

$$\Sigma^J = RR^T,$$

and we explain in the next section how the matrix product above can be calculated very easily and efficiently.

Once we know the variances of the wavelet coefficients, we apply a thresholding function η to each wavelet coefficient yielding modified wavelet coefficients $\tilde{d}_{j,k}$

$$\tilde{d}_{j,k} = \eta(d_{j,k}, \tau_{j,k}),$$

where η is either the soft thresholding function

$$\eta_S(d_{j,k}, \tau) = \operatorname{sgn}(d_{j,k})(|d_{j,k}| - \tau)_+$$

or the hard thresholding function

$$\eta_H(d_{j,k}, \tau) = d_{j,k} \cdot I\{|d_{j,k}| \geq \tau\}.$$

In this thesis we consider two ways of choosing the thresholds $\tau_{j,k}$, i.e. versions of *VisuShrink* and *SureShrink* that were introduced by Donoho and Johnstone (see [39] and [40]). They have in common that they are calculated from the $\sigma_{j,k}$, for example our version of *VisuShrink* is simply given by

$$\tau_{j,k} = \sqrt{2\log(n)} \cdot \sigma_{j,k}. \tag{4.3}$$

Our adaptation of the SureShrink method is introduced in Section (4.4).

Finally, performing the inverse transform on $\tilde{d}$ gives us the estimate $\hat{f}$ for f:

$$\hat{f} = \mathcal{W}^T \tilde{d}.$$

The noise level σ is usually unknown. In the case where the time points are equally spaced, Donoho et al. (see [41]) have suggested the estimation of the the noise level by taking the median absolute deviation of the coefficients at the finest scale of resolution, and dividing by 0.6745.

However, in our setting it is necessary to divide each coefficient $d_{J-1,k}$ by $\sqrt{\gamma_{J-1,k}}$ beforehand. Moreover, it can happen that some of the $\gamma_{J-1,k}$ are zero, when all grid points that have influence on the coefficient $d_{J-1,k}$ lie between two original observations and hence on one straight line. In this case it follows from the property of vanishing moments and the linear interpolation that is used in the grid transform that both the wavelet coefficient $d_{j,k}$ and its variance factor $\gamma_{j,k}$ equal zero, as long as the wavelet basis is at least of order 1.

We take this into account and suggest the following estimator:

$$\hat{\sigma} = \mathrm{MAD}\{d_{J-1,k}/\sqrt{\gamma_{J-1,k}} : \gamma_{J-1,k} > 0.0001\}/0.6745. \tag{4.4}$$

4.3 Computation of the factors for the variances

We describe now how the constants $\gamma_{j,k}$ can be calculated. As we already pointed out, we only need to be concerned with the computation of $\Sigma^J = RR^T$, because the calculation of $\mathcal{W}\Sigma^J\mathcal{W}^T$ was the topic of Chapter 3.

Recall the definition of our grid transform R. It is clear that every new observation is an average over one (at the left and right edge) or two neighbouring observations.

Therefore, each row of R contains no more than two non-zero elements and their sum equals 1. Consequently, we can store the matrix R in two vectors q and r, each of length 2^J: We define q and r for each $i \in \{0, \ldots, 2^J - 1\}$ by

$$q_i = \min\{n-2, \min_{k \in \{0,\ldots,n-1\}} \{k : R_{i,k} \neq 0\}\} \tag{4.5}$$

and

$$r_i = R_{i,q_i}. \tag{4.6}$$

This can be easily done when constructing the grid data from the original observations.

Let us consider now what the matrix Σ^J looks like. Obviously, Σ^J is square and symmetric. Moreover, it is (usually) a band matrix, because for $i \in \{1, \ldots, n-3\}$ and $k \in \{0, \ldots, 2^J - 1\}$ with $t_i \leq \tilde{t}_k < t_{i+1}$ it is clear that the new grid observation y_k is stochastically independent of any y_l with $\tilde{t}_l < t_{i-1}$ or $\tilde{t}_l \geq t_{i+2}$. It can be argued similarly for the remaining values of k, where the corresponding grid point lies either left or right of all observations. Note that a large gap in the time points t_i leads to a large bandwidth of Σ^J because many of the new time points fall in the gap, so that all the new observations at these time points are correlated.

Let us now proceed with the description of the calculation of Σ^J. Since Σ^J is a band matrix it seems reasonable to store its values as

$$s^J_{d,k} = \Sigma^J_{k,d+k},$$

where $d \in \{0, \ldots, 2^J - 1\}$ and $k \in \{0, \ldots, 2^J - 1 - d\}$. Memory is only allocated for the diagonals of Σ^J that are not empty.

When computing the values of Σ^J, note that we obtain for each $0 \leq k \leq i \leq 2^J - 1$

$$s_{i-k,k} = \Sigma^J_{i,k} = \sum_{l=0}^{2^J-1} R_{i,l} R_{l,k} = \begin{cases} r_i \cdot (1 - r_k), & \text{if } q_i - q_k = 1, \\ r_i \cdot r_k + (1 - r_i) \cdot (1 - r_k), & \text{if } q_i - q_k = 0, \\ 0, & \text{otherwise.} \end{cases} \tag{4.7}$$

One way to carry out the computation of Σ^J can be set out as follows:

Algorithm 4.1.

$m := 0;$

for $i := 0$ **to** $2^J - 1$ **do**

  $k := m;$

  while $q_i - q_k \geq 2$ **do**

    $k := k + 1;$

  od

  $m := k;$

  for $k := k$ **to** i **do**

    if $q_i - q_k = 1$ **then** $s_{i-k,k} = r_i \cdot (1 - r_k);$

      else $s_{i-k,k} = r_i \cdot r_k + (1 - r_i) \cdot (1 - r_k);$

    fi

  od

od

4.4 An adaptation of SureShrink

We have already considered in Section 2.5 the threshold selection method *SureShrink* which was introduced by Donoho and Johnstone (see [40]) for the i.i.d. Gaussian error setting, and adapted for correlated noise by Johnstone and Silverman (see [67]). In this section we will adapt the procedure for the situation where the time points are unequally spaced. While the VisuShrink threshold is designed to produce noiseless reconstructions of the unknown signal, the threshold selection with SureShrink tries to obtain a low mean squared error

$$\sum_{i=1}^{2^J}\left(f(\tilde{t}_i)-\hat{f}(\tilde{t}_i)\right)^2.$$

The method is based on a result of Stein (see [104]) who showed that the expected mean square error of an estimator $\hat{w}=X+g(X)$ of w where $X\sim\mathcal{N}_n(w,\Sigma)$ may be written

$$\mathbb{E}\left(||\hat{w}-w||^2\right)=\mathbb{E}\left(\operatorname{tr}(\Sigma)+||g(X)||^2+2\operatorname{tr}\left(\Sigma\cdot Dg(x)\right)\right), \tag{4.8}$$

where $Dg(X)$ denotes the $n\times n$ matrix with entries $\partial g_i/\partial x_{j,k}(X)$. Suppose now that X are the wavelet coefficients that we obtain by applying the grid transform and the DWT to a noisy sample y. In the case of soft thresholding with thresholds $\tau=\{\tau_{j,k}\}$, g has components

$$g_{j,k}(x)=\begin{cases}-\tau_{j,k}, & x_{j,k}>\tau_{j,k},\\ -x_{j,k}, & |x_k|\le\tau_{j,k},\\ \tau_{j,k}, & x_{j,k}<-\tau_{j,k}.\end{cases}$$

Therefore, $g_{j,k}$ is a function of $x_{j,k}$ alone, and the matrix Dg in (4.8) is diagonal. From

(4.8) it follows that

$$\mathrm{SURE}(\tau;\ x) = \sum_{j,k} \left(\hat{\sigma}^2_{j,k} + \min(x^2_{j,k}, \tau^2_{j,k}) - 2\hat{\sigma}^2_{j,k} \cdot \{|x_{j,k}| \le \tau_{j,k}\} \right) \tag{4.9}$$

may be used as an estimate of the risk in (4.8):

$$\mathbb{E}\left(\mathrm{SURE}(\tau;\ x)\right) = \mathbb{E}\left(||\hat{w} - w||^2 \right).$$

If the estimates $\hat{\sigma}^2_{j,k}$ were replaced by the true values $\sigma^2_{j,k}$, then $\mathrm{SURE}(\tau;\ x)$ would be an unbiased estimate of this risk. In addition, this is not exactly the risk we are interested in because of the error involved in replacing the wavelet transform of the true grid values of the signal by the values w. Remember that w is the mean of X and thus is the vector of wavelet coefficients obtained by interpolating the values $f(t_i)$ to the regular grid and applying the DWT to the gridded values. Because of these approximations, we regard the unbiased risk property of the criterion $\mathrm{SURE}(\tau;\ x)$ as heuristic rather than rigorous justification for its use.

In using the unbiased risk criterion, we consider thresholds which are proportional to some power of the noise

$$\tau_{j,k} = \alpha \cdot \sigma^s_{j,k}$$

and define $S(\alpha;\ x) = \mathrm{SURE}(\alpha \cdot \sigma^s_{j,k};\ x)$. The SureShrink thresholds are now obtained by minimising $S(\alpha;\ x)$ over the range $[0, \sqrt{2\log(n)}]$. As explained in [40] this minimisation can be accomplished in $O(n\log(n))$ time. The performance of this SureShrink estimator is analysed in Section (5.9).

4.5 An implementation in WaveThresh

The algorithm for computing the variances was implemented in S-Plus, using its object-oriented facilities. The functions are intended to be used in combination with the wavelet package "WaveThresh" which was developed by Nason (see [85]) and is available from the Statlib archive. Both packages run under the free S-Plus clone R (see [63] for more information about this program), which can also be downloaded from the Statlib archive. The source code of the new functions is available from the web page

`http://feat.mathematik.uni-essen.de/~kovac/thesis.html`

Typically, the first procedure which is invoked by the user, is the `makegrid` function, which applies the grid transform (4.1) to irregularly spaced data. The S-Plus command is

```
> makegrid(time = (1:length(y)) - 0.5)/length(y),
  y,gridn=2^(floor(log(length(time) - 1, 2)) + 1))
```

where the given data `time` and `y` are numerical vectors and `gridn` an integer, specifying the number of grid points. It is not necessary that the entries of the time vector `time` are sorted.

The function `makegrid` returns an object of class `griddata` and consists of four components: Most interesting for the user are the new grid data, given by `gridt` and `gridy`. The other entries `G` and `Gindex` describe the matrix of the used grid transform and correspond to the vectors q and r, given by (4.5) and (4.6).

In the following example, a noisy sample is generated from a sine function at uniform distributed time points and the grid transform is plotted (see Figure 4.1) where the crosses

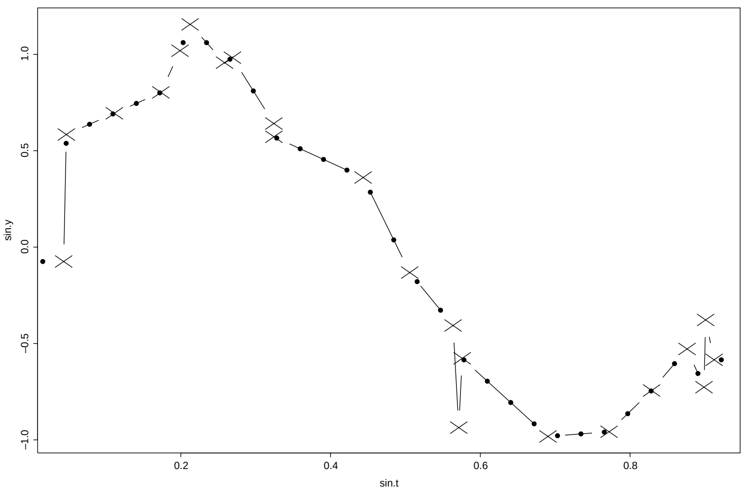

Figure 4.1: A simple example for the grid transform

are the original data and the black dots the new grid data.

```
> sin.t<-runif(22)

> sin.y<-sin(sin.t*2*pi)+rnorm(22,0,0.15)

> sin.grid<-makegrid(sin.t,sin.y,gridn=32)

> plot(sin.t,sin.y,pch=4)

> points(sin.grid$gridt,sin.grid$gridy,pch=0)
```

The second function is called `irregwd` and calculates the empirical wavelet coefficients of grid data as well as the factors for the variances of each coefficient.

```
function (gd, filter.number = 2, family = "DaubExPhase",

bc = "periodic", verbose = F)
```

Its arguments are the same as for the usual `wd` WaveThresh function (see [85]) except that `gd` is an `griddata` object as obtained by the `makegrid` function. It returns an object of class `irregwd` which looks mostly the same as an `wd` object, but has an extra

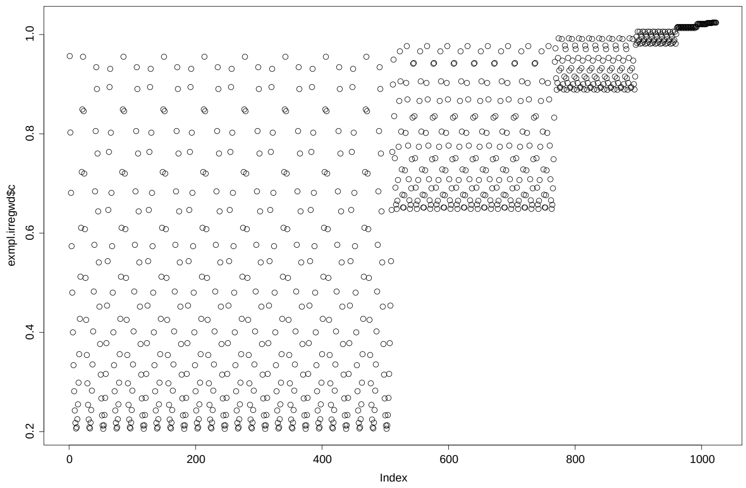

Figure 4.2: Example for the variance factors

component c. The entries of this vector are the factors for the variances of each wavelet coefficient as given by (4.2). In the next example, the coefficients are plotted for the case where 1000 regularly spaced data are transformed on a grid of size 1024 (see Figure 4.2)

```
> exmpl.t<-((1:1000)-0.5)/1000
> exmpl.y<-double(1000)
> exmpl.grid<-makegrid(exmpl.t,exmpl.y,1024)
> exmpl.irregwd<-irregwd(exmpl.grid)
> plot(exmpl.irregwd$c)
```

The coefficients can also be plotted by invoking the `plot` function with an `irregwd` object. The output is very similar to WaveThresh's `plot` function for wavelet coefficients. For example,

```
> plot(exmpl.irregwd,scal="global")
[1] 1.024028 1.024028 1.024028 1.024028 1.024028 1.024028
[7] 1.024028 1.024028 1.024028
```

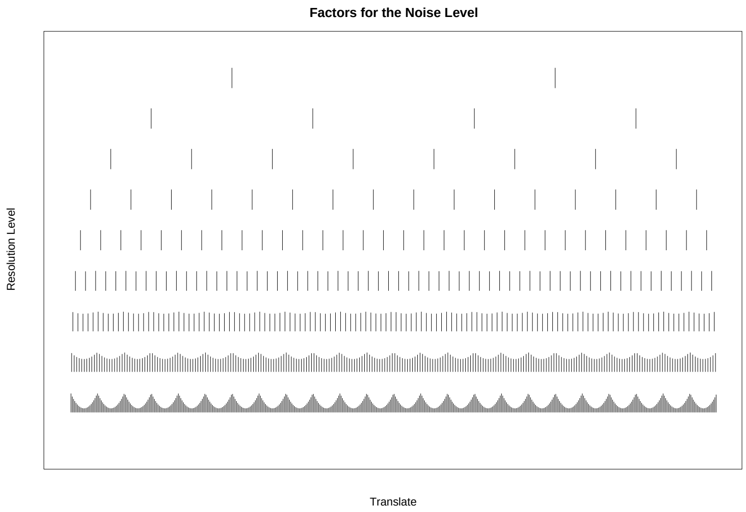

Figure 4.3: Another representation of the variance factors

produces Figure 4.3. See WaveThresh's manual page of the `plot` function for wavelet coefficients for more details like additional parameters.

Finally, `threshold` is defined for `irregwd` objects. It has all the arguments of the usual one-dimensional `threshold`-procedure and an extra argument `s` which is by default equal to 1. In this case, noise-proportional thresholds are used. If $s = 0$, a global threshold is applied instead. The noise level is estimated in the way described above.

4.6 Application to real data sets

It is always interesting to see how theoretical developments can be used to analyse real data sets. Therefore, we apply in this section our ideas of thresholding unequally spaced data to two data sets from the real world.

The first example (see [122]) deals with the waiting time between eruptions and the duration of the eruption of the Old Faithful geyser in Yellowstone National Park (see Figure 4.4). The geyser is one of the main attractions of the park, and the interval between two eruptions varies from 40 to 100 minutes. Therefore the National Park Service wanted to predict the time of the next eruption for the convenience of the visitors. It is apparent that a positive association exists between the two variables, with a longer interval until the next eruption following longer eruptions.

One problem with the application of the algorithm of the last sections is that the interval lengthes were rounded to one decimal place which resulted in several identical design points $t_i = t_{i+1} = \cdots = t_j$. Therefore the grid transform was slightly modified by the following way where all observations at the same design point are replaced by their arithmetical mean and the usual grid transform is applied, but the first step taken into account during the computation of the initial covariance matrix Σ. More precisely, we consider the new set of distinct design points $t'_1 < t'_2 < \cdots < t'_l$ where each original design point is included exactly once and define a vector z such that each element z_i is the inverse of the number of observations which have the design point t'_i:

$$z_i = 1/\#\{j \in \{1, \ldots, n\} : t'_i = t_j\}.$$

Moreover, we replace all observations with the same design point by their arithmetic average:

$$y'_i = \sum_{j=1}^{n} \{t'_i = t_j\} \cdot y_j \cdot z_i.$$

Then we apply the grid transform R as given by formula (4.1) to the new data set (t'_i, y'_i), obtaining grid observations $\tilde{y}_i$ and design points $\tilde{t}_i$. The initial covariance matrix Σ is now given by $\Sigma = \sigma^2 \cdot R \cdot z \cdot R^T$. Therefore, we have to adjust formula (4.7) such that each

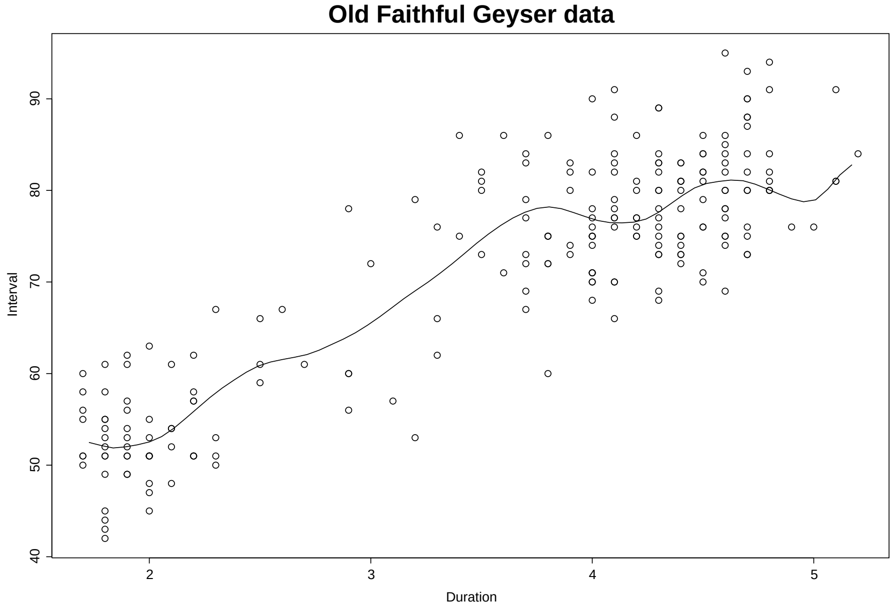

Figure 4.4: Waiting time between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park. The solid line was calculated with coefficient-dependent VisuShrink thresholds.

element $\Sigma^J_{i,k}$ is given by

$$s_{i-k,k} = \begin{cases} r_i \cdot z_i \cdot (1-r_k) \cdot z_{k+1}, & \text{if } q_i - q_k = 1, \\ r_i \cdot z_i \cdot r_k \cdot z_k + (1-r_i) \cdot z_{i+1} \cdot (1-r_k) \cdot z_{k+1}, & \text{if } q_i - q_k = 0, \\ 0, & \text{otherwise.} \end{cases}$$

Afterwards the variances of the coefficients of a wavelet decomposition of $\tilde{y}_i$ can be computed in the usual manner with the algorithm of the previous chapter.

This procedure was applied to the geyser data using a wavelet basis with six vanishing moments and symmetric boundary conditions. Soft thresholding was performed with the coefficient-dependent VisuShrink thresholds as defined in (4.3), and the result can also be seen in Figure 4.4. The curve looks very smooth as a consequence of the VisuShrink

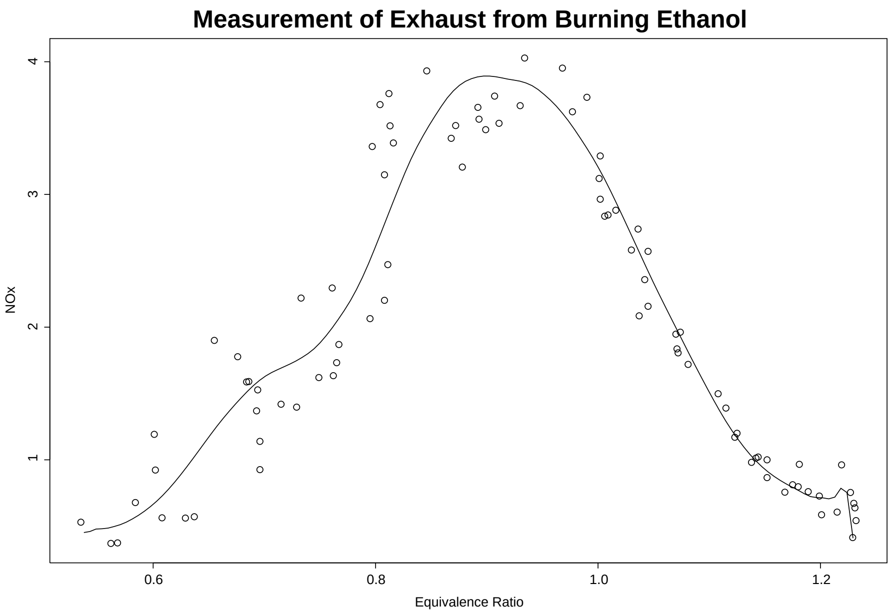

Figure 4.5: Measurement of exhaust from burning ethanol

threshold and its slope seems to be lower for durations above 3.7 than below this value. One could now apply simple linear regression to each of the intervals $[1, 2.7]$ and $[2.7, 6]$ separately—in fact, this has already be done by several authors who have analysed this data set with different smoothing techniques like splines and kernel estimators (see for example [99]).

The second data set has also been discussed by other authors (see [14], [24] and [57]). The data which are displayed in Figure 4.5 consist of 88 measurements from an experiment in which ethanol was burned in a single cylinder automobile test engine. The concentration of the sum of nitric oxide (NO) and nitrogen dioxide (NO_2) in engine exhaust, normalised by the work done by the engine, is related to the equivalence ratio, a measure of the richness of the air/ethanol mix. Hastie has fitted splines to this data set (see [57]). The original data set contains another predictor variable, the compression ratio of the engine. Hastie explains that the data cannot be modelled by a simple additive model and Cleveland et al. analyse the data set by means of localised polynomial regression techniques (see [24]).

We will now apply wavelet thresholding to the exhaust data. Like the previous data set we have to change our algorithm slightly to analyse the data more accurately. This time the problem is that the variance of the data does not appear to be constant because the data exhibit considerably more variation for smaller equivalence ratios. One possibility to cope with this situation is to estimate the variance of each data point separately and to take the new variances into account, similarly to the approach for the first data set. However, this topic is far beyond the scope of this thesis and an interesting field of possible future research.

We consider here a much simpler approach: We estimate the variance of the first half of the new grid points and of the second half separately, using formula (4.7). This yields variances of 0.4 for the left and 0.17 for the right half of the data. During the actual threshold step, the threshold of each coefficient is determined either as $\sqrt{2\log(128)} \cdot 0.4^2 \cdot c_{j,k}$ or $\sqrt{2\log(128)} \cdot 0.17^2 \cdot c_{j,k}$, depending on the data points where the coefficient is calculated from.

This approach was applied to the ethanol data, using the parameters as in the previous example with the exception of periodic instead of symmetric boundary conditions. The resulting curve is also plotted in Figure 4.5 and seems to fit the data very well.

Chapter 5

Thresholding and the Mean Square Error

"What a chimera, then, is man! What a novelty, what a monster,

what a chaos, what a subject of contradiction, what a prodigy!

A judge of all things, feeble worm of the earth, depositary of the truth,

cloaca of uncertainty and error, the glory and the shame of the universe!"

—Blaise Pascal, *Thoughts, Chap. x*

In the previous chapter we discussed a new method for thresholding unequally spaced data, but we have not yet described how the thresholds can be chosen. We will adapt several techniques which are known from the situation where the time points are equally spaced to the new situation. We also compare their performances by determining the expected mean square error for four specific data sets.

We start by deriving an exact formula for the risk. This has already been done by other authors in the equally spaced case (see for example [4] and [39]), and, of course, the formula reduces nicely to their results when the time points are equally spaced.

The next section is concerned with optimal thresholds, that minimise the expected risk for a given signal and known noise level. Such thresholds are, of course, not available in practice, but one can regard the mean square error that can be achieved with these thresholds as a lower bound when evaluating the performance of a threshold method. No threshold choice can do better than these thresholds in terms of L^2 error.

While Section 5.2 is concerned with optimal thresholds in the sense that we optimise each coefficient separately, we consider a somewhat more realistic example in Section 5.3. This time we assume that the thresholds are chosen *proportional* to the noise level of each coefficient, and we optimise the factor. All the results of the first three sections are used in Section 5.4 for a comparison between optimal thresholds and the performance of the *VisuShrink* estimator that we have introduced in Chapter 2 of the thesis for the equally spaced case. It turns out that the ratio between the *VisuShrink* thresholds and the optimal noise-proportional thresholds is approximately 3. But there is some evidence that there are better models than noise-proportional thresholds.

We compare in Section 5.5 optimal noise-proportional thresholds with optimal global thresholds and see an example where surprisingly global thresholds perform slightly better. We try to give reasons for this phenomenon in Section 5.6 where we generalise the results to thresholds that are proportional to a power of the noise level. We also have a look at hard thresholding in Section 5.7, but cannot find substantial differences with regard to the expected mean square error. We finish this Chapter in Section 5.8 with a comparison between several estimators in the form of a small simulation study.

5.1 Calculating the risk

When proposing some specific choices for the threshold, it is important to evaluate how well a threshold works for a given data set. One way of doing that is to calculate the mean square error which is defined as

$$\mathrm{MSE}(\hat{f}) = \frac{1}{2^J} \sum_{j=1}^{2^J} (f_i - \hat{f}_i)^2$$

Here, f and $\hat{f}$ are vectors of length 2^J. We will assume throughout this chapter that periodic boundary conditions are applied. In this case, the wavelet transform is orthogonal, and therefore, using Parseval's equation, the mean square error can also be calculated as

$$\mathrm{MSE}(\hat{f}) = \frac{1}{2^J} \sum_{j=j_0}^{J-1} \sum_{k=0}^{2^j-1} \Big((\mathcal{W}f)_{j,k} - \eta_S((\mathcal{W}R(\tilde{f}+\varepsilon))_{j,k}, \tau_{j,k}) \Big)^2, \tag{5.1}$$

where $\tau_{j,k}$ are the coefficient-dependent thresholds, j_0 the "cut-off-level" and $\tilde{f}$ the true values of f at the time points t_i:

$$\tilde{f} = (f(t_i))_{i=1}^n$$

on a grid of length n. We are interested in the risk which is simply the expectation of the mean square error.

Let us first think about how to calculate the risk, when the signal f, the time points t_i and the noise level σ are all given. It is clear that $(\mathcal{W}R(\tilde{f}+\varepsilon))_{j,k}$ is normally distributed with mean $(\mathcal{W}R\tilde{f})_{j,k}$ and variance $\sigma_{j,k}^2$ which can be calculated with the algorithms that were introduced above. Therefore, we only have to explore the expectation

$$\rho(\tau;\ \mu_1, \mu_2, \sigma) = \mathbb{E}(\mu_1 - \eta_S(X, \tau))^2, \tag{5.2}$$

where X is a normally distributed random variable with mean μ_2 and variance σ^2. The risk is then given by

$$\mathbb{E}\Big(\mathrm{MSE}(\hat{f})\Big) = \frac{1}{2^J} \sum_{j=j_0}^{J-1} \sum_{k=0}^{2^j-1} \rho(\tau_{j,k};\ (\mathcal{W}f)_{j,k}, (\mathcal{W}R\tilde{f})_{j,k}, \sigma_{j,k}). \tag{5.3}$$

Let ϕ and Φ be density and distribution function of a standard normal distribution. Using the definition of the thresholding function, we get

$$\begin{aligned}
\rho(\tau;\ \mu_1, \mu_2, \sigma) &= \mathbb{E}(\mu_1 - \eta_S(X,\tau))^2 \\
&= \int (\mu_1 - \eta_S(x,\tau))^2 \tfrac{1}{\sigma}\phi\left(\tfrac{x-\mu_2}{\sigma}\right) dx \\
&= \int (\mu_1 - \eta_S(\sigma y + \mu_2, \tau))^2 \phi(y) dy \\
&= \int_{-\infty}^{\frac{-\tau-\mu_2}{\sigma}} (\mu_1 - \sigma y - \mu_2 - \tau)^2 \phi(y) dy \\
&\quad + \int_{\frac{-\tau-\mu_2}{\sigma}}^{\frac{\tau-\mu_2}{\sigma}} {\mu_1}^2 \phi(y) dy \\
&\quad + \int_{\frac{\tau-\mu_2}{\sigma}}^{\infty} (\mu_1 - \sigma y - \mu_2 + \tau)^2 \phi(y) dy.
\end{aligned} \tag{5.4}$$

It is well known and easy to see that

$$\int_{-\infty}^{s} y\phi(y)dy = -\phi(s) \tag{5.5}$$

and

$$\int_{-\infty}^{s} y^2\phi(y)dy = \Phi(s) - s\phi(s). \tag{5.6}$$

By combining these equations with (5.4) we obtain

$$\begin{aligned}
\rho(\tau;\ \mu_1, \mu_2, \sigma) = \sigma^2 &\left(\Phi(\tfrac{-\tau-\mu_2}{\sigma}) - (\tfrac{-\tau-\mu_2}{\sigma})\phi(\tfrac{-\tau-\mu_2}{\sigma})\right) - 2\sigma(\tau + \mu_2 - \mu_1)\phi(\tfrac{-\tau-\mu_2}{\sigma}) \\
&+ (\tau + \mu_2 - \mu_1)^2 \Phi(\tfrac{-\tau-\mu_2}{\sigma}) \\
&+ {\mu_1}^2 (\Phi(\tfrac{\tau-\mu_2}{\sigma}) - \Phi(\tfrac{-\tau-\mu_2}{\sigma})) \\
&+ \sigma^2 \left(\Phi(\tfrac{\mu_2-\tau}{\sigma}) - (\tfrac{\mu_2-\tau}{\sigma})\phi(\tfrac{\mu_2-\tau}{\sigma})\right) - 2\sigma(\tau + \mu_1 - \mu_2)\phi(\tfrac{\mu_2-\tau}{\sigma}) \\
&+ (\tau + \mu_1 - \mu_2)^2 \Phi(\tfrac{\mu_2-\tau}{\sigma})
\end{aligned} \tag{5.7}$$

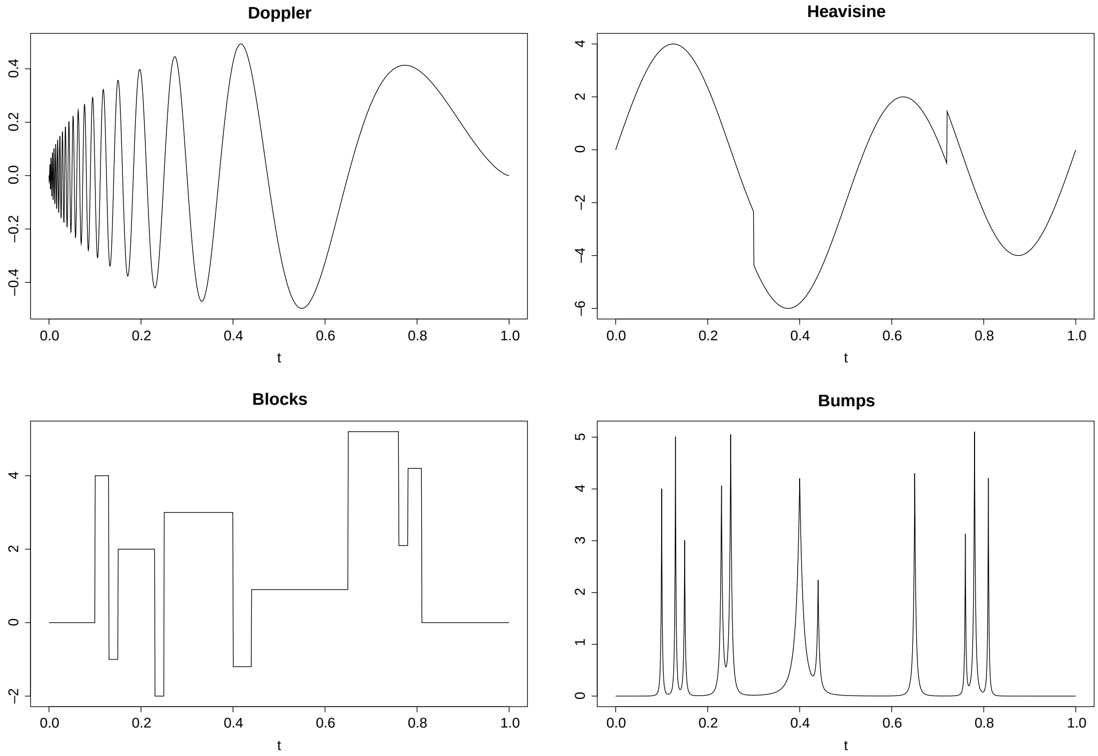

Figure 5.1: Donoho and Johnstone's test signals: Doppler, Heavisine, Blocks, Bumps

$$
\begin{aligned}
&= \mu_1^2 + (\sigma^2 + (\tau + \mu_2 - \mu_1)^2 - \mu_1^2) \cdot \Phi(\tfrac{-\mu_2 - \tau}{\sigma}) \\
&+ (\sigma^2 + (\tau + \mu_1 - \mu_2)^2 - \mu_1^2) \cdot \Phi(\tfrac{\mu_2 - \tau}{\sigma}) \\
&\quad + \sigma \cdot \Big((2\mu_1 - \mu_2 - \tau)\phi(\tfrac{-\mu_2 - \tau}{\sigma}) - (2\mu_1 - \mu_2 + \tau)\phi(\tfrac{\mu_2 - \tau}{\sigma}) \Big) \, . \qquad (5.8)
\end{aligned}
$$

This formula coincides nicely with results that were obtained by other authors in the conventional setting where $\mu_1 = \mu_2$ (see [4] and [39]).

5.2 Regular grids of arbitrary length

In this section we consider the performance of the methods on data where the t_i are regularly placed but the number of points is not necessarily a power of two. The basic idea is to use our algorithm to map the data to a grid of length 2^J so that standard DWT implementations can be used. In this context, we use our exact risk formula to examine the mean square error for four test signals *Doppler*, *Heavisine*, *Blocks* and *Bumps* (see

Figure 5.1) that were analysed by Donoho (1993a) and Donoho and Johnstone (1994). Rescaled versions of the signals were used such that

$$\sum_{i=1}^{2048} (f_i - \bar{f})^2 = 10000$$

where $f_i = f((i + 0.5)/2048)$ and $\bar{f} = \frac{1}{2048} \sum_{i=1}^{2048} f_i$. The time points were given by

$$t_i = (i + 0.5)/n,$$

but with their number n taking each value in $\{17, \ldots, 2048\}$. The size of the grid $(\tilde{t}_j)$ was chosen to be the smallest power of two not less than n. The noise level σ was chosen as 0.35 (corresponding to a signal-to-noise ratio $SD(f)/\sigma \approx 6.3$) and the threshold chosen with *VisuShrink*, so that

$$\tau_{jk} = \sqrt{2\log(n)} \cdot \sigma_{jk},$$

assuming the variances to be known.

Finally, Daubechies' wavelet with two vanishing moments, periodic boundary conditions and a cut-off-point j_0 of 3 were used. The results are plotted as solid lines in Figure 5.2.

We can get a lot of information from the figure. First of all, the expected mean square error decays very fast. This was one reason why we chose logarithmic scales on both axes. Note further that the risk does not decrease monotonically and that the amount of variation is different among the four test signals. The variation in the risk of the Heavisine and Doppler data is relatively small while the Blocks and Bumps data exhibit a lot more oscillation, in particular for small n with the Bumps data and for large n with the Blocks data. Finally, the risk seems to be exceptionally small if n is a power of two.

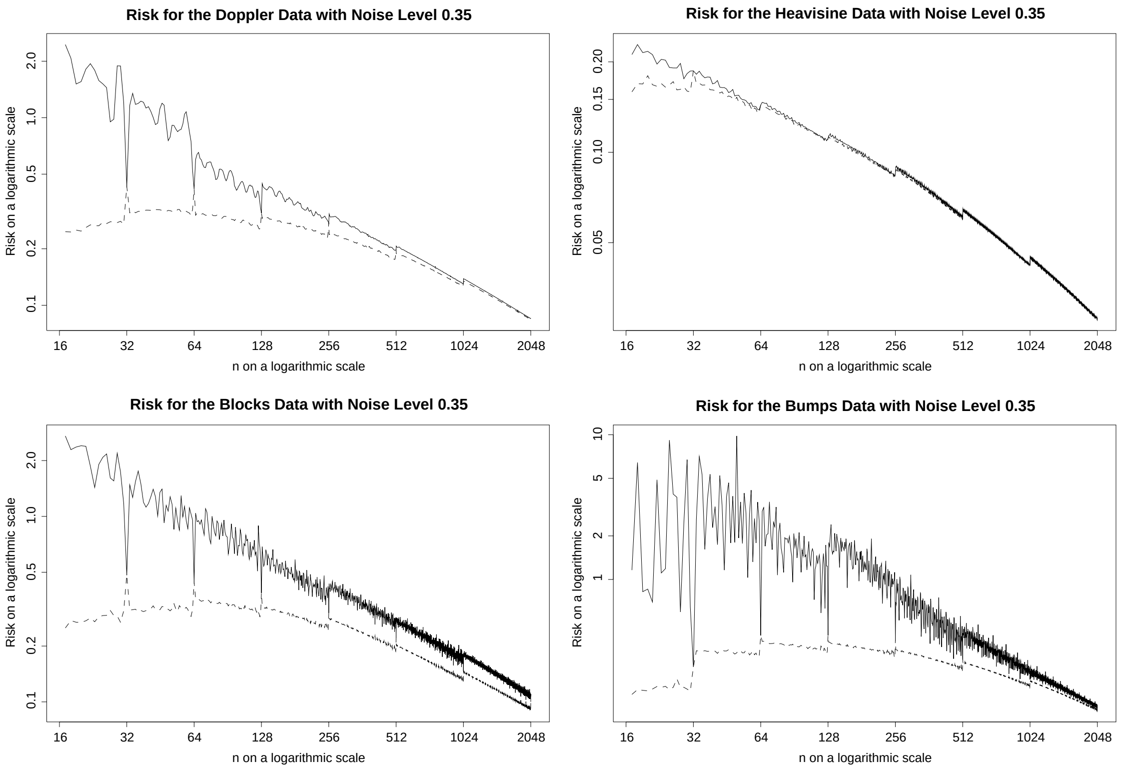

Figure 5.2: The risk for the test functions and *VisuShrink* on regular grids. Also shown as dashed lines is the "unbiased" risk which represents the amount of error which is caused by the actual thresholding step and does not contain the bias which is caused by the grid transform. The risk and the number of data points are always plotted on logarithmic scales.

All these properties are due to the bias that is caused by the grid transform

$$\frac{1}{2^J}\|f - R\tilde{f}\|_2^2.$$

In particular, the Blocks data are very sensitive to small changes in the time structure because they have a lot of discontinuities. This explains why the variation is considerable even for large n.

On the other hand, the risk for the Bumps Data depends strongly on how well its peaks can be approximated. For small numbers n_1 and n_2 the goodness of this approximation can obviously be very different, even if n_1 and n_2 are close together. However, for larger n_1 and n_2 the continuity of the Bumps function ensures that the approximations are similar, so that the amount of variation is relatively small for larger sampling rates.

Similar arguments can be put forward for the Heavisine and Doppler signals which are much smoother than the other two signals. Finally, the low values for powers of two that can be seen very clearly for the Blocks data are also caused by the absence of bias.

These statements are confirmed by the dotted lines in figure 5.2 which represent the expected mean square error after "removing the bias", obtained by assuming that the wavelet coefficients of the grid points are the true coefficients:

$$\frac{1}{2^J}\sum_{j=j_0}^{J-1}\sum_{k=0}^{2^j}\mathbb{E}\left((\mathcal{W}R\tilde{f})_{jk} - \eta_S(w_{jk}, \tau_{jk})\right)^2 \tag{5.9}$$

Here, w_{jk} is normally distributed with mean $(\mathcal{W}R\tilde{f})_{jk}$ and variance σ_{jk}^2.

Note that this effect and the bias of the grid transform do not add up exactly to the total expected mean square error: If we look at one coefficient and use the notation that was introduced above, then we can write

$$\mathbb{E}(\mu_1 - \eta_S(X,\tau))^2 = (\mu_1 - \mu_2)^2 + \mathbb{E}(\eta_S(X,\tau) - \mu_2)^2 + 2\cdot(\mu_1 - \mu_2)\cdot(\mu_2 - \mathbb{E}(\eta_S(X,\tau))).$$

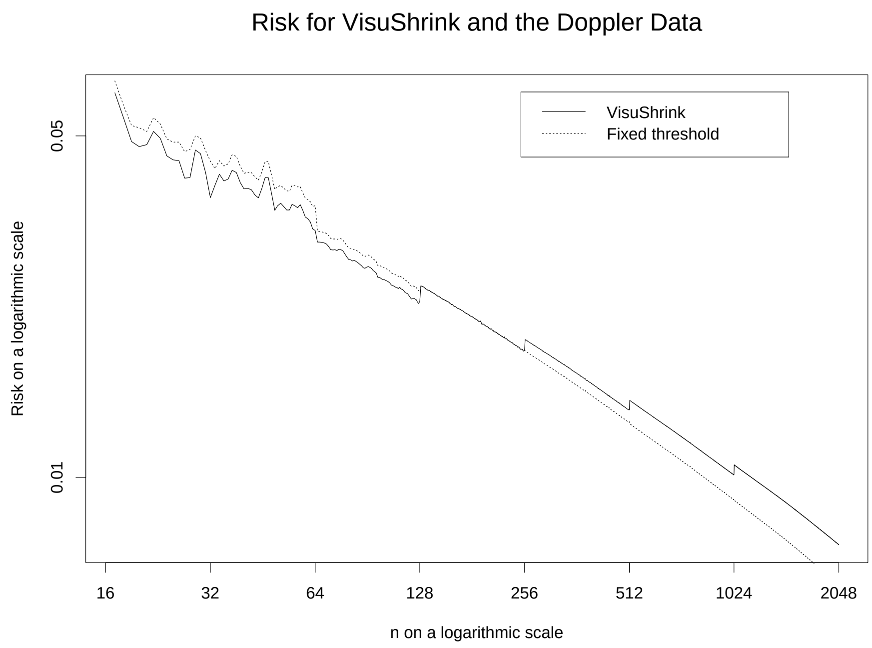

Figure 5.3: The risk for the Doppler data with *VisuShrink* compared with a fixed threshold choice on regular grids. The number of data points n and the risk are both plotted on logarithmic scales.

The first term is due to the "grid transform bias", which we obtain by summing up the first terms over all coefficients because of Parseval's identity, while summing up the second terms yield the remaining risk as defined in formula (5.9). Finally, one may hope that the effect of the third term is negligible as positive and negative terms cancel each other out when summing up, and indeed this was the case with the examples.

The remaining risk looks much smoother than the total risk, so indeed all the variation is due to the bias of the grid transform. Moreover, we can see that for powers of two both curves take the same value, because the grid transform is then the identity function and causes no bias at all.

Finally, there can be seen another effect in the figure near powers of two, in particular for the Heavisine and the Doppler Data where the variation is small: The risk functions exhibit small steps near those values. They can be explained by the values of the thresholds which depend explicitly on the number of grid points and increase for every n which is equal to $2^j + 1$ for some j. We will see later that visual thresholds are usually larger than "optimal" thresholds, which minimise the expected mean square error. Therefore, it is not surprising that the increment of the threshold enlarges the risk even more. This can also be seen in Figure 5.3. The solid line shows the risk for the Doppler data using VisuShrink as already seen in Figure 5.2. The dashed line represents the risk for fixed thresholds

$$\tau_{jk} = \sqrt{2\log(256)} \cdot \sigma_{jk}.$$

Note that the steps have disappeared.

We will see at the end of Section 5.5 another effect which occurs near powers of two and has in particular influence on the Bumps signal.

Before we finish this section we remark that we checked formula (5.7) by S-PLUS simulation in the following way. We evaluated for a given number of data points n the signal function f at the grid points $t_i = (i+0.5)/n$ ($i \in \{0, \ldots, n-1\}$). Then we created K samples ε^k of size n from a normal distribution with mean 0 and variance σ^2. Each sample was added to the function values, the grid transformation followed by the usual threshold procedure with *VisuShrink* were applied, and finally the mean squared error m_k was calculated. Averaging over the m_k for $k \in \{1, \ldots, K\}$ yielded an estimate for the risk.

Table 5.1 shows some results for the Doppler function (K was chosen as 1000). The difference is always smaller than one per cent of the risk.

n	17	18	19	20	127	128	129
Formula	0.04254	0.03600	0.02664	0.02728	0.00629	0.00602	0.00833
Simulation	0.04253	0.03610	0.02662	0.02731	0.00633	0.00603	0.00836

Table 5.1: Calculation of the Mean Square Error with formula and simulation for the Doppler data and several equally spaced time sets

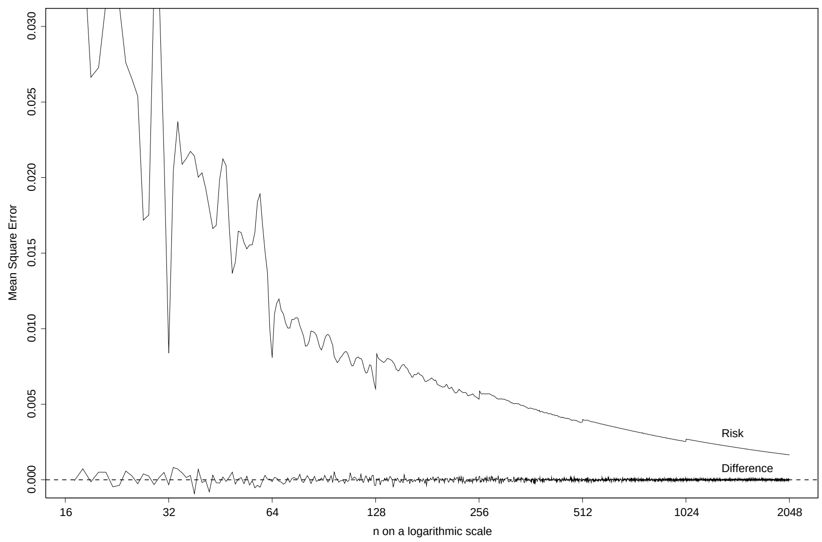

Figure 5.4: The true risk function for the Doppler signal and the difference from a simulated version

Additionally we simulated the risk for the Doppler function with $\sigma = 0.05$ and *VisuShrink.* For each value of n the mean over 10 replications was calculated. Figure 5.4 shows that the difference between the risk function and the simulated version is irrelevant.

5.3 Optimal coefficient-dependent thresholds

When comparing different choices for the thresholds it is interesting to ask for the minimal risk that can be obtained for a specific function f. In practice, of course, such optimal thresholds are not available because the signal is unknown, but at least they do give a reference point against which a practical threshold choice can be judged.

So suppose that we know the signal f and want to find optimal *coefficient-dependent* thresholds $\tau_{j,k} \geq 0$ such that the risk becomes minimal—other types of optimal thresholds are discussed in the following sections. It is obvious from equation (5.3) that the problem can be reduced to the minimisation of $\rho(\tau;\ \mu_1, \mu_2, \sigma)$ in equation (5.2) for fixed μ_1, μ_2 and σ over all $\tau \geq 0$.

Using (5.7), the first derivative of ρ (with respect to τ) is given by

$$
\begin{aligned}
&\rho'(\tau;\, \mu_1, \mu_2, \sigma) \\
&\quad = 2(\tau + \mu_2 - \mu_1)\Phi(\tfrac{-\mu_2-\tau}{\sigma}) + (\sigma^2 + (\tau + \mu_2 - \mu_1)^2 - \mu_1^2) \cdot (-\tfrac{1}{\sigma})\phi(\tfrac{-\mu_2-\tau}{\sigma}) \\
&\qquad + 2(\tau + \mu_1 - \mu_2)\Phi(\tfrac{\mu_2-\tau}{\sigma}) + (\sigma^2 + (\tau + \mu_1 - \mu_2)^2 - \mu_1^2) \cdot (-\tfrac{1}{\sigma})\phi(\tfrac{\mu_2-\tau}{\sigma}) \\
&\qquad + \sigma\left(-\phi(\tfrac{-\mu_2-\tau}{\sigma}) + (2\mu_1 - \mu_2 - \tau) \cdot (-\tfrac{1}{\sigma}) \cdot (\tfrac{\mu_2+\tau}{\sigma})\phi(\tfrac{-\mu_2-\tau}{\sigma})\right) \\
&\qquad - \sigma\left(\phi(\tfrac{\mu_2-\tau}{\sigma}) + (2\mu_1 - \mu_2 + \tau) \cdot (-\tfrac{1}{\sigma}) \cdot (\tfrac{\tau-\mu_2}{\sigma})\phi(\tfrac{\mu_2-\tau}{\sigma})\right) \\
&\quad = 2(\tau + \mu_2 - \mu_1)\Phi(\tfrac{-\mu_2-\tau}{\sigma}) + 2(\tau + \mu_1 - \mu_2)\Phi(\tfrac{\mu_2-\tau}{\sigma}) \\
&\qquad - 2\sigma\phi(\tfrac{-\mu_2-\tau}{\sigma}) - 2\sigma\phi(\tfrac{\mu_2-\tau}{\sigma}) \qquad\qquad (5.10) \\
&\qquad - \frac{1}{\sigma}\left((\tau + \mu_2 - \mu_1)^2 - \mu_1^2 + (2\mu_1 - \mu_2 - \tau)(\mu_2 + \tau)\right)\phi(\tfrac{-\mu_2-\tau}{\sigma}) \\
&\qquad - \frac{1}{\sigma}\left((\tau + \mu_1 - \mu_2)^2 - \mu_1^2 + (2\mu_1 - \mu_2 + \tau)(\tau - \mu_2)\right)\phi(\tfrac{\mu_2-\tau}{\sigma}) \\
&\quad = 2(\tau + \mu_2 - \mu_1)\Phi(\tfrac{-\mu_2-\tau}{\sigma}) + 2(\tau + \mu_1 - \mu_2)\Phi(\tfrac{\mu_2-\tau}{\sigma}) \\
&\qquad - 2\sigma\phi(\tfrac{-\mu_2-\tau}{\sigma}) - 2\sigma\phi(\tfrac{\mu_2-\tau}{\sigma}).
\end{aligned}
$$

It does not seem possible to find closed formulas for the solution of

$$\rho'(\tau;\ \mu_1, \mu_2, \sigma) = 0,$$

so we plan to find the roots of ρ' by Newton iteration.

But let us first think about existence and uniqueness of positive roots of ρ'. We prove the following lemma:

Lemma 5.1. *If* $\mu_1 \cdot \mu_2 \le 0$ *then* ρ' *is strictly negative on* $[0, \infty[$. *If* $\mu_1 \cdot \mu_2 > 0$ *and* $\rho'(0) \ge 0$ *then* ρ' *is strictly positive on* $[0, \infty[$. *Otherwise,* ρ' *has exactly one positive root.*

Proof. We need for the proof the second and third derivatives of ρ. Let us start with the second derivative:

$$\begin{aligned}
&\rho''(\tau;\, \mu_1, \mu_2, \sigma)\\
&\quad= 2\Phi(\tfrac{-\mu_2-\tau}{\sigma}) + 2(\tau + \mu_2 - \mu_1) \cdot (-\tfrac{1}{\sigma})\phi(\tfrac{-\mu_2-\tau}{\sigma})\\
&\qquad + 2\Phi(\tfrac{\mu_2-\tau}{\sigma}) + 2(\tau + \mu_1 - \mu_2) \cdot (-\tfrac{1}{\sigma})\phi(\tfrac{\mu_2-\tau}{\sigma})\\
&\qquad - 2\sigma(-\tfrac{1}{\sigma}) \cdot (\tfrac{\tau+\mu_2}{\sigma})\phi(\tfrac{-\mu_2-\tau}{\sigma}) - 2\sigma(-\tfrac{1}{\sigma}) \cdot (\tfrac{\tau-\mu_2}{\sigma})\phi(\tfrac{\mu_2-\tau}{\sigma})\\
&\quad= 2\Phi(\tfrac{-\mu_2-\tau}{\sigma}) + 2\Phi(\tfrac{\mu_2-\tau}{\sigma}) - \tfrac{2}{\sigma}(\tau + \mu_2 - \mu_1 - (\tau + \mu_2))\phi(\tfrac{-\mu_2-\tau}{\sigma}) \qquad (5.11)\\
&\qquad - \tfrac{2}{\sigma}(\tau + \mu_1 - \mu_2 - (\tau - \mu_2))\phi(\tfrac{\mu_2-\tau}{\sigma})\\
&\quad= 2\Phi(\tfrac{-\mu_2-\tau}{\sigma}) + 2\Phi(\tfrac{\mu_2-\tau}{\sigma}) + 2 \cdot \tfrac{\mu_1}{\sigma}\phi(\tfrac{-\mu_2-\tau}{\sigma}) - 2 \cdot \tfrac{\mu_1}{\sigma}\phi(\tfrac{\mu_2-\tau}{\sigma}).
\end{aligned}$$

Now we calculate the third derivative of ρ:

$$\begin{aligned}
&\rho'''(\tau;\, \mu_1, \mu_2, \sigma)\\
&\quad= 2\phi(\tfrac{-\mu_2-\tau}{\sigma}) \cdot (-\tfrac{1}{\sigma}) + 2\phi(\tfrac{\mu_2-\tau}{\sigma}) \cdot (-\tfrac{1}{\sigma}) + 2\tfrac{\mu_1}{\sigma}(\tfrac{\mu_2+\tau}{\sigma})\phi(\tfrac{-\mu_2-\tau}{\sigma})\\
&\qquad - 2\tfrac{\mu_1}{\sigma}(\tfrac{\tau-\mu_2}{\sigma})\phi(\tfrac{\mu_2-\tau}{\sigma})\\
&\quad= -\tfrac{2}{\sigma}\left(\phi(\tfrac{-\mu_2-\tau}{\sigma}) + \phi(\tfrac{\mu_2-\tau}{\sigma}) + \tfrac{\mu_1}{\sigma}(\tfrac{\mu_2+\tau}{\sigma})\phi(\tfrac{-\mu_2-\tau}{\sigma}) + \tfrac{\mu_1}{\sigma}(\tfrac{\mu_2-\tau}{\sigma})\phi(\tfrac{\mu_2-\tau}{\sigma})\right) \qquad (5.12)\\
&\quad= -\tfrac{2}{\sigma}\left(1 + \tfrac{\mu_1(\mu_2+\tau)}{\sigma^2}\right)\phi(\tfrac{-\mu_2-\tau}{\sigma}) - \tfrac{2}{\sigma}\left(1 + \tfrac{\mu_1(\mu_2-\tau)}{\sigma^2}\right)\phi(\tfrac{\mu_2-\tau}{\sigma}).
\end{aligned}$$

Obviously, we have to distinguish three cases, depending on the signs of μ_1 and μ_2, in particular whether their product is greater than, equal to or less than zero.

Let us now first regard the case where $\mu_1 \cdot \mu_2 < 0$. We claim that ρ' is strictly negative on $[0, \infty[$. It is easy to see that

$$\rho'(\tau;\ \mu_1, \mu_2, \sigma) = \rho'(\tau;\ -\mu_1, -\mu_2, \sigma),$$

so that we only need to consider the case where $\mu_1 < 0 < \mu_2$. Note first that obviously for $\tau > 0$

$$\phi(\tfrac{-\mu_2-\tau}{\sigma}) < \phi(\tfrac{\mu_2-\tau}{\sigma})$$

and consequently

$$\rho''(\tau;\ \mu_1, \mu_2, \sigma) > 2 \cdot \tfrac{\mu_1}{\sigma}(\phi(\tfrac{-\mu_2-\tau}{\sigma}) - \phi(\tfrac{\mu_2-\tau}{\sigma})) > 0,$$

so that ρ' is an isotone function on $[0, \infty[$. On the other hand it is clear from formula (5.10) that ρ' converges to 0 when $\tau \to \infty$ because $\tau\Phi(\tau)$ converges to 0, too. Therefore, ρ' is strictly negative on $[0, \infty[$.

If $\mu_1 = 0$ or $\mu_2 = 0$ then equation (5.11) reduces to the sum of the first two terms which are both positive, so that ρ' is again isotone on $[0, \infty[$.

We regard now the case where $\mu_1 \cdot \mu_2 > 0$ in more detail. Without restriction (see above) we can assume that $\mu_1 > 0$ and $\mu_2 > 0$. Using the definition of the standard normal density function ϕ we can transform the third derivative of ρ as follows:

$$\begin{aligned}
\rho'''(\tau;\ \mu_1, \mu_2, \sigma) &= -\tfrac{2}{\sigma}\left(1 + \tfrac{\mu_1(\mu_2+\tau)}{\sigma^2}\right)\phi(\tfrac{-\mu_2-\tau}{\sigma}) - \tfrac{2}{\sigma}\left(1 + \tfrac{\mu_1(\mu_2-\tau)}{\sigma^2}\right)\phi(\tfrac{\mu_2-\tau}{\sigma}) \\
&= -\tfrac{2}{\sigma} \cdot \tfrac{1}{\sqrt{2\pi}} \cdot \exp\left(-\tfrac{1}{2}\left(\tfrac{(\mu_2-\tau)^2}{\sigma^2}\right)\right) \\
&\quad \cdot \left(\left(1 + \tfrac{\mu_1(\mu_2+\tau)}{\sigma^2}\right) + \left(1 + \tfrac{\mu_1(\mu_2-\tau)}{\sigma^2}\right)\exp\left(\tfrac{\tau\mu_2}{\sigma^2}\right)\right) \\
&= -\tfrac{2}{\sigma} \cdot \tfrac{1}{\sqrt{2\pi}} \cdot \exp\left(-\tfrac{1}{2}\left(\tfrac{(\mu_2-\tau)^2}{\sigma^2}\right)\right) \cdot \left(1 + \tfrac{\mu_1(\mu_2+\tau)}{\sigma^2}\right) \qquad (5.13) \\
&\quad \cdot \left(1 + \frac{1 + \frac{\mu_1(\mu_2-\tau)}{\sigma^2}}{1 + \frac{\mu_1(\mu_2+\tau)}{\sigma^2}} \cdot \exp\left(\tfrac{\tau\mu_2}{\sigma^2}\right)\right)
\end{aligned}$$

To simplify this expression, define

$$C(\tau) = \tfrac{2}{\sigma} \cdot \tfrac{1}{\sqrt{2\pi}} \cdot \exp\left(-\tfrac{1}{2}\left(\tfrac{(\mu_2-\tau)^2}{\sigma^2}\right)\right) \cdot \left(1 + \tfrac{\mu_1(\mu_2+\tau)}{\sigma^2}\right) > 0.$$

Equation (5.13) then becomes

$$\begin{aligned}
\rho'''(\tau;\ \mu_1, \mu_2, \sigma) &= -C(\tau) \cdot \left(1 + \left(1 - \frac{\frac{2\tau\mu_1}{\sigma^2}}{1 + \frac{\mu_1(\mu_2+\tau)}{\sigma^2}}\right) \cdot \exp\left(\tfrac{\tau\mu_2}{\sigma^2}\right)\right) \\
&= -C(\tau) \cdot \left(1 + \left(1 - \tfrac{2\tau\mu_1}{\sigma^2+\mu_1(\mu_2+\tau)}\right) \cdot \exp\left(\tfrac{\tau\mu_2}{\sigma^2}\right)\right) \\
&= -C(\tau) \cdot \left(1 + h(\tau) \cdot \exp\left(\tfrac{\tau\mu_2}{\sigma^2}\right)\right)
\end{aligned} \tag{5.14}$$

where

$$h(\tau) = 1 - \frac{2\tau\mu_1}{\sigma^2 + \mu_1(\mu_2 + \tau)}.$$

Differentiating h

$$h'(\tau) = -\frac{2\mu_1(\sigma^2 + \mu_1(\mu_2 + \tau)) - 2\tau\mu_1^2}{(\sigma^2 + \mu_1(\mu_2 + \tau))^2} = -\frac{2\mu_1(\sigma^2 + \mu_1\mu_2) - 2\tau\mu_1^2}{(\sigma^2 + \mu_1(\mu_2 + \tau))^2} < 0$$

shows that h is strictly decreasing. Moreover, $h(0) = 1$ and $h(\tau)$ converges to -1 when $\tau \to \infty$. Consequently, h has a unique root which we call $\tilde{\tau}_1$. Furthermore, the function

$$h_2(\tau) = 1 + h(\tau)\exp(\tfrac{\tau\mu_2}{\sigma^2})$$

is positive on $[0, \tilde{\tau}_1[$ and diverges monotonically to $-\infty$ when $\tilde{\tau}_1 < \tau \to \infty$, so that h_2 has also a unique root $\tilde{\tau}_2$. This means that ρ''' is negative for $\tau \in [0, \tilde{\tau}_2[$ and positive for $\tau \in]\tilde{\tau}_2, \infty[$. It is clear from formula (5.11) that $\rho''(0) = 2$ and that $\rho''(\tau;\ \mu_1, \mu_2, \sigma)$ converges to 0 when $\tau \to \infty$. Together with the properties of ρ''' it follows that there is again a unique $\tilde{\tau}_3$ such that $\rho''(\tau;\ \mu_1, \mu_2, \sigma)$ is positive for $\tau \in [0, \tilde{\tau}_3[$ and negative for $\tau > \tilde{\tau}_3$. Finally, we conclude for our function of interest ρ' that $\rho'(\tau;\ \mu_1, \mu_2, \sigma)$ is strictly increasing on an interval $[0, \tilde{\tau}_4]$ and strictly decreasing and positive for $\tau > \tilde{\tau}_4$. This completes the second part of the proof. $\square$

The lemma motivates the following approach to calculate the minimal risk $R_{\min}$:

Algorithm 5.1.

(1) If $\mu_1 \cdot \mu_2 \leq 0$, then set $R_{\min} = \mu_1^2$ and stop.

(2) Otherwise, evaluate $q = \rho'(0;\ \mu_1, \mu_2, \sigma)$.

(3) If $q \geq 0$, then set $R_{\min} = \rho(0;\ \mu_1, \mu_2, \sigma)$ and stop.

(4) Set $\tau_0 = 0$.

(5) Calculate $\tau_1, \tau_2, \ldots, \tau_n$ as

$$\tau_j = \tau_{j-1} - \frac{\rho'(\tau_{j-1};\ \mu_1, \mu_2, \sigma)}{\rho''(\tau_{j-1};\ \mu_1, \mu_2, \sigma)} \tag{5.15}$$

for $j \in \{1, \ldots, n\}$, where n is the smallest integer such that

$$|\rho(\tau_n;\ \mu_1, \mu_2, \sigma) - \rho(\tau_{n-1};\ \mu_1, \mu_2, \sigma)| < \varepsilon$$

for a given precision ε.

(6) Set $R_{\min} = \rho(n;\ \mu_1, \mu_2, \sigma)$.

It is clear from Lemma 5.1 that the risk function does not have a minimum in the case where $\mu_1 \cdot \mu_2 \leq 0$. On the other hand, it converges very fast to μ_1^2, so that the first step of the algorithm makes sense.

The final step is now to ensure that the Newton iterations converge. This is done by the following theorem.

Theorem 5.1. *The sequence $(\tau_n)_1^\infty$, recursively defined by $\tau_0 = 0$ and equation (5.15), converges for every μ_1, μ_2, σ with $\mu_1 \cdot \mu_2 > 0$ and $\rho'(0;\ \mu_1, \mu_2, \sigma) < 0$ to the positive root of ρ', which is uniquely determined by Lemma 5.1.*

Proof. It is easy to see from the proof of Lemma 5.1 that the variables $\tilde{\tau}_2, \tilde{\tau}_3, \tilde{\tau}_4$ (defined in the proof) and the optimal threshold τ_{opt} satisfy $\tau_{opt} < \tilde{\tau}_4 < \tilde{\tau}_3 < \tilde{\tau}_2$. Therefore, ρ is an isotone concave function on $[0, \tau_{opt}[$. This means for $\tau_{j-1} \in [0, \tau_{opt}[$ that

$$\tau_j = \tau_{j-1} - \frac{\rho'(\tau_{j-1};\ \mu_1, \mu_2, \sigma)}{\rho''(\tau_{j-1};\ \mu_1, \mu_2, \sigma)} > \tau_{j-1}$$

and

$$\begin{aligned}
&\rho'(\tau_j;\ \mu_1, \mu_2, \sigma) \\
&\quad = \rho'(\tau_{j-1} - \frac{\rho'(\tau_{j-1};\ \mu_1, \mu_2, \sigma)}{\rho''(\tau_{j-1};\ \mu_1, \mu_2, \sigma)};\ \mu_1, \mu_2, \sigma) \\
&\quad = \rho'(\tau_{j-1};\ \mu_1, \mu_2, \sigma) + \int_{\tau_{j-1}}^{\tau_{j-1} - \frac{\rho'(\tau_{j-1};\ \mu_1, \mu_2, \sigma)}{\rho''(\tau_{j-1};\ \mu_1, \mu_2, \sigma)}} \rho''(t;\ \mu_1, \mu_2, \sigma) dt \\
&\quad < \rho'(\tau_{j-1};\ \mu_1, \mu_2, \sigma) + \rho''(\tau_{j-1};\ \mu_1, \mu_2, \sigma) \cdot \left(- \frac{\rho'(\tau_{j-1};\ \mu_1, \mu_2, \sigma)}{\rho''(\tau_{j-1};\ \mu_1, \mu_2, \sigma)} \right) \qquad (5.16) \\
&\quad = 0
\end{aligned}$$

and consequently, $\tau_j < \tau_{opt}$. Therefore, the (τ_n) form a non-negative, bounded and monotonically increasing sequence. It follows that the (τ_n) converge to a positive limit, and because of the fixed point iteration (5.15) this must be the positive root of ρ'. □

Before we proceed, we make a short remark, considering the special case where $\mu_1 = \mu_2$ which appears in usual wavelet analysis where 2^J time points are regularly spaced. In this case, there is exactly one positive root unless $\mu_1 \neq 0$.

Corollary 5.1. *If $\mu_1 = \mu_2 \neq 0$, then ρ' has exactly one positive root.*

Proof. The proof is obvious because

$$\rho'(0;\ \mu_1, \mu_1, \sigma) = -2\sigma \cdot (\phi(-\tfrac{\mu_2}{\sigma}) + \phi(\tfrac{\mu_2}{\sigma})) < 0.$$

□

We used Algorithm 5.1 to calculate the risk for Donoho and Johnstone's four test functions and optimal thresholds and got figures that were very similar to those in the previous section where we examined the VisuShrink threshold, with the only difference that the risk was much smaller. We discuss the results in Section 5.4.

5.4 Optimal noise-proportional thresholds

In the last section the thresholds $\tau_{j,k}$ could be chosen arbitrarily. However, in practice noise-proportional thresholds of the form

$$\tau_{j,k} = \lambda \cdot \sigma_{j,k}$$

seem to be more suitable. We saw already Donoho and Johnstone's popular *VisuShrink* where λ equals $\sqrt{2\log(n)}$. But one might ask if there are better choices for λ. Therefore, a natural extension to the results that we have obtained so far is to find for a given signal and given time points the value of λ which minimises the risk using these thresholds.

In other words, we are trying to minimise

$$r(\lambda) = \frac{1}{2^J} \sum_{j=j_0}^{J-1} \sum_{k=0}^{2^j} \rho(\lambda \cdot \sigma_{j,k};\ (\mathcal{W}f)_{j,k}, (\mathcal{W}R\tilde{f})_{j,k}, \sigma_{j,k}). \tag{5.17}$$

Unfortunately, the properties of the risk function and its derivatives are not as nice as in the last section. In particular, the Newton iterations do not always converge, and even if they do it is not clear if they converge to the global maximum, because it is possible that the risk function has local extrema as well. As an example, Figure (5.5) shows the function

$$\tilde{r}(\lambda) = \rho(20\lambda;\ 20, 20, 20) + \rho(\lambda;\ 5, 19, 1)$$

and its first derivative. The function $\tilde{r}$ can essentially be reagarded as the risk function r

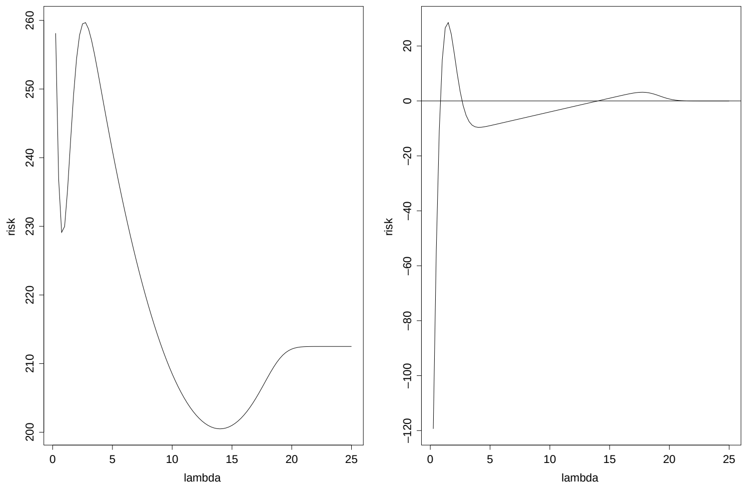

Figure 5.5: A simple example for a risk function with two local minima and its first derivative

for a special choice of f, σ, t and the wavelet basis and exhibits two local minima which are clearly visible.

One possibility to get an approximation for the minimal risk and the optimal constant λ is of course to calculate the values of R on a grid. Questions arise such as how to choose the width of the grid and the number of grid points. The following lemma helps to get the answer to the first question.

Lemma 5.2. *Let $\mu_1, \mu_2 \in \mathbb{R}$ and $\sigma > 0$. Set*

$$\tilde{\lambda} = \frac{|\mu_2|}{\sigma} + \sqrt{\max(1, -2\log(\frac{\sqrt{2\pi}\varepsilon}{8\sigma^2 + 4\sigma|\mu_1|}))}, \tag{5.18}$$

for a given precision ε. Then the first derivative of $\rho(\lambda\sigma;\ \mu_1, \mu_2, \sigma)$ (with respect to λ) has exponential decay and

$$\left|\frac{\partial}{\partial\lambda}\rho\left(\lambda\sigma;\ \mu_1, \mu_2, \sigma\right)\right| < \varepsilon$$

for all $\lambda > \tilde{\lambda}$.

The proof of the lemma requires a simple statement about ϕ and Φ.

Lemma 5.3. *If* $t \geq 0$ *then*

$$t \cdot \Phi(-t) < \phi(t).$$

Moreover, if $t \geq 1$ *then*

$$\Phi(-t) < \phi(t).$$

Proof. We can write $t \cdot \Phi(-t)$ as

$$\begin{aligned} t \cdot \Phi(-t) = \int_0^t \Phi(-s) - s\phi(-s)ds &= \int_0^t \Phi(-s)ds + \int_0^t -s\phi(s)ds \\ &= \phi(t) + \int_0^t \Phi(-s)ds - \phi(0). \end{aligned}$$

Therefore, we only have to show that the integral on the right side is less than $\phi(0)$:

$$\begin{aligned} \int_0^t \Phi(-s)ds = \int_0^t \int_{-\infty}^{-s} \phi(u)duds &= \int_0^t \int_{-\infty}^{\infty} \{u \leq -s\}\phi(u)duds \\ &= \int_{-\infty}^{\infty} \int_0^t \{u \leq -s\}ds\phi(u)du \\ &= \int_{-\infty}^{0} \phi(u)\min(t, -u)du \\ &< -\int_{-\infty}^{0} u\phi(u)du = \phi(0). \end{aligned} \tag{5.19}$$

The second part of the lemma is trivial. $\square$

Now we can prove Lemma (5.2).

Proof. Note first that similarly to (5.10) the first derivative becomes

$$\begin{aligned} \frac{\partial}{\partial\lambda}\rho(\lambda\sigma;\ \mu_1, \mu_2, \sigma) &= 2\sigma(\lambda\sigma + \mu_2 - \mu_1)\Phi(-\lambda - \tfrac{\mu_2}{\sigma}) + 2\sigma(\lambda\sigma + \mu_1 - \mu_2)\Phi(-\lambda + \tfrac{\mu_2}{\sigma}) \\ &\quad - 2\sigma^2\phi(-\lambda - \tfrac{\mu_2}{\sigma}) - 2\sigma^2\phi(-\lambda + \tfrac{\mu_2}{\sigma}) \\ &= 2\sigma^2(\lambda + \frac{\mu_2}{\sigma}\Phi(-\lambda - \tfrac{\mu_2}{\sigma}) + 2\sigma^2(\lambda - \tfrac{\mu_2}{\sigma}\Phi(-\lambda + \tfrac{\mu_2}{\sigma}) \\ &\quad - 2\sigma\mu_1(\Phi(-\lambda - \tfrac{\mu_2}{\sigma}) - \Phi(-\lambda + \tfrac{\mu_2}{\sigma})) \\ &\quad - 2\sigma^2(\phi(-\lambda - \tfrac{\mu_2}{\sigma}) + \phi(-\lambda + \tfrac{\mu_2}{\sigma})). \end{aligned} \tag{5.20}$$

If now $\lambda \geq \tilde{\lambda}$, then

$$-\lambda - \tfrac{\mu_2}{\sigma} \leq -1 \text{ and } -\lambda + \tfrac{\mu_2}{\sigma} \leq -1.$$

Therefore, applying Lemma (5.3) to the absolute value of equation (5.20) results in

$$\begin{aligned}\left|\frac{\partial}{\partial\lambda}\rho(\lambda\sigma;\ \mu_1,\mu_2,\sigma)\right| &\leq 2\sigma^2\phi(-\lambda - \tfrac{\mu_2}{\sigma}) + 2\sigma^2\phi(-\lambda + \tfrac{\mu_2}{\sigma}) \\ &\quad + 2\sigma|\mu_1|(\phi(-\lambda - \tfrac{\mu_2}{\sigma}) + \phi(-\lambda + \tfrac{\mu_2}{\sigma})) \\ &\quad + 2\sigma^2(\phi(-\lambda - \tfrac{\mu_2}{\sigma}) + \phi(-\lambda + \tfrac{\mu_2}{\sigma})) \\ &\leq (8\sigma^2 + 4\sigma|\mu_1|)\phi(\tilde{\lambda} - \frac{|\mu_2|}{\sigma}).\end{aligned} \tag{5.21}$$

If ε is very large such that

$$\varepsilon > \frac{8\sigma^2 + 4\sigma|\mu_1|}{\sqrt{2\pi}},$$

then (5.21) becomes

$$\left|\frac{\partial}{\partial\lambda}\rho(\lambda\sigma;\ \mu_1,\mu_2,\sigma)\right| \leq (8\sigma^2 + 4\sigma|\mu_1|)\frac{1}{\sqrt{2\pi}} < \varepsilon.$$

Otherwise, the terms in the next line are well defined:

$$\left|\frac{\partial}{\partial\lambda}\rho(\lambda\sigma;\ \mu_1,\mu_2,\sigma)\right| \leq (8\sigma^2 + 4\sigma|\mu_1|)\phi\left(\sqrt{-2\log\left(\frac{\sqrt{2\pi}\varepsilon}{8\sigma^2 + 4\sigma|\mu_1|}\right)}\right) = \varepsilon. \tag{5.22}$$

This completes the proof. $\square$

In particular, one consequence of the Lemma 5.3 is that the first derivative of the risk function has an absolute value less than ε at least for all

$$\lambda > \tilde{\lambda} = \max_{j,k} \tilde{\lambda}_{j,k} \tag{5.23}$$

where the $\tilde{\lambda}_{j,k}$ are computed with formula (5.18) with $\mu_1 = (\mathcal{W}f)_{j,k}$, $\mu_2 = (\mathcal{W}R\tilde{f})_{j,k}$ and $\sigma = \sigma_{j,k}$. Because of the exponential decay this gives us an upper bound for the width of the grid. One possibility to calculate the risk is given by the following algorithm which we used with the parameters K and N equal to 20 and 3.

Algorithm 5.2.

(1) Set $j = 0$, $l_0 = 0$ and calculate $r_0 = \tilde{\lambda}$ as defined in (5.23).

(2) Set $\lambda_i = \frac{i\cdot(r_j - l_j)}{K}$ for each $i \in \{0, \ldots, K\}$.

(3) Calculate the risk R_i for each λ_i

$$r_i = r(\lambda_i) = \frac{1}{2^J} \sum_{j=j_0}^{J-1} \sum_{k=0}^{2^j} \rho(\lambda_i \cdot \sigma_{j,k};\ (\mathcal{W}f)_{j,k}, (\mathcal{W}R\tilde{f})_{j,k}, \sigma_{j,k}).$$

(4) Find the three smallest values of $(R_i)_1^K$ as well as their corresponding λ and denote them $\lambda_{(1)}$, $\lambda_{(2)}$ and $\lambda_{(3)}$.

(5) Set

$$l_{j+1} = \max(0, \lambda_{(1)} - \frac{r_j - l_j}{K})$$

and

$$r_{j+1} = \lambda_{(3)} + \frac{r_j - l_j}{K}.$$

(6) Increment $j = j + 1$ and continue with step (2) if $j < N$.

(7) Choose $R_{opt} = \min_i R_i$ and the corresponding λ as the optimal threshold.

We also tried fixed point iterations similarly to the last section. Starting with $\lambda_0 = 0$ we created a sequence recursively by

$$\lambda_j = \lambda_{j-1} - \frac{R^{(1)}(\lambda_{j-1})}{R^{(2)}(\lambda_{j-1})}.$$

As we have pointed out above, it can happen that this sequence does not converge. Indeed, this happened sometimes for small values of n, especially with the Bumps data (and also sometimes with the Blocks data). However, the sequence always converged for the Doppler function, so we calculated the difference between the result obtained by Algorithm 5.2 and the fixed point iteration, and found that the relative error was always smaller than 1 per cent.

Other examples showed that the accuracy of the fixed point iteration is usually much higher than the grid calculations if the iterations converge, so that in much less computing time better results can be achieved. Therefore, we used for the remainder of this chapter a mixture of both methods: First, up to 20 fixed point iterations were applied, and only when there was no convergence, then the grid method was used.

5.5 Comparison between optimal thresholds and VisuShrink

In this section we want to compare the *VisuShrink* estimator with the optimal thresholds of the last two sections. Similarly to Section 5.1 we calculated the risk for Donoho and Johnstone's four test signals and n regularly spaced time points with n taking all values between 17 and 2048. Moreover we used four different noise levels ($\sigma = 0.05, 0.15, 0.25, 0.35$). The solid lines in figure 5.6 show the results for the Heavisine data and $\sigma = 0.35$. The results for the other data sets and noise levels do not look substantially different.

It is also interesting to compare the values of the thresholds. It turns out that (except for small n) the optimal thresholds are much smaller. We calculated the fractions

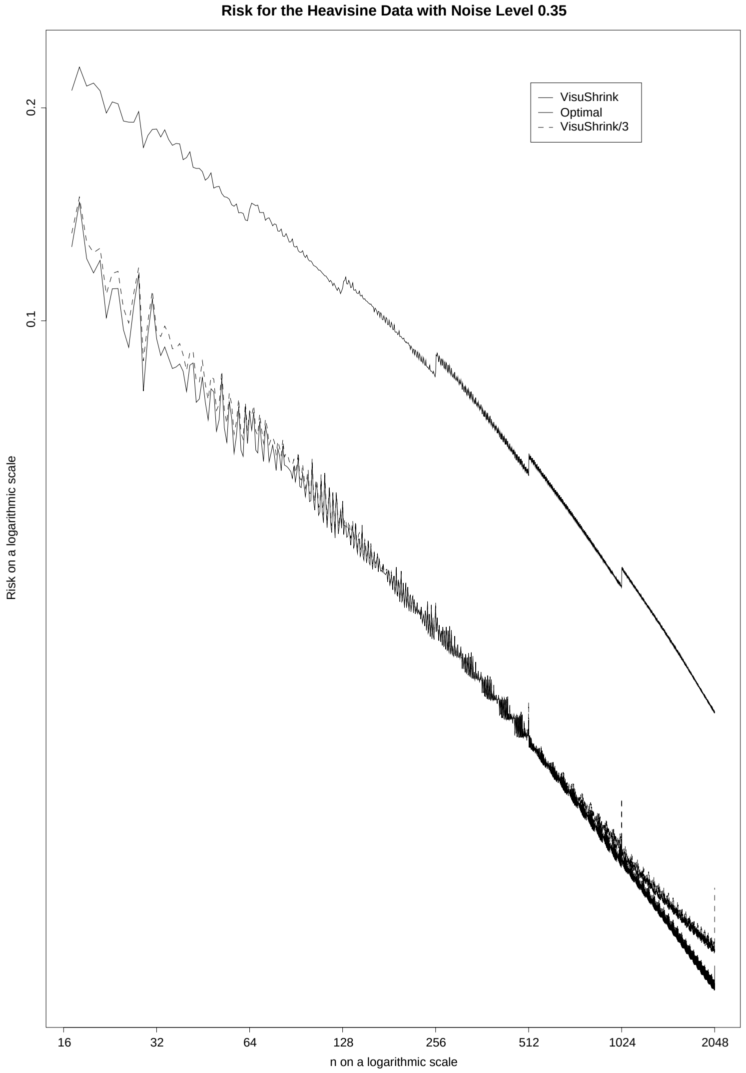

Figure 5.6: The risk for the Heavisine data and noise level 0.35 for three threshold choices on a regular grid of size n. The dashed and solid lines show the risk for the VisuShrink threshold choice and optimal chosen thresholds respectively. The dott ed curve represents a threshold choice where the VisuShrink threshold is divided by 3.

of the *VisuShrink* thresholds divided by the optimal thresholds for all 32512 test cases (four noise levels, four signals, 2032 values for n) and got a median of 3.443 and the mean 3.857. The 15%-quantile was 2.57 while the 85%-quantile was 4.7. If one is interested in obtaining a small MSE, then a simple idea is to use the *VisuShrink* thresholds divided by 3. The dotted line in figure 5.6 shows the resulting MSE which is surprisingly small and not much larger than the MSE for noise-proportional thresholds. It is not astonishing that a threshold choice which is smaller than VisuShrink reduces the expected mean squared error: In order to get a low MSE one has to trade off bias and variance, but the VisuShrink eliminates noise completely, thus gives too much bias. However, keep in mind that *VisuShrink* thresholds have the property of "noiseless", smoother reconstructions which is often more important than a low MSE. Moreover, the following remark gives some evidence that there might be better threshold choices than noise-proportional thresholds:

Let μ_1, μ_2 be real numbers such that $\mu_1 \cdot \mu_2$ is greater than 0, and let $\tau(\sigma)$ denote the optimal threshold for a given σ such that $\tau(\sigma)$ minimises $\rho(\tau;\ \mu_1, \mu_2, \sigma)$. Then the first derivative of $\tau(\sigma)$ does not converge to a finite number c for $\sigma \to \infty$.

Proof. It is clear from Lemma 5.1 that for each $\sigma > 0$

$$\rho'(\tau(\sigma);\ \mu_1, \mu_2, \sigma) = 0.$$

Differentiating both sides with respect to σ and rearranging yields

$$\tau'(\sigma) = -\frac{\frac{\partial}{\partial\sigma}\rho'(\tau(\sigma);\ \mu_1, \mu_2, \sigma)}{\frac{\partial}{\partial\tau}\rho'(\tau(\sigma);\ \mu_1, \mu_2, \sigma)}.$$

Further investigation of the numerator shows that

$$
\begin{aligned}
\frac{\partial}{\partial\sigma}\rho'(\tau(\sigma);\,\mu_1,\mu_2,\sigma)& \\
&= 2\cdot(\tau+\mu_2-\mu_1)\phi(\tfrac{-\mu_2-\tau}{\sigma})\cdot\tfrac{\mu_2+\tau}{\sigma^2} + 2\cdot(\tau+\mu_1-\mu_2)\phi(\tfrac{\mu_2-\tau}{\sigma})\cdot\tfrac{\tau-\mu_2}{\sigma^2} \\
&\quad - 2\cdot\phi(\tfrac{-\mu_2-\tau}{\sigma}) - 2\phi(\tfrac{\mu_2-\tau}{\sigma}) \\
&\quad - 2\sigma\cdot\frac{\mu_2+\tau}{\sigma}\cdot\phi(\tfrac{-\mu_2-\tau}{\sigma})\cdot\tfrac{\mu_2+\tau}{\sigma^2} - 2\sigma\cdot\tfrac{\tau-\mu_2}{\sigma}\cdot\phi(\tfrac{\mu_2-\tau}{\sigma})\cdot\tfrac{\tau-\mu_2}{\sigma^2} \qquad (5.24)\\
&= 2\left(-\frac{\mu_1(\mu_2+\tau)}{\sigma^2}-1\right)\phi(\tfrac{-\mu_2-\tau}{\sigma}) + 2\left(\tfrac{\mu_1(\tau-\mu_2)}{\sigma^2}-1\right)\phi(\tfrac{\mu_2-\tau}{\sigma}),
\end{aligned}
$$

so that τ satisfies the following non-linear differential equation:

$$
\tau'(\sigma) = \frac{\left(1+\frac{\mu_1(\mu_2+\tau)}{\sigma^2}\right)\phi(\frac{-\mu_2-\tau}{\sigma}) + \left(1-\frac{\mu_1(\tau-\mu_2)}{\sigma^2}\right)\phi(\frac{\mu_2-\tau}{\sigma})}{\Phi(\frac{-\mu_2-\tau}{\sigma}) + \Phi(\frac{\mu_2-\tau}{\sigma}) + \frac{\mu_1}{\sigma}\phi(\frac{-\mu_2-\tau}{\sigma}) - \frac{\mu_1}{\sigma}\phi(\frac{\mu_2-\tau}{\sigma})}. \qquad (5.25)
$$

Now suppose that there exists a finite number $c \in \mathbb{R}$ such that $\tau'(\sigma)$ converges to c when $\sigma \to \infty$. It is clear from the implicit definition of $\tau(\sigma)$ and formula (5.10) for the first derivative of ρ that $\tau(\sigma)$ itself converges to ∞ when $\sigma \to \infty$. Consequently, we conclude that

$$
c = \lim_{\sigma\to\infty}\tau'(\sigma) = \lim_{\sigma\to\infty}\frac{\tau(\sigma)}{\sigma}.
$$

By combining this with equation (5.25) and using Lemma 5.3 we get the following contradiction:

$$
c = \frac{2\phi(-c)}{2\Phi(-c)} > \frac{c\Phi(-c)}{2\Phi(-c)} = c.
$$

This finishes the proof. $\square$

Although we are not looking at infinitely large noise levels, the preceding statement raises the question how other simple threshold models than noise-proportional thresholds perform. Some answers are shown in Section 5.6 and 5.7.

We have a look at the case where the time structure is no longer regular. For a given n we calculated 20 different time structures as samples of size n from a Beta $B(2,2)$

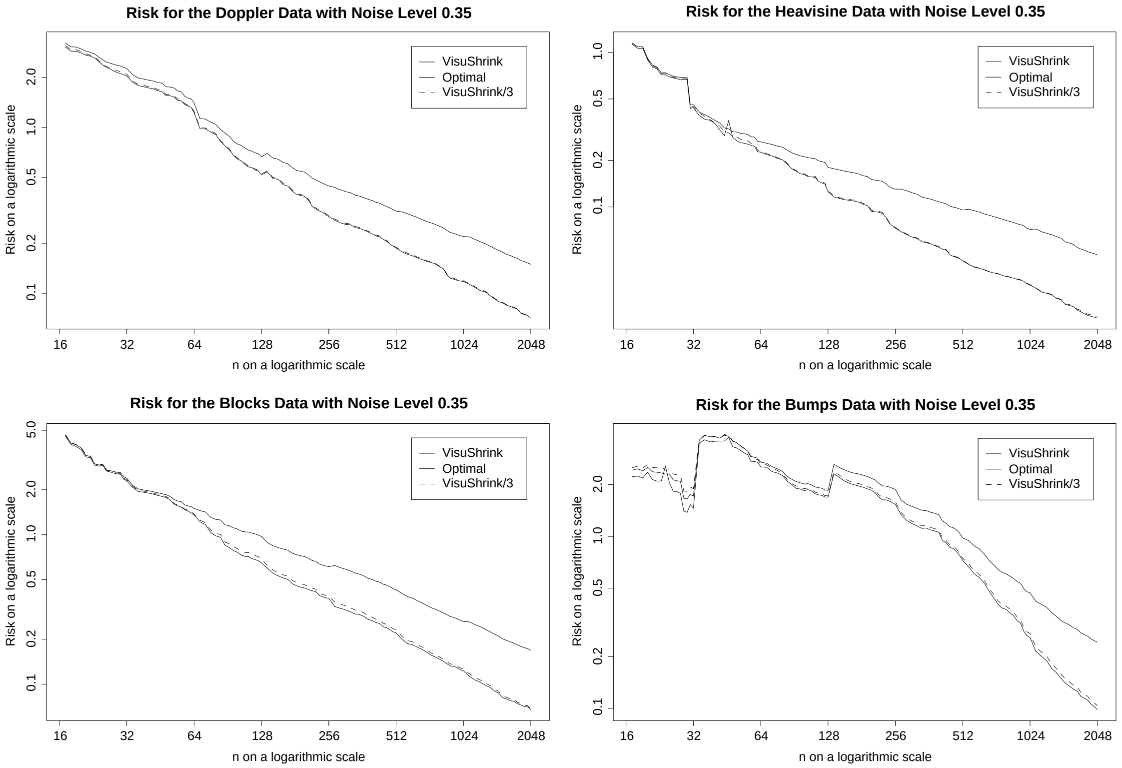

Figure 5.7: The expected MSE for DJ's test signals and noise level 0.35 for three threshold choices on irregular time structures of size n. Time points were independently drawn from a $Beta(2, 2)$ distribution and the exact risk calculated. Shown is the average over 20 replications. The dashed and solid lines represent the risk for VisuShrink and L^2-optimal thresholds of the form $\alpha\sigma_{j,k}$ while the dotted curve shows the risk for reduced Visushrink thresholds. The risk and n are both plotted on logarithmic scales.

distribution. Afterwards, for each time structure the exact MSE of a signal f and a fixed noise level σ was computed, and the average calculated. The procedure was carried out for optimal and VisuShrink thresholds as well as for the reduced VisuShrink thresholds. The results are plotted in Figure 5.7. The most obvious difference to figures 5.2 and 5.6 is that the curves are much smoother now. This is a consequence of the change in the time structures which is no longer fixed for given n. On the other hand, it is again astonishing how well the reduced VisuShrink thresholds perform in terms of expected mean square error. A difference to optimal thresholds is hardly visible.

Somewhat more explanation is needed for effects near some powers of two which occur in particular for the Bumps signal. The argument which was given earlier in Section 5.2 has to be ruled out since the jumps are not only visible for VisuShrink thresholds which change near powers of two, but also for the other threshold types. In fact, the reason is rather that the discrete version of the mean square error which we consider in this chapter depends on the grid length. The quality of the estimates does not appear to be much different for $n = 32$ and $n = 33$: Most of the peaks are either not resolved or massively flattened. However, the quantification via the mean square error is achieved by a comparison with two very different samples of the original signal: While the sample on a regular grid of size 32 exhibits the same features as the reconstructions, the peaks are much better resolved on a grid of size 64, leading to large residuals near those peaks.

A short simulation was carried out to check this result. Again the expected mean square error was determined for $n = 32$ and $n = 33$. Instead of calculating the mean square error for each time set with the formulae, simulated normal noise was added to the original signal and the VisuShrink procedure was performed. In the case $n = 32$ the vector

J	Doppler	Heavisine	Blocks	Bumps
4	1.25	0.147	2.11	5.24
5	0.67	0.079	0.94	3.57
6	0.42	0.054	0.69	2.56
7	0.30	0.049	0.44	1.85
8	0.23	0.046	0.41	1.39
9	0.24	0.047	0.37	1.34
10	0.22	0.047	0.35	1.34
11	0.24	0.048	0.37	1.34

Table 5.2: Simulated mean squared error for different sizes of the grid. The mean square error was calulated for each of Donoho and Johnstone'S four test signals and several values of J at 256 time points that were randomly sampled from a uniform distribution.

of the reconstructed signal was extended to size 64 by the usual grid transform. The risk was estimated in both cases using the squared residuals with respect to the values of the Bumps signal on a grid of size 64. While the usual mean square errors were approximately 0.21 and 0.39, the result of the simulation yielded 0.39 in both cases.

We close this section by examining the influence of the number of grid points on the expected mean squared error. Intuitively it is clear that the risk decreases initially with the size of the grid increasing at first, as the resolution of the output increases and more and more data points are used by the grid transform. However, once the distance between two grid points is small enougher than the minimum distance between each two data points, no further improvement is likely.

This supposition was checked briefly by a simulation study. The mean squared error of the reduced VisuShrink thresholds was simulated for uniform time samples of size 256 and several values of the output resolution J. However, like the previous example the mean squared error was always calculated on a grid of size 2048.

The results are displayed in Figure 5.2. Obviously an enlargement of J is especially effective for small values of J, but once the resolution is fine enough ($J \geq 8$) the mean squared error remains nearly constant. Note that the choice $J = 8$ is also suggested by the choice of J that we have mentioned earlier where J is chosen such that 2^J is the smallest power of two that is not smaller than n. This example is not only consistent with the considerations of the last paragraphs and shows that our ad-hoc method for the choice of the output resolution coincides with the "optimal" choice at least in this particular example.

5.6 Global thresholds vs. coefficient-dependent thresholds

In the last sections we examined coefficient-dependent thresholds. Now, we are going to calculate optimal global thresholds as well as their risk for the four test data sets, and surprisingly, we will see that often the risk will get smaller than in the situation where we use thresholds of the form $\alpha \cdot \sigma_{j,k}$ with optimally chosen α. The calculations were carried out in the same way as in Section 5.3: First fixed point iterations were tried. If after 30 iterations no convergence had taken place (which occurred rarely), we used the grid method. The results were very similar to the figures in the section where the thresholds were chosen proportional to the noise levels for each coefficient. But Figure 5.8 shows

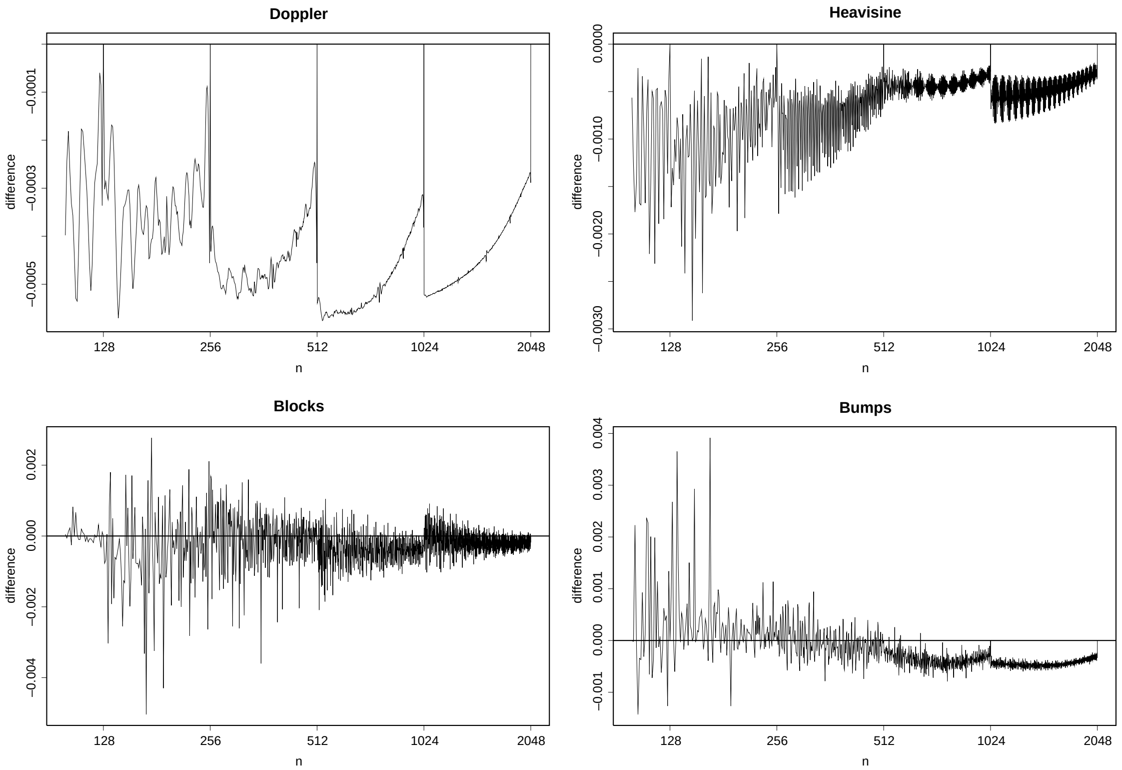

Figure 5.8: The differences between the risk for optimal global thresholds and optimal noise-proportional thresholds for the various test data

the differences between the risks for global and noise-proportional thresholds. [1] The first values for $n < 100$ are omitted, so that the trend is more clearly visible. Obviously, most differences are negative. To be more precise: 2032 different values for n were tried, and the risk for global thresholds was smaller than the risk for noise-proportional thresholds in 2009 (Doppler), 2005 (Heavisine), 1514 (Blocks) and 1812 (Bumps) cases.

Let us make some experiments to find out where this behaviour comes from. First of all, we use the Doppler data with $n = 1500$. The optimal constant for noise-proportional thresholds is $\alpha_{opt} = 1.3667$ and the optimal universal threshold is $\tau_{opt} = 0.2089$. Now we determine the risks $r^1_{j,k}$ and $r^2_{j,k}$ for each wavelet coefficient using the noise-dependent thresholds $\alpha_{opt} \cdot \sigma_{j,k}$ and the global threshold τ_{opt}. We separate the wavelet coefficients in two groups by defining the index sets

$$I_1 = \{(j,k) : r^1_{j,k} - r^2_{j,k} > 0.001\}$$

and

$$I_2 = \{(j,k) : r^1_{j,k} - r^2_{j,k} \leq 0.001\},$$

so that the wavelet coefficients where global thresholds perform clearly better than noise-dependent thresholds are labelled with an index from I_1.

We now take a look at the values of the wavelet coefficients that one gets when applying a wavelet transform to the Doppler function on a regular grid of size 2048. With the notation of the last sections these are the values of μ_1. Figure 5.9 shows first the coefficients with an index of I_1, then the ones with an index of I_2, and finally again those with an index of I_1 but this time the scale of the y-axis has changed.

[1] Note that the jumps to 0 for all values of n that are a power of two are due to the fact that all coefficients have the same variance in that case, so that there is no difference between optimal global and noise-proportional thresholds.

It can clearly be seen that the coefficients with an small absolute value coincide with the coefficients with an index in I_2. In other words, the small coefficients are set to zero with both global and noise-proportional thresholds. But the large coefficients can (at least in this example) be better modelled by global thresholds. We will see some reasons for this in the next section in a more general context.

The same experiment was carried out for the wavelet coefficients that are obtained by first transforming the Doppler on a regular grid of size 1500 to a regular grid of size 2048 and afterwards applying the wavelet transform. Not surprisingly, the result (see Figure 5.10) does not look very different.

5.7 Thresholds that are proportional to a power of the noise level

One might consider global and noise-proportional thresholds as special cases of a more general threshold model of the form

$$\tau_{j,k} = \alpha \cdot \sigma_{j,k}^{s}. \tag{5.26}$$

Here, the parameter s allows for more general dependencies of the threshold in terms of the variances, and yields global and noise-proportional thresholds for $s = 0$ and $s = 1$. Irregular spaced time points were created by drawing a sample of size 1024 from a uniform distribution on $[0, 1]$. We used a regular grid with 2048 grid points and calculated for different $s \in [-1, 2]$ the mean square error for the Doppler data, thresholds using model (5.26) and optimal α. The result is shown in Figure 5.11.

The global threshold achieves not only a better MSE than the noise-proportional threshold, using negative values for s yields even better results.

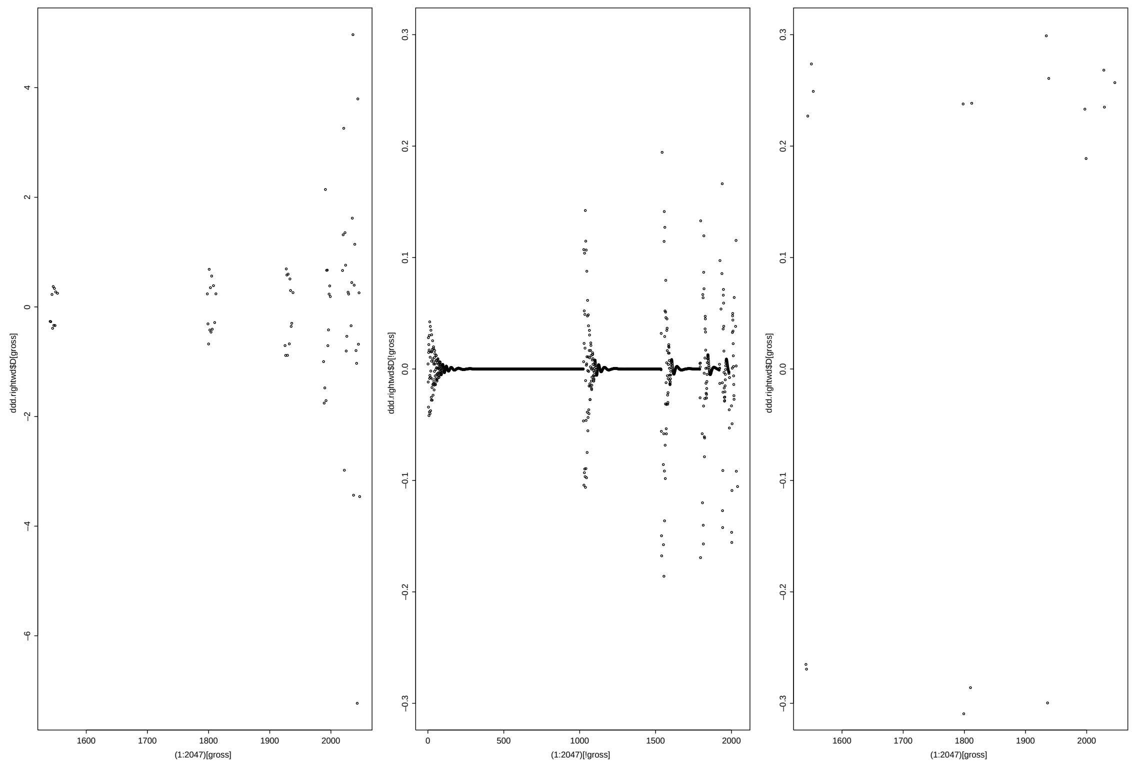

Figure 5.9: Wavelet coefficients separated by their risk behaviour. The DWT was applied to 2048 equally spaced points of the Doppler function. The left panel represents those wavelet coefficients where global thresholds perform clearly better than noise-proportional thresholds. The second panel shows the remaining coefficients, and the third figure shows again the coefficients from the first panel but on the same scale as the second figure.

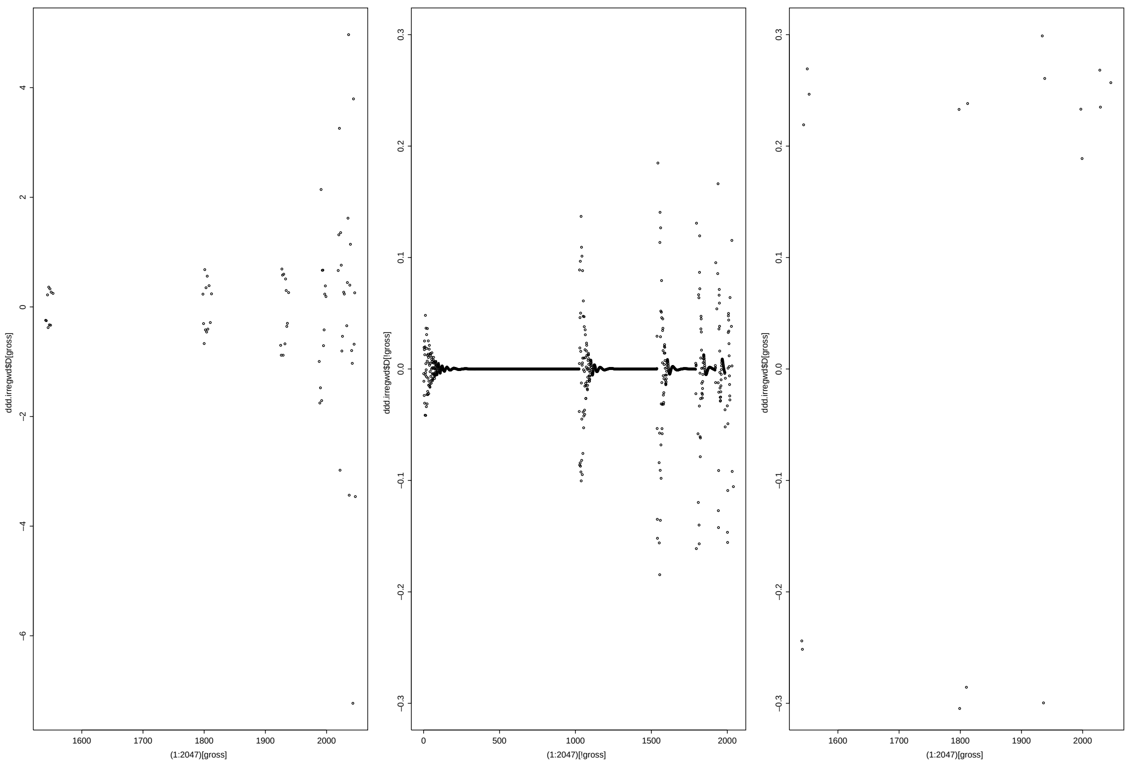

Figure 5.10: Wavelet coefficients separated by their risk behaviour. 1500 equally spaced points of the Doppler function were transformed on a grid of size 2048, and then the DWT was applied to the gridded data. The left panel represents those wavelet coefficients where global thresholds perform clearly better than noise-proportional thresholds. The second panel shows the remaining coefficients, and the third figure shows again the coefficients from the first panel but on the same scale as the second figure.

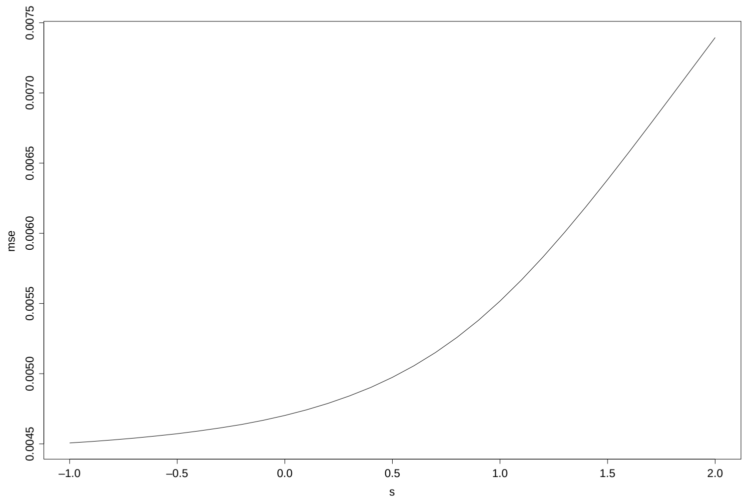

Figure 5.11: Mean Square Error for the Doppler data, irregular spaced time points and thresholds proportional to the s-th power of the local noise level with optimal factor

To understand this astonishing result we first consider another two examples. Suppose that the vectors of the true wavelet coefficients μ_1 and the biased wavelet coefficients μ_2 are given by $\mu_1 = \mu_2 = (1, 1, 1, 1, 1, 1, 1)$ and that the noise level looks like $\sigma = (1, 2, 3, 4, 5, 6, 7)$. Figure 5.12 shows the MSE functions for each coefficient. The function with the clearly visible local minimum is $\rho(\cdot;\ 1, 1, 1)$, and the other functions starting at the left-hand side are $\rho(\cdot;\ 1, 1, 2), \rho(\cdot;\ 1, 1, 3), \ldots, \rho(\cdot;\ 1, 1, 7)$.

What will an optimal α look like for different values of s? If s is equal to 0 then we are looking for a global threshold. Although a value $\alpha = 0.73$ would be fine for $\rho(\cdot;\ 1, 1, 1)$, since this function takes its minimum in this point, it certainly does not minimise the sum of the MSE functions. The optimal global threshold for $s = 0$ is ∞ and the mean square error converges to 1, because the MSE function for each coefficient converges to 1.

The MSE becomes smaller if s is greater than zero. For $s = 1$ (this corresponds to the noise-proportional thresholds that we have discussed earlier) we obtain an MSE of 0.99,

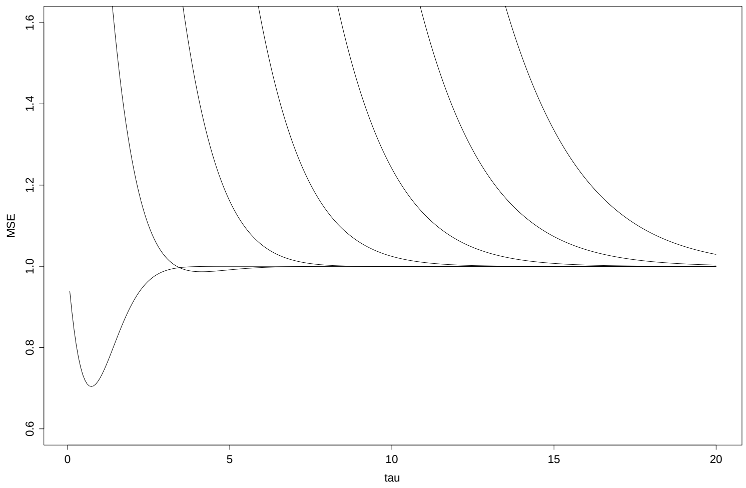

Figure 5.12: The MSE functions in a simple example for each coefficient. Each wavelet coefficient has mean 1 while the standard deviation takes from left to right each integer between 1 and 7.

and for $s = 2$ the MSE becomes 0.96. In the case $s = 2$, the optimal α is equal to 0.922. In particular, the threshold that is used for the first coefficient is 0.922. Thus the larger value of s leads to a threshold choice where the MSE function for the first coefficient nearly attains its minimum, while the other functions are approximately 1.0.

Finally, if s converges to ∞, one can check that with an optimal threshold choice the threshold for the first coefficient converges to the value that minimises $\rho(\cdot;\ 1, 1, 1)$ while the other thresholds converge to ∞.

We consider now a second example which is only slightly different from the first one. Again we choose

$$\sigma = (1, 2, 3, 4, 5, 6, 7),$$

but

$$\mu_1 = \mu_2 = (1, 1, 1, 1, 1, 1, 100).$$

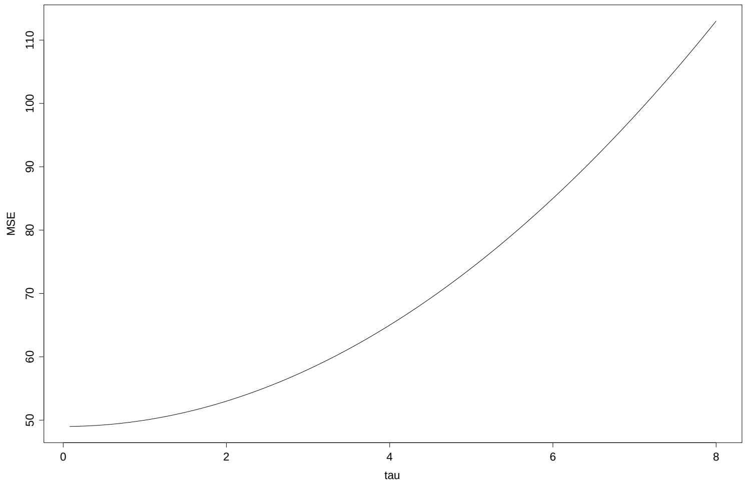

Figure 5.13: The MSE function for a wavelet coefficient with mean 100 and standard deviation 7.

We will see later that this artificial example is very similar to the wavelet coefficients that can be obtained by applying the DWT to a grid transform of noisy samples. The MSE functions for the first up to the sixth coefficient are of course the same as above. The function for the seventh coefficient $\rho(\cdot;\ 100, 100, 7)$ is shown in Figure 5.13. It is obviously totally different from the other MSE functions, since it is strictly monotonically increasing.

What should optimal thresholds look like? The MSE function of the seventh wavelet coefficient is minimal for $\tau = 0$. On the other hand, all the other coefficients require large thresholds (except possibly the first coefficient, but we cannot expect that our simple threshold model (5.26) will take this into account).

Thus, we may expect that negative values of s give better results than positive values, because the noise level of the seventh coefficient is higher than the noise of the other coefficients, and this assumption is confirmed by table 5.3 where the minimal expected mean square error is shown for several values of s.

s	-3	-2	-1	0	1	2	3
MSE	10.38	10.91	11.65	16.29	20	20	20

Table 5.3: The minimal expected mean square error for an artificial data set and different threshold models

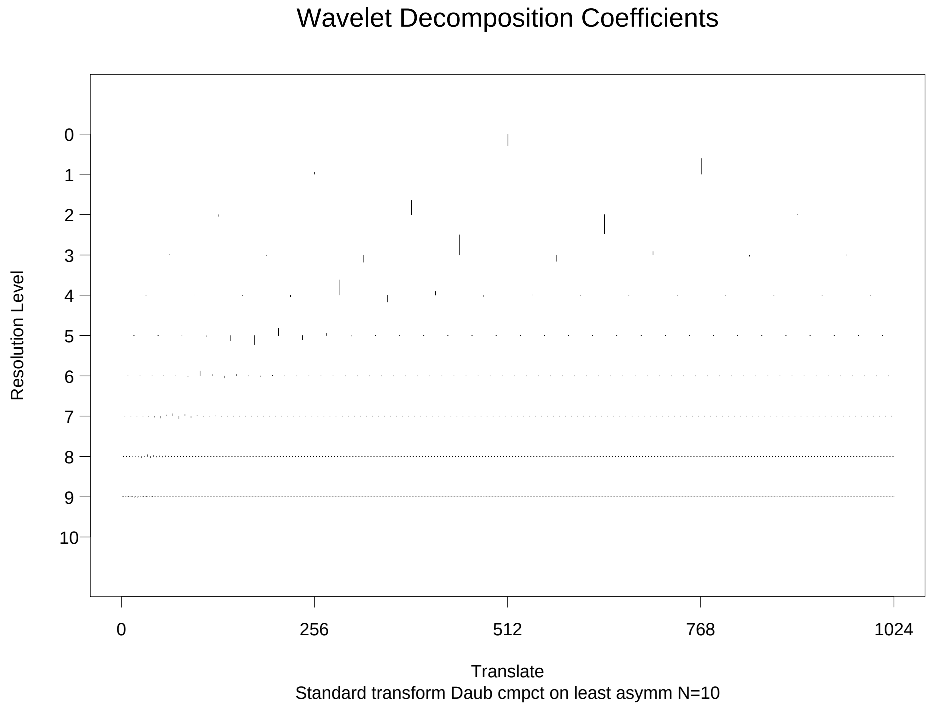

Figure 5.14: Wavelet Decomposition for 2048 regular spaced points of the Doppler data

The values for $s \geq 1$ are identical, because the optimal α is then equal to 0.

It was already mentioned that the preceding example is very similar to the wavelet coefficients that are obtained by the grid transform. One well known property of the wavelet transform is that smooth functions have economical wavelet expansions. Only a few number of wavelet coefficients are large, and these coefficients are usually found in lower levels.

On the other hand, applying the grid transform and the DWT to data usually yields wavelet representations where the average noise level for coefficients in lower levels is

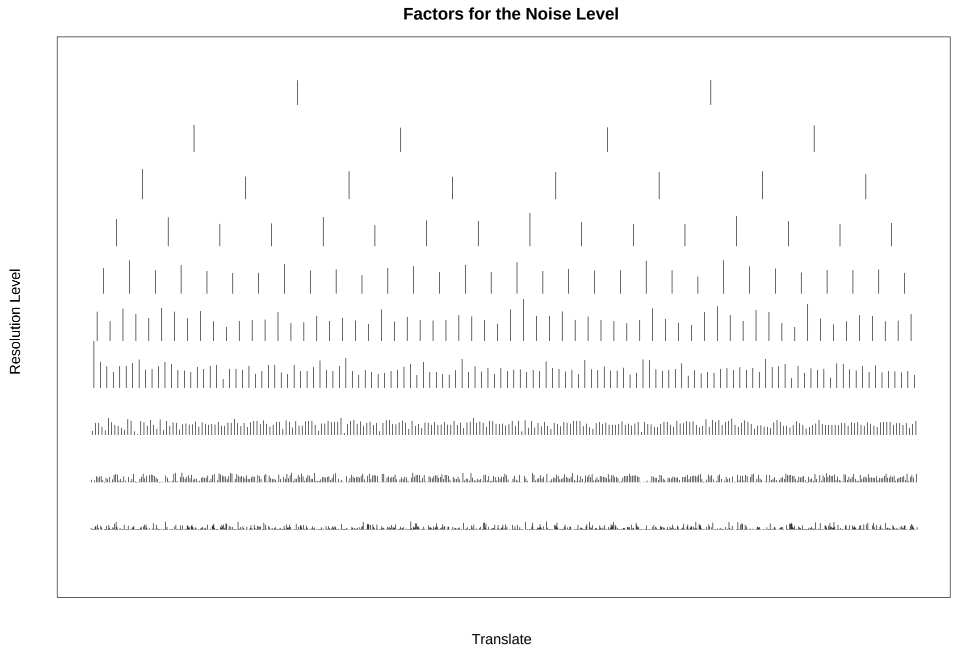

Figure 5.15: Noise level of each wavelet coefficient in a wavelet decomposition of a randomly spaced sample of the Doppler function. The Doppler function was evaluated at 1024 randomly drawn points of a uniform distribution on $[0, 1]$, and these data were transformed on a grid. Afterwards the variance of each wavelet coefficient was determined with the methods of Chapter 3.

slightly higher than for the other coefficients. For example, consider the data set we have regarded at the beginning of this section. Figures 5.14 and 5.15 show the wavelet coefficients that are obtained by applying the DWT to 2048 regular spaced points of the Doppler data and the noise levels we got for the sample of a uniform distribution on $[0, 1]$ which was described above, and another example was already given in Figures 4.2 and 4.3.

5.8 How good is hard thresholding?

Until now we have only used soft thresholding. In this section we investigate whether we can achieve better results using hard thresholding. First of all, the formulas for the mean square error function and its first derivatives are, for $s \in \mathbb{R}$:

$$\begin{aligned}\rho_H(\alpha \cdot \sigma^s;\ \mu_1, \mu_2, \sigma) = \mu_1^2 + \left(\sigma^2 + (\mu_2 - \mu_1)^2 - \mu_1^2\right) \cdot (\Phi(\tfrac{-\mu_2 - \alpha\cdot\sigma^s}{\sigma}) + \Phi(\tfrac{\mu_2 - \alpha\cdot\sigma^s}{\sigma}))\\ + \sigma \cdot (2\mu_1 - \mu_2 + \tau) \cdot \phi(\tfrac{-\mu_2 - \alpha\cdot\sigma^s}{\sigma})\\ - \sigma \cdot (2\mu_1 - \mu_2 - \tau) \cdot \phi(\tfrac{\mu_2 - \alpha\cdot\sigma^s}{\sigma})\end{aligned} \tag{5.27}$$

and

$$\begin{aligned}\frac{\partial}{\partial\alpha}\rho_H(\alpha \cdot \sigma^s;\ \mu_1, \mu_2, \sigma) = \phi(\tfrac{-\mu_2 - \alpha\cdot\sigma^s}{\sigma}) \cdot \sigma^{2s-1} \cdot (-2\mu_1\alpha - \alpha^2\sigma^s)\\ + \phi(\tfrac{\mu_2 - \alpha\cdot\sigma^s}{\sigma}) \cdot \sigma^{2s-1} \cdot (2\mu_1\alpha - \alpha^2\sigma^s)\end{aligned} \tag{5.28}$$

We used these formulas to calculate the optimal mean square error for the four test signals and a sample of size 2000 from a uniform $\mathcal{U}(0, 1)$ distribution. We tried three different noise levels, six values for s, hard and soft thresholding, and even two different cut-off levels. The results are shown in Table 5.4.

We can draw a lot of conclusions from the table. First of all, the MSE for hard thresholding is smaller than the MSE for soft thresholding in 85 of 144 cases. On the

			Doppler		Heavisine		Blocks		Bumps	
Cut-Off	σ	s	Hard	Soft	Hard	Soft	Hard	Soft	Hard	Soft
3	0.05	0.0	0.00110	0.00106	0.00153	0.00167	0.04907	0.04968	0.01538	0.01603
3	0.05	0.2	0.00109	0.00107	0.00154	0.00168	0.04894	0.04965	0.01539	0.01604
3	0.05	0.4	0.00108	0.00107	0.00154	0.00167	0.04895	0.04964	0.01540	0.01606
3	0.05	0.6	0.00107	0.00109	0.00152	0.00168	0.04886	0.04964	0.01537	0.01608
3	0.05	0.8	0.00107	0.00111	0.00153	0.00170	0.04892	0.04964	0.01537	0.01611
3	0.05	1.0	0.00109	0.00113	0.00154	0.00173	0.04892	0.04965	0.01537	0.01614
3	0.15	0.0	0.00528	0.00453	0.00565	0.00563	0.05579	0.05648	0.02238	0.02408
3	0.15	0.2	0.00532	0.00460	0.00553	0.00564	0.05481	0.05629	0.02235	0.02415
3	0.15	0.4	0.00539	0.00470	0.00547	0.00569	0.05384	0.05620	0.02199	0.02428
3	0.15	0.6	0.00550	0.00483	0.00536	0.00580	0.05338	0.05623	0.02175	0.02447
3	0.15	0.8	0.00570	0.00501	0.00533	0.00596	0.05310	0.05637	0.02181	0.02472
3	0.15	1.0	0.00602	0.00524	0.00535	0.00619	0.05305	0.05659	0.02209	0.02502
3	0.50	0.0	0.02481	0.02123	0.03563	0.03475	0.11396	0.11444	0.08436	0.07870
3	0.50	0.2	0.02476	0.02145	0.03558	0.03524	0.11293	0.11501	0.08400	0.07956
3	0.50	0.4	0.02481	0.02178	0.03571	0.03594	0.11208	0.11591	0.08420	0.08079
3	0.50	0.6	0.02502	0.02226	0.03613	0.03694	0.11167	0.11730	0.08522	0.08243
3	0.50	0.8	0.02550	0.02297	0.03708	0.03836	0.11174	0.11937	0.08746	0.08460
3	0.50	1.0	0.02659	0.02402	0.03903	0.04034	0.11317	0.12218	0.09140	0.08737
5	0.05	0.0	0.00109	0.00104	0.00153	0.00163	0.04908	0.04962	0.01537	0.01598
5	0.05	0.2	0.00108	0.00104	0.00154	0.00162	0.04895	0.04958	0.01538	0.01599
5	0.05	0.4	0.00107	0.00105	0.00154	0.00161	0.04895	0.04955	0.01540	0.01600
5	0.05	0.6	0.00106	0.00106	0.00152	0.00161	0.04887	0.04954	0.01536	0.01602
5	0.05	0.8	0.00106	0.00107	0.00153	0.00162	0.04893	0.04954	0.01536	0.01604
5	0.05	1.0	0.00107	0.00109	0.00155	0.00164	0.04893	0.04955	0.01536	0.01607
5	0.15	0.0	0.00506	0.00431	0.00549	0.00495	0.05585	0.05616	0.02232	0.02373
5	0.15	0.2	0.00508	0.00435	0.00533	0.00489	0.05489	0.05591	0.02289	0.02374
5	0.15	0.4	0.00513	0.00441	0.00525	0.00485	0.05394	0.05574	0.02192	0.02382
5	0.15	0.6	0.00520	0.00449	0.00512	0.00485	0.05347	0.05570	0.02166	0.02396
5	0.15	0.8	0.00534	0.00461	0.00506	0.00489	0.05319	0.05577	0.02170	0.02416
5	0.15	1.0	0.00558	0.00478	0.00505	0.00497	0.05313	0.05593	0.02196	0.02441
5	0.50	0.0	0.02273	0.02102	0.02538	0.02454	0.11327	0.11061	0.08270	0.07000
5	0.50	0.2	0.02273	0.02103	0.02497	0.02439	0.11216	0.11073	0.08192	0.07600
5	0.50	0.4	0.02273	0.02108	0.02461	0.02428	0.11114	0.11110	0.08158	0.07669
5	0.50	0.6	0.02273	0.02118	0.02432	0.02426	0.11051	0.11185	0.08190	0.07773
5	0.50	0.8	0.02273	0.02135	0.02415	0.02437	0.11025	0.11319	0.08320	0.07920
5	0.50	1.0	0.02273	0.02164	0.02424	0.02468	0.11113	0.11526	0.08592	0.08124

Table 5.4: Comparison of the optimal Mean Square Error for hard and soft thresholding, Donoho and Johnstone's four test signals, three different noise levels, two cut-off levels and six values for s.

	0.0	0.2	0.4	0.6	0.8	1.0
Soft	16	2	5	6	2	0
Hard	3	4	2	12	8	4

Table 5.5: Comparison of the performance of several threshold models

other hand, hard thresholding is in particular effective for small σ (better in 41 of 48 cases), while it does not perform as good for medium (29 of 48) or large σ (15 of 48).

We can also have a look at how different values of s have an influence on the performance of hard and soft thresholding. Table 5.5 shows how often a value of s was best for a given data set, noise level and cut-off level. If different values for s attained the same MSE, they were all counted.

The first row of the table shows what we found out in the last sections. Global thresholds are better than hard thresholds, and if we had tried out negative values for s they probably would have been even better. However, using hard thresholding, there is something like an "optimal" value for s near 0.6.

5.9 Final comparison

The aim of this final section is to compare five techniques for estimating a function on a grid in terms of their MSE. Among them are the new version of SureShrink which was already introduced in Section 4.4 and two new approaches that were not analysed above, but were mentioned in the introductory chapter and in Section 3.1.

This is the idea: We suppose that the number of data points is a power of two. Then, instead of applying the grid transform, we ignore the time structure and perform immediately the DWT on the given data (y_i), followed by usual thresholding. This gives us an estimate for the function on the irregularly spaced time points. Afterwards we do the grid transform to get an estimate on a grid.

This algorithm clearly has the advantage that it does not require the calculation of the variances and can easily be generalised to arbitrary numbers of data points by imposing boundary conditions. On the other hand, we will see that these methods work not very well as far as their MSE is concerned. [2] This is due to the fact that wavelet transforms of a sample which is taken from irregularly spaced time points are usually not as economical as in the regular setting, because the property of vanishing moments does not apply anymore.

Table 5.6 shows the result of a simulation study that was carried out in R to compare five methods:

IrregSure Transform the data to a grid, apply the DWT to the grid data, calculate the global *SureShrink* threshold, do soft thresholding, and apply the inverse DWT.

IrregVis Same as IRREGSURE, but use noise-proportional thresholds $\alpha \cdot \sigma_{j,k}$, where the factor α equals to $\sqrt{2\log(n)}$.

IrregVis3 Like IRREGVIS, but divide the factor α by 3, i.e. use thresholds $\tau_{j,k} = \sqrt{2\log(n)} \cdot \sigma_{j,k}/3$.

RankSure Apply Donoho and Johnstone's *SureShrink* to the data and then perform the grid transform to get an estimate on a grid.

RankVis Perform Donoho and Johnstone's *VisuShrink* followed by the grid transform.

[2] There are other disadvantages as well. Lenarduzzi shows (see [75]) that the resulting curves do not look "graphically pleasant".

Time	Signal	IrregSure	IrregVis	IrregVis3	RankSure	RankVis
B(1,1)	Doppler	0.03169	0.11872	0.03571	0.03583	0.10346
B(1,1)	Heavisine	0.01357	0.03874	0.01611	0.01442	0.03276
B(1,1)	Blocks	0.06136	0.15882	0.06401	0.06058	0.13696
B(1,1)	Bumps	0.07620	0.23126	0.08427	0.08542	0.23103
B(2,2)	Doppler	0.07013	0.15594	0.06869	0.07283	0.15354
B(2,2)	Heavisine	0.01769	0.04926	0.01876	0.02824	0.07676
B(2,2)	Blocks	0.06518	0.17633	0.06700	0.06004	0.13811
B(2,2)	Bumps	0.09446	0.25400	0.10060	0.09780	0.23939
B(3,3)	Doppler	0.17641	0.23014	0.15911	0.18473	0.29209
B(3,3)	Heavisine	0.05966	0.08860	0.05417	0.09928	0.24964
B(3,3)	Blocks	0.09131	0.24600	0.09891	0.08598	0.17727
B(3,3)	Bumps	0.17272	0.37531	0.18685	0.18254	0.34727
B(4,4)	Doppler	0.34255	0.33497	0.30162	0.36192	0.54619
B(4,4)	Heavisine	0.15240	0.13265	0.12577	0.19245	0.42891
B(4,4)	Blocks	0.14067	0.35623	0.15457	0.13686	0.25261
B(4,4)	Bumps	0.37138	0.57469	0.38522	0.38152	0.55463

Table 5.6: Comparison of the average Mean Squared Error for five different thresholding techniques. The methods IRREGSURE, IRREGVIS and IRREGVIS3 are based on the grid transform and coefficient-dependent thresholds as introduced in Section 3 and Section 4. RANKSURE and RANKVIS do not take the irregular time structure into account at first and perform ordinary thresholding followed by a grid transform. The time points were chosen randomly from four different Beta distributions. For each thresholding technique, signal, and model for the time structure, the average of the MSE over 50 replications was calculated.

For each of the four test functions and for each method we calculated the average mean square error over 50 replications where the noise was white noise with $\sigma = 0.35$ and the time points samples of size 2048 from four Beta distributions B(1,1) (identical with a uniform distribution), B(2,2), B(3,3) and B(4,4).

Obviously, the methods based on *VisuShrink* do not perform very well. But that is not very surprising, because the conservative threshold choice does not attempt to obtain a low MSE. As described earlier, the main idea is to produce "noise-free" reconstructions.

Another interesting observation that can be made is that RankSure always attains the smallest MSE for the Blocks data. As we pointed out above, wavelet decompositions of smooth functions that are sampled on an irregular grid are usually not as economical as in the equally spaced setting. However, for the piecewise constant Blocks signal samples on regular grids look essentially the same compared with samples on an irregular grid. Therefore it is not astonishing that RankSure performs very well in this situation.

On the other hand IrregSure always performs better than RankSure for the other three test signals, in particular for the Heavisine data which is the smoothest signal and has the most economical wavelet representation, hence is the most unfavourable for the RankSure method. The simple IrregVis3 method exhibits again a surprisingly small MSE which is in 5 cases even smaller than the MSE for IrregSure.

5.10 Summary

This chapter was concerned with an analysis of the performance of our new wavelet methods measured by the mean squared error. We summarize the most important results:

- An exact formula for the risk was derived and used to avoid, as far as possible, the need for simulation.

- The grid transform analysed in this thesis is responsible for a considerable amount of bias. It might be interesting to see whether other transformations perform similarly or if it is possible to reduce the bias.

- The performance of optimal thresholds which minimise the risk for a known signal was assessed. It turned out that reduced VisuShrink thresholds $\sqrt{2\log(n)}/3$ come very close to the optimal thresholds.

- The influence of the output resolution J on the risk was examined. In a simulation study with data sets of size 256 the choice $J = 8$ appeared to be optimal in the sense that smaller values for J would considerably enlarge the risk while larger values would only increase the computing time, but would not decrease the mean squared error.

- Global thresholds instead of noise-proportional thresholds were considered. Apparently these thresholds perform slightly better in many cases due to some properties of the variances of wavelet coefficients which have to be investigated further in future work.

- Our method was compared with another wavelet method for unequally spaced data where at first usual thresholding is applied to the given data regardless of the time structure and then the reconstructed signal is transformed to a grid. In this example our new methods performed better except for piecewise constant signals where the other method has advantages.

Chapter 6

Robust Wavelet Thresholding

He ate and drank the precious words,
His spirit grew robust;
He knew no more that he was poor,
Nor that his frame was dust.

He danced along the dingy days,
And this bequest of wings
Was but a book. What liberty
A loosened spirit brings!

—Emily Dickinson, *Poems, Ch. I. Life*

In the previous sections we have developed tools for smoothing unequally spaced data using wavelet methods. We always assumed that the noise was normally distributed and obtained good results in applying the new algorithms to two data sets. However, an important problem arises when a normal distribution is not a suitable model for the noise, in particular when its distribution has heavy tails. A similar problem arises when all observations are disturbed by Gaussian noise, but some of them are for some reason contaminated by outliers. In both cases classical wavelet thresholding does not represent a suitable method for dealing with those data sets, as we will see at the beginning of this chapter.

Median filtering (often called *running medians*) represents a very popular robust smoothing method and offers several advantages while having the serious disadvantages that median-filtered data are usually very rough. Nevertheless, median filters will be one of our main ingredients in Section 6.4, 6.5 and 6.7 where we propose robust wavelet regression estimators. Two approaches try to recognise outliers, and apply wavelet thresholding on the remaining data points. While running medians are used for the outlier test, the algorithms for thresholding unequally spaced data are also of great importance, because the "cleaned" data set will rarely be equally-spaced, even if the original data is. The other approach tries to smooth the output of the median filter: Again we also make use of the fast algorithm for calculating the variances when we determine suitable thresholds.

Therefore we start this chapter by describing the implications of outliers on wavelet thresholding in Section 6.1. Then we review some material on robust wavelet regression and median filters in Section 6.2 and 6.3. In Section 6.4 and 6.5 we introduce two methods where possible outliers are removed from the data. Several parameters have to be chosen in both approaches, and Section 6.6 deals with this problem, mainly based on some statements on order statistics. Another robust wavelet regression technique is introduced in Section 6.7, and finally we discuss several examples in Section 6.8.

6.1 Wavelet Thresholding and robustness

An important problem arises in non-parametric regression when some of the observations are contaminated by outliers. In particular wavelet shrinkage seems to be very sensitive to outliers because large observations in the data typically lead to large coefficients in a wavelet decomposition, which will not be removed by usual thresholding approaches.

We illustrate this with a simple example, which is similar to an example that was provided by Donoho (see [34]). A scaled sine curve was evaluated at 2048 equally spaced points and the values were corrupted by Cauchy noise $\mathfrak{C}(0, 0.15)$ (see figure 6.1). The absolute values of 80 percent of the noise sample were smaller than 0.43, and only 5 percent of the data had absolute values larger 1.5. Nevertheless, it is hard to recognise the sine curve in the figure at all.

The figure also shows two attempts to reconstruct the underlying sine curve. First, VisuShrink was applied to the data, leaving the outliers nearly intact. Then also all remaining non-zero coefficients in the finest four levels were set to zero as well. Although the underlying sine curve is visible this time, it looks still very unpleasant. The aim of this chapter is to develop algorithms that produce good results even if the the noise is not normally distributed or if the data are contaminated by outliers. We will place an emphasis on situations where several outliers follow each other and, using the terminology of Bruce et al. (see [15]), we call those collections of data points *outlier patches.* The reason is that in many applications unusual large observations do not occur independently—they appear very often as a consequence from a temporary disturbance of the measuring device. For example, we will analyse in Section 6.8 a data set which consists of radiation measurements of a weather balloon where sometimes a rope cuts the measuring device from the direct sun light, causing extremely large outlier patches with up to 150 consecutive outliers. We will also revisit the sine curve in the same section and see that it is possible to get back the original curve using robust wavelet regression methods.

In the context of robust statistics the concept of *breakdown point* is well-known as a measure for quantifying the sensitivity of an estimator to outliers in a dataset. Loosely

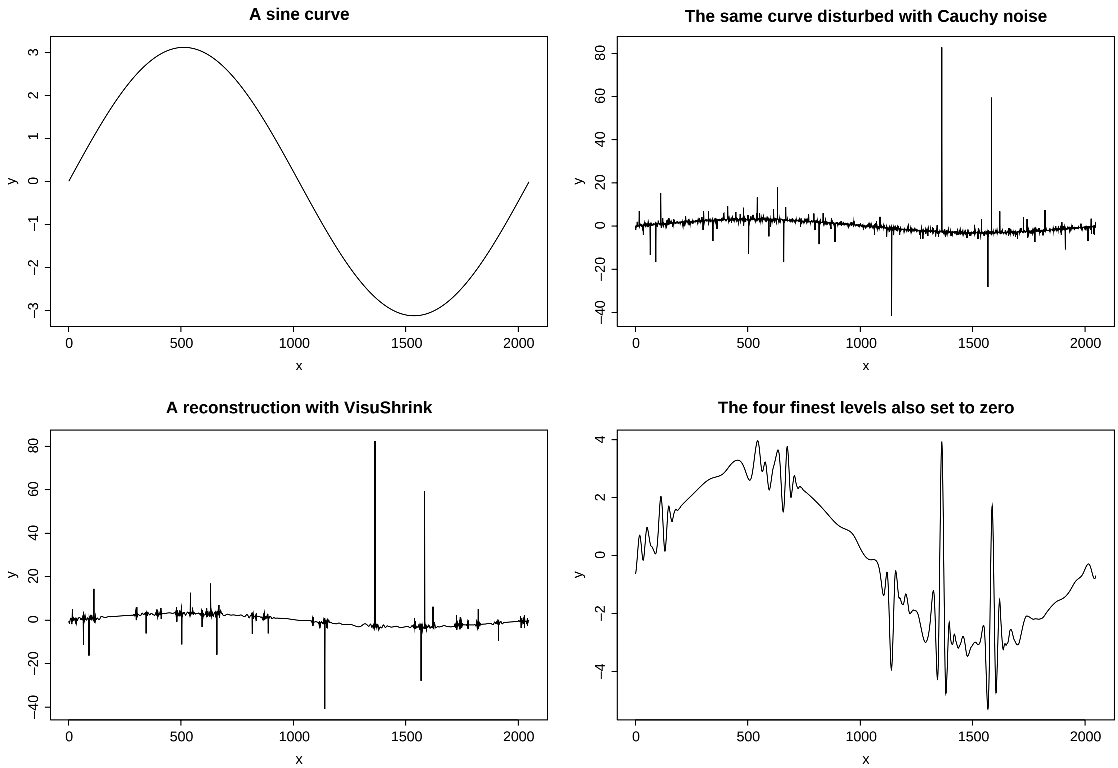

Figure 6.1: A sine curve is disturbed by Cauchy noise along with two attempts for reconstructing it

speaking, it measures how much bad data an estimator can withstand before "breaking down" completely. Although Hodges used already some of the ideas in an early paper (see [60]), Hampel is generally credited with having introduced this concept (see [52] and [53]) and several estimators were analysed using his definition, for example, by Andrews et al. (see [5]) and Huber (see [62]).

However, while Hampel's breakdown point deals with the asymptotic behaviour of estimators, Donoho and Huber found (see [37]) that it was possible to define the breakdown point without recourse to a probability model. The measure they defined is called the *finite sample breakdown point* because the breakdown point is computed for estimators with a finite sample size.

Definition 6.1. *Let Θ be a subset of a metric space with metric d. The finite sample breakdown point of an estimator $\hat{\theta}_n$ at $z = (z_1, \ldots, z_n)$ is the fraction*

$$\varepsilon(\hat{\theta}_n, z) = \inf_{1 \leq m \leq n} \left\{ \frac{m}{n} : b(m, n;\ z, \hat{\theta}_n) = \infty \right\},$$

where

$$b(m, n;\ z, \hat{\theta}_n) = \sup_{\tilde{z} \in D_m^n(z)} d(\hat{\theta}_n(\tilde{z}), \hat{\theta}_n(z))$$

and $D_m^n(z)$ is the subset of Θ which is created by choosing arbitrarily m observations from the dataset z and replacing each of the data points with an arbitrarily chosen element of Θ.

Recently, other definitions of the finite sample breakdown point were suggested by Stromberg and Ruppert (see [107]) and Sakata and White (see [96]) because the Donoho-Huber breakdown point is always 1 when the parameter space is bounded. However, we will not pursue these ideas any further and use Definition 6.1 throughout the chapter.

It is possible to determine the breakdown point for many estimators which are of interest in practice. For example, the sample mean of a dataset of n numbers has a breakdown point of $\frac{1}{n}$ while the median can withstand contamination of up to $n/2$ data points without exploding which is at the same time the best possible breakdown point for location estimators. In linear regression Rousseeuw and Leroy found (see [94]) the breakdown points for various estimators including Least Median of Squares.

Finally, for wavelet threshold estimators Kovac proves (see [71]) the obvious fact that hard and soft thresholding have both a breakdown point of 0. He also shows that the maximum distance between the given data set and the estimate is bounded by a constant γ which is proportional to the threshold, so that thresholding cannot remove outliers completely if the outlier is larger than the constant γ.

Before we analyse wavelet based methods with higher breakdown points in the next sections it is important to draw attention to a general problem of robust non-parametric regression. As long as we do not assume that the noise is independently distributed, but allow for outlier patches, where several outliers follow each other, the methods must offer the possibility to adjust the length of the largest outlier patch. Consider for example on the one hand a spectroscopic data set where it is not possible to deal with a large number of consecutive outliers, as long as the peaks from the signal shall not be removed. On the other hand we will see in Section 6.8 below a data set with extremely large outlier patches that are caused by the rope of a weather balloon which occasionally cuts the measuring device from the direct sun light. It is not possible that the same algorithm with the same parameters is able to cope with both situations—the statistician who applies the algorithm to the data has to adjust it such that it does not blur out interesting features of the underlying signal, but can cope with outlier patches that occur in the particular application.

6.2 Robust wavelet methods for data on a regular grid

We have seen in the last section that already a small number of outlying observations can have a disastrous effect on the wavelet estimator. Therefore we examine now different wavelet methods which have the property that they work well even when some data points are contaminated by outliers or when the distribution of the noise is considerably different from a normal distribution. The approaches in this section do all assume the time points are regularly spaced. On first sight, it might be tempting to relax this condition by transforming the given data on a regular grid, using the grid transformation that was extensively used in previous chapters. On the other hand, outliers in regions where the number of time points is relatively small would typically influence several grid points and aggravate the estimation procedure. Therefore, more sophisticated methods for the irregular case are discussed in the next sections.

One wavelet approach for dealing with outliers is due to Bruce, Donoho, Gao and Martin (see [15]) and part of the wavelet package *S+Wavelets* which was already mentioned several times. The algorithm is based on a robust wavelet transform which the authors refer to as *smoother/cleaner wavelets* and its idea is to modify the discrete wavelet transform as described in Chapter 2 by integrating a cleaner step before each application of the low and high-pass filters $\mathcal{H}$ and $\mathcal{G}$. This is done in the following manner: Suppose that c^j is the smooth component at level j.

1. A vector of "robust residuals" r^j is calculated by

$$r_i^j = \eta_S(c_i^j - \tilde{c}_i^j, \lambda)$$

where η_S is the soft thresholding function and $\tilde{c}^j$ a robust estimate of c^j, which can be calculated by running medians of length 5 (see Section 6.3). The threshold λ is chosen so that most of the robust residuals are zero. [1]

2. The cleaned smooth component u^j at level j is then defined as $u^j = c^j - r^j$.

3. The usual filter operations are now applied to u^j instead of c^j.

The advantages of this robust transform lie in the fact that in the Gaussian noise case the smoother/cleaner wavelet transform produces essentially the same decomposition as the classical wavelet transform while it is at the same time robust in the sense that outlier patches of length $2^l + 2$ only effect coefficients at levels $j \leq l$ instead of all levels.

The authors also suggest several approaches for combining the robust transform with wavelet thresholding. The simplest idea is to calculate the robust smoother/cleaner decomposition of the data, shrink small coefficients to zero and apply the inverse DWT to the data afterwards. This works perfectly if the data contain only isolated or pairs of outliers, as they are removed in the first step, before any filter operations are carried out. However, longer outlier patches are in general not removed correctly. If there are, for example, three large consecutive outliers and running medians of 5 are used, then the first cleaner operation will remove the first and the third outlier, e.g. replace it by the local median, slightly increased or decreased by the threshold λ. However, the second outlier will remain and will lead to one large wavelet coefficient at the finest scale.

[1] The authors suggest obtaining this value by simulation. Compare this with the theoretical results in Section 6.6.

To get around this problem two suggestions are made: A *zero cones* and a *clean and repeat* procedure. The idea behind the first method is that every patch of outliers will eventually be detected by some robust residual r_i^j. Then all coefficients at levels $j, \ldots, J-1$ that are computed from the outlier patch associated with r_i^j are shrunk to zero. In practice, this is only done for moderate or large robust residuals or, alternatively, in interaction with the user who can can select cones with the mouse.

The *clean and repeat* procedure consists simply of a multiple application of the robust transform and the inverse DWT to the data. The authors mention that in practice only two passes are necessary to clean the data.

Furthermore, Bruce et al. briefly discuss in the same paper the usability of l^1-fitting for wavelet decompositions. The idea is to choose the smooth component at each level such that the l^1-norm is minimised and not the l^2-norm. They also discuss the minimisation of a hybrid loss function like

$$G(t) = \begin{cases} t^2 & \text{for } |t| \leq C \\ \\ |t| & \text{for } |t| > C \end{cases}$$

since they realize that l^1-fits tend to be quite rough. However, a drawback with both methods is that they are computationally too inefficient for practical use. They do not make use of the recursive filtering pyramid which makes wavelet approaches so attractive.

Kovac analyses several other wavelet-based methods for robust non-parametric regression (see [71]), but they have in common that they are very slow. For example, one approach is based on the hope that soft thresholding, though not able to remove outliers, creates large residuals at data points which are contaminated by outliers. This leads to the idea to create a new data set which looks mainly like the original data except the data points with the largest differences are replaced either by the soft thresholding estimates or local medians. This procedure is repeated several times until the difference between two consecutive estimates is sufficiently small. Other ideas include approaches which try to recognise outliers and outlier patches from large wavelet coefficients as well as methods where the l^1-norm is minimised. Finally the most promising approach where outliers are detected by comparing the data points with the value of a median filter will be introduced and considerably improved in Section 6.4.

6.3 Median filters

In the last section a median filter appeared as a part of the robust smoothing technique of Bruce et al. We will make further use of a median filter during the next sections when we introduce robust wavelet techniques for unequally spaced data, so we review now some of its properties.

Median and median-type filters are widely used as a tool for robust signal recovery. Tukey (see [114] and [115]) is generally credited with the idea of introducing nonlinear filters based on moving sample medians of the input signal and he did the pioneering work in this area. A lot of interesting applications can be found throughout the engineering literature, ranging from image restoration (see [103]) and speech processing (see [126]) to sonar imagery (see [76]) and the analysis of mammograms (see [23]).

If $\{y_j\}_{j\in\mathbb{Z}}$ is a doubly infinite sequence, then the median-filtered sequence $\{\tilde{y}_j\}$ is given by the simple relationship

$$\tilde{y}_j = \operatorname{med}(y_{j-N}, y_{j-N+1}, \ldots, y_{j+N}).$$

where the window length N of the median filter is an integer.

Note that some gain in computational efficency can be obtained by using results about the order of $y_{j-N}, \ldots, y_{j+N}$ that are gathered during the computation of $\tilde{y}_j$ when determining the median of the next window $\tilde{y}_{j+1}$. Juhola et al. have analysed (see [68]) several ways in terms of their computational complexity

Theoretical analysis of median filters can be divided into two broad fields: One area is concerned with the analysis of certain signal structures that are invariant to median filtering, called root signals. For example, Gallagher and Wise (see [47]) and Tyan (see [116]) have shown that these roots are composed solely of combinations of regions of identically valued points of extent $N + 1$ points or more (constant neighbourhoods) and of regions of monotonically rising or falling points (edges) where all edges of opposite sign are separated by a constant neighbourhood. Gallagher and Wise have further shown that any finite length signal will be reduced to a root after a finite number of repeated passes by a median filter.

The second branch of theoretical research deals with distribution theory. Unfortunately, due to the non-linear nature of median filters, it is often very difficult to obtain results. A lot of material can be found in the monograph on order statistics by David (see [28]). However, he only considers order statistics that are obtained from the same sequence, but not from several windows. Bivariate output distributions of median filters have been examined by Kuhlmann and Wise (see [72]) and Justusson (see [69]), giving statements about the spectrum of the output distribution, and higher order distributions have been determined by Nodes and Gallagher (see [86]). Usually, the formulas for calculating distributions have exponential complexity in the number of random variables, but

Boncelet has recently developed (see [11] and [12]) recursive algorithms for evaluating distribution functions which have only polynomial complexity.

For our purposes median filters have several advantages: First of all they are robust in the sense that they do not break down unless there is at least one window which contains more than $N+1$ outliers. Thus, they can withstand outlier contamination of $N/(N+1)$ if the structure of the outliers is suitable, and they are resistant to outlier patches from length up to N provided that the next outlier patch is separated by enough uncontaminated data points. On the other hand, of course, the pessimistic view is that the median filter can collapse if the data contain $N+1$ consecutive outliers. This means that the breakdown point of the median filter as defined in Definition 6.1 is $N+1/n$ where n is the number of data points, and shows that the usage of Donoho-Hampel's breakdown point in nonparametric regression is limited as a consequence of the necessary localisation of smoothing techniques and in contrast to location estimators or linear regression where one or more parameters are estimated globally. It appears that statements like the one at the beginning of this paragraph are more helpful for characterising robustness of a smoothing technique.

Other advantages are that median filters can retain discontinuities very well, especially if the signal is locally either monotonically increasing or decreasing and that they are computationally very attractive. Maybe the most serious drawback is that they produce usually very rough estimates. Therefore, quite often a smoothing method is applied to the filtered data. For example, Sinha and Schunck (see [100]) propose a two-stage algorithm for surface reconstruction: In the first step the given data are mapped to a grid by a moving local LMS (least median of squares) approximation [2] while the second step fits a weighted

[2]Without going into too much detail it should be remarked that the LMS estimator may be viewed as a generalisation of the sample median for linear regression and yields the median as a special case when a

bicubic spline to the output of the local LMS filter. We will pick up this idea in Section 6.7 when we apply wavelet shrinkage to the output of a median filter, but first we examine in the next sections another application of the median filter and use it to recognise outliers.

6.4 Removing outliers of irregularly spaced data

One idea in non-parametric regression and other fields of statistics for dealing with data sets that are contaminated by outliers is to remove the outliers. This is discussed extensively by Hampel et al. (see [55]) and specific applications are examined, for example, by Carey et al. for hypothesis testing (see [18]), by Chaudhuri et al. for density estimation (see [19]) and by Chen et al. for principal components analysis (see [20]). Numerous tests for identifying outliers have been suggested, for example, see Hampel (see [54]), Hawkins (see [59]), Davies and Gather (see [30]) as well as Barnett and Lewis (see [8]).

However, regularly spaced classical thresholding cannot then be used even if the original data are regularly spaced because the cleaning procedure will create gaps and thus irregularly spaced data. Nevertheless, Kovac obtains smooth robust fits with a time-intensive algorithm (see [71]) where the gaps are filled with new data points in a way such that the L^2-norm of the wavelet coefficients is minimised. This technique can be viewed as another approach for dealing with a special case of irregularly spaced data where the time points lie all on grids, but have gaps inside.

---

constant function is fitted. While usual least squares regression minimises the sum of the squared residuals, LMS minimises the median of the squared residuals, and by doing that it attains a breakdown point of nearly $1/2$. It has two large disadvantages: First of all, the number of operations which is required for the computation of a LMS estimate increases exponentially with the number of data points, so that in practice only approximations are possible. The second drawback lies in the fact that the LMS is very inefficient—therefore quite often the LMS estimator is combined with weighted least squares regression. More details are provided by Rousseeuw (see [93]) and Rousseeuw and Leroy (see [94]).

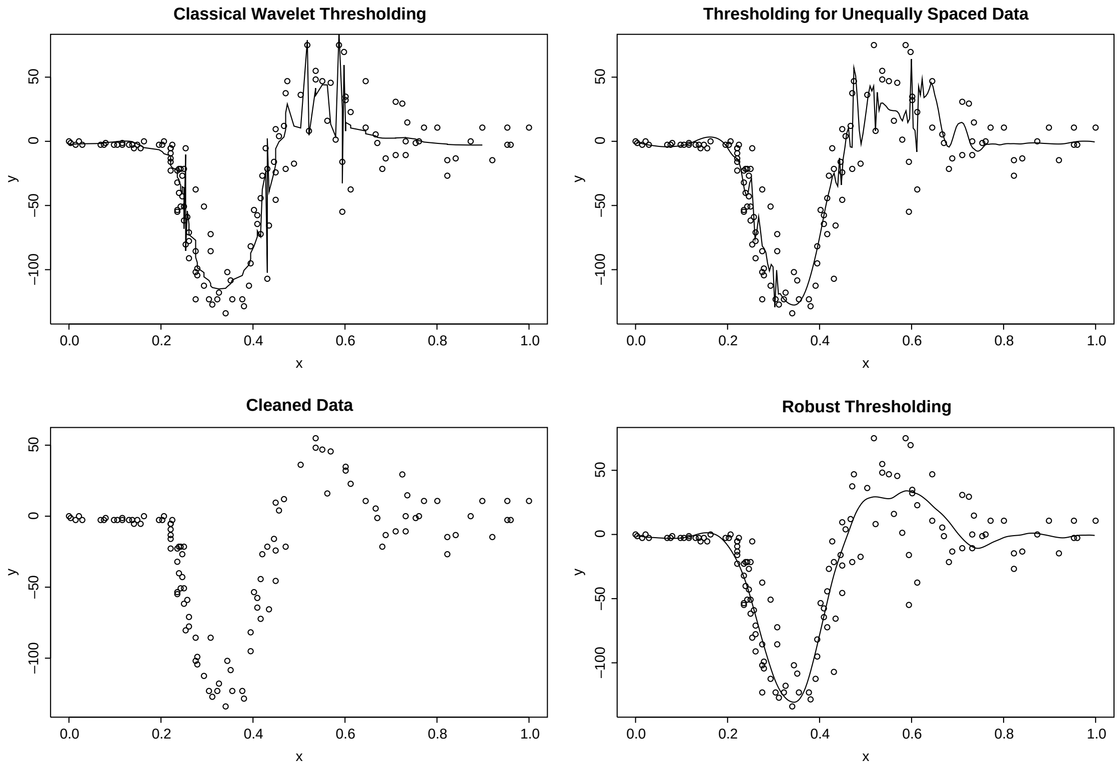

Figure 6.2: Haerdle's crash data with several wavelet estimates

On the other hand, our new algorithms for thresholding unequally spaced data can deal with this situation much more easily and also work if the original data are unequally spaced. We start with a motivating example and have a closer look at some refinements later.

Figure 6.2 shows a data set that has been analysed extensively in the field of non-parametric regression. It has been discussed by Silverman (see [97]) and consists of 133 observations from a crash test and shows the acceleration of a motorcyclist's head during a crash. However, classical wavelet thresholding and our new algorithms for unequally spaced data produce ugly curves like those in the first row of Figure 6.2. In both cases VisuShrink was applied and Daubechies' wavelet basis of extremal phase with six vanishing moments was used.[3] Both reconstructions exhibit large high frequency phenomena which look especially bad for classical thresholding. While this problem could be avoided by an unusual large threshold, the minimum of the curve would be overestimated in both cases which is already the case for classical thresholding.

One possible explanation for this problems is to regard some of the observations as outliers, and a reasonable solution would be to try to remove the outliers and extreme observations. We suggest the following simple algorithm:

Algorithm 6.1.

(1) *Estimate the variance of the data by calculating the MAD of the differences $y_{2j} - y_{2j-1}$ and dividing by $\sqrt{2}$:*

$$\hat{\sigma} = \mathrm{MAD}\{y_{2j} - y_{2j-1} \,|\, j \in \{1, \ldots, \lfloor n/2 \rfloor\}\}/\sqrt{2}. \tag{6.1}$$

(2) *Determine for each data point (t_i, y_i) the median m_i over a small window:*

$$m_i = \mathrm{med}\{y_j : |t_j - t_i| \leq \gamma\}$$

If the difference between a data point and the median is greater than $\lambda \cdot \hat{\sigma}$, then remove the point.

(3) *Apply the thresholding algorithm of the last chapter to the modified data set and use $\hat{\sigma}$ as an estimate of the noise level.*

This method was carried out with $\gamma = 0.03$ and $\lambda = 1.96$ and the same wavelet basis as above. The cleaned data set and the reconstruction from it can be seen in the second

[3] Moreover, classical wavelet thresholding was only applied to the first 128 observations to satisfy the condition that the number of data points has to be a power of two.

line of Figure 6.2. Obviously, the new estimates look much more credible. In particular there are no high frequency phenomena and the minimum is not seriously overestimated. We will see this data set again at the end of this chapter when we apply other robust wavelet regression techniques in Section 6.8.

Let us now proceed with remarks on several aspects of the algorithm.

- **Computational Complexity:** The computational complexity is still $O(n)$.

- **Estimation of the noise level:** The approach for estimating the noise level in step (1) of the algorithm can be viewed as the usual estimation procedure by means of a wavelet decomposition (see 2.18) where the Haar basis is used. Alternatively, another wavelet basis could be used, but the size of the corresponding filter should be very small, because otherwise possible outliers could have too much influence on the estimate.

- **Robustness:** The algorithm can collapse if there is a window $[t_j - \gamma, t_j + \gamma]$ where at least half of the observations in the window are outliers. Therefore, robustness depends strongly on the time structure and the choice of the window width. The extreme cases occur for $\gamma = 0$ and $\gamma = 1$. In the first case, all data points except those where several observations belong to the same time point are compared with themselves, so that essentially no outliers will be detected. In the second case the breakdown point is $\lceil n/2 \rceil$. However it is possible that nearly every data point is suspected to be an outlier.

- **Choice of the outlier threshold:** In the example, λ was chosen as 1.96 which corresponds to the 0.975-quantile of a standard normal distribution. The idea is to hope that the windows are small enough, so that that the signal does not show a lot of variation within the window. Furthermore, for independent normal noise $\mathcal{N}(\mu, \sigma^2)$ the distribution of the difference between a data point and its local median will roughly look like a normal distribution with mean 0 and variance σ^2. Therefore, it is reasonable to suspect data points which differ from its local median by more than $1.96 \cdot \sigma$ to be contaminated by an outlier. Note that this method will also remove five percent of the regular noise. A more precise choice of the outlier threshold based on the exact distribution of the difference between data point and local median will be suggested in Section 6.6.

- **Choice of the window width:** The choice of γ is maybe the most difficult part of the procedure. As we pointed out at the end of Section 6.1, this parameter has to be adjusted by the statistician who analyses the data. In general, there is a trade-off between an unnecessary large number of rejected data points and a high sensitivity against outlier patches. Therefore the filter width should be adjusted as small as possible, but still large enough such that the rejection procedure can cope with the outliers. The latter condition requires assumptions or previous knowledge about the distribution of the time points and the structure of the outliers. In the example, γ was chosen as 0.03 because this was the smallest number such that each window (except near the borders) contained at least five data points.

  The implication of the filter width on the number of "detected" outliers is illustrated in Figure 6.3 which shows the result of a short simulation study in S-Plus. The solid lines represent the average number of detected outliers over 30 replications

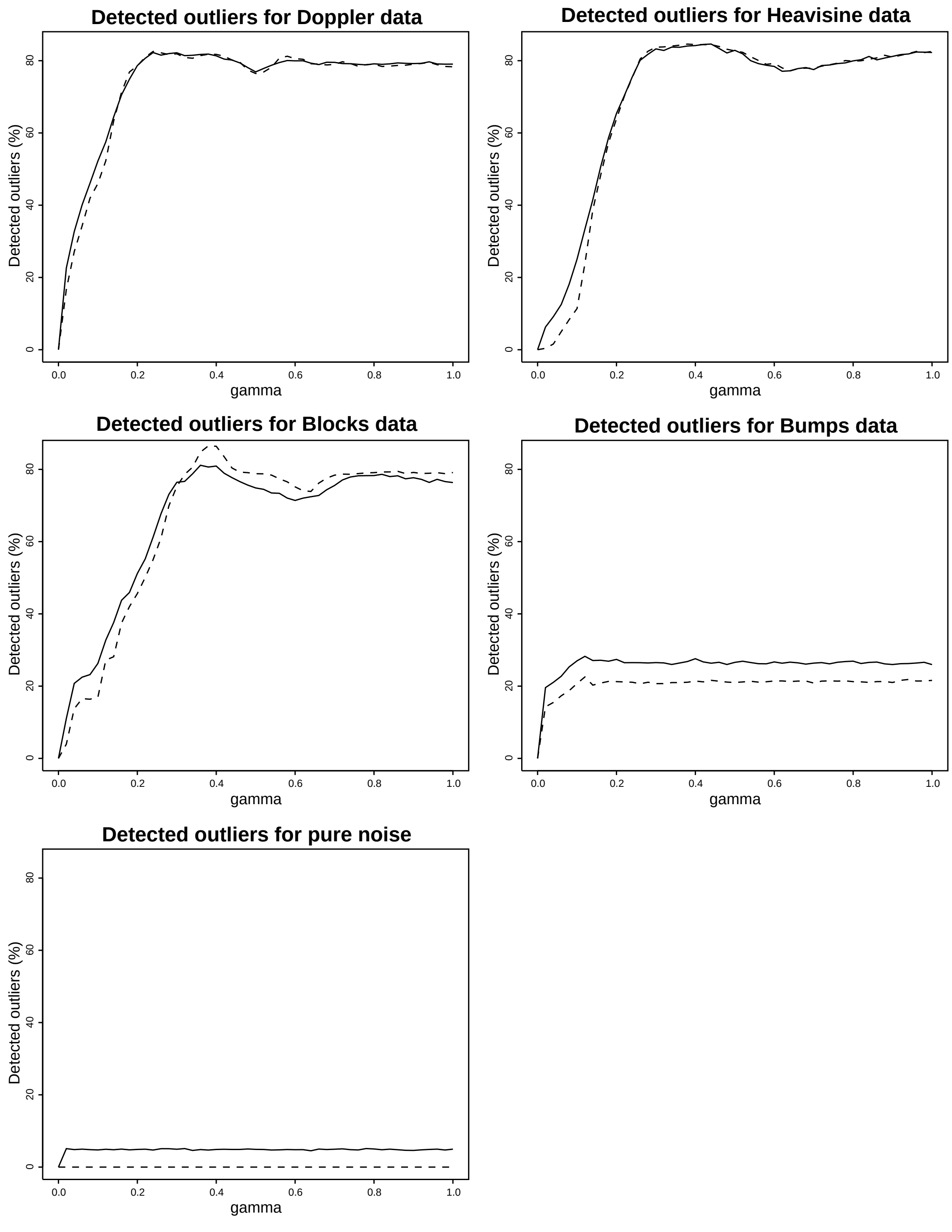

Figure 6.3: Numbers of detected outliers when applying algorithm 6.1 to the various test signals and to pure noise

for different values of γ and Donoho and Johnstone's four test signals as well as for the zero function. Each replication was created by drawing a random time sample of size 1000 from a $\mathcal{U}(0, 1)$ distribution, evaluating each function at the time points and adding white noise with $\sigma = 0.35$. Finally, the dashed lines show the result of the same procedure without adding white noise.

The result for the zero function which actually means applying a median filter to white noise confirms that nearly five percent of the data are removed as a consequence from the outlier threshold choice which corresponds to the 0.975-quantile of a normal distribution. The peaks of the Bumps signal are, of course, completely removed even if the window size is quite small, and the remaining points are all zero. Therefore it does not surprise that its curve is very similar to the curve for pure noise with the exception that a fixed amount of approximately 20 per cent data points is consistently declared as outliers.

The slope for the Doppler function is very large near 0. This reflects the fact that even for small window sizes outliers are detected as a consequence of the oscillation. On the other hand, once the threshold exceeds a certain limit, the local medians are always close to zero and the number of detected outliers is nearly invariant under different values for γ.

Although the graph for the Heavisine function looks very similar at first sight, there is an important difference for small window sizes γ which are, of course, most interesting for our purposes: The slope of the dashed line is initially very small, at least much smaller than the slope of the solid curve. This shows that most of the detected outliers are actually outliers and not extreme values of the signal. A similar effect can be seen for the Blocks function where the locally constant signal is again quite invariant under median filtering.

We will return to this algorithm in Section 6.8 where we compare several robust wavelet methods and present further examples.

6.5 Another approach for removing outliers

In the last section we introduced a robust method where outliers were removed from the data and wavelet thresholding was applied to the remaining data points. There is one disadvantage with this procedure: If large outlier patches can occur, then the width of the median window has to be chosen quite large. To illustrate this problem, let us assume for the moment that the time points are regularly spaced, then even an isolated outlier at position i which could probably be detected by a comparison with the median of $y_{i-2}, \ldots, y_{i+2}$ is actually compared with the median of 21 values if the procedure has been designed to detect ten consecutive outliers. If the variation of the underlying signal is large at some places, then it is possible that relatively mild isolated outliers are not noticed which would have been recognised in a smaller window.

It is interesting to see how Bruce et al. (see [15]) have solved this problem in their algorithm that we discussed in Section 6.2. They start with comparing the original data sequence with running medians of 5. This enables them to detect outliers which either are isolated or come in pairs. Next, they replace suspicious data points by some estimates (see above for details) and apply the low- and high-pass filter of the DWT. Let us assume that the Haar basis is used, then the next set of smooth coefficients is given by the scaled differences $(y_{2i} + y_{2i-1})/\sqrt{2}$. Each of these coefficients is subsequently compared with the local median of 5 smooth coefficients, and again new values are substituted for those coefficients where the difference to the median is larger than a threshold. Note that with each comparison the mean of two data points is compared with an location estimate of ten data points and that each replacement actually means that two points of the original data sequence are changed. In the next step the mean of four points is related to a mixture of median and means of 20 points and so on.

We will now consider a variation of this algorithm and replace the means by medians. In particular we suggest the following steps, still assuming that the time points are regularly spaced:

Algorithm 6.2.

(1) Start with $k := 0$.

(2) Define a vector z by

$$z_i = \mathrm{med}\{y_{i-2^k+1}, y_{i-2^k+2}, \ldots, y_{i+2^k-1}\}$$

where $i \in \{2^k, \ldots, n-2^k+1\}$.

(3) Determine for $i \in \{2^k, \ldots, n-2^k+1\}$ the local median

$$\hat{z}_i = \mathrm{med}\{z_{i-2^{k+2}+2}, z_{i-2^{k+1}+1}, z_i, z_{i+2^{k+1}-1}, z_{i+2^{k+2}-2}\}.$$

(4) For all i which satisfy $|z_i - \hat{z}_i| > \lambda_k$, mark $y_{i-2^k+1}, y_{i-2^k+2}, \ldots, y_{i+2^k-1}$ as suspected outliers.

(5) If protection against $2^{k+2}-1$ or more outliers is required, then increase $k := k+1$ and proceed with step (2) else remove all marked data points.

This procedure compares for $k = 0$ each data point with the median in a small window, and for $k > 0$ each local median with the median of the surrounding local medians. The sizes of the windows are adjusted such that theoretically in stage k up to $5 \cdot 2^k - 3$ consecutive outliers can be detected. If the signal is known to be very smooth, then a larger window might be used in the fourth step, for example running medians of 7, to increase the probability that relatively mild outliers can be recognised. Finally, the choice of the outlier thresholds λ_k is discussed in the next section.

Let us now try to generalise this algorithm in a way such that it can deal with irregularly spaced time points. Although there are several possibilities we only discuss one method which has the advantage that there is a fairly easy way for choosing the outlier threshold that is discussed in the next section. The approach is essentially based on two ideas: First, the goodness of the estimate $\hat{z}_i$ depends on the size of the window which is used for the computation of $\hat{z}_i$. Here we assume, of course, that we do not have any extra knowledge about the underlying signal, in particular about its local variation. Therefore we execute step (4) only for those indices i where the difference between the extreme time points that are used for the computation of $\hat{z}_i$ are bounded by a constant κ :

$$|t_{i-5\cdot 2^k+3} - t_{i+5\cdot 2^k-3}| < \kappa. \tag{6.2}$$

The second idea is to use another definition of outlier patches: Instead of considering a number of consecutive data points to be an outlier patch, we rather think of it as an interval $I \subset [0, 1]$ where besides Gaussian noise which influences the entire data set the signal is additionally disturbed by another effect. This model coincides with several real data sets where the reasons for outliers can easily be identified—see for example Bruce et al. (see [15]) for an application of their method to radar glint noise data, or a data set that we will analyse in Section 6.8 and that consists of measurements of radiation from a weather balloon where the direct radiation from the sun was occasionally cut off by the ropes. Therefore it makes sense to perform the outlier test in (4) only for those indices i where inequality (6.2) and the following condition are satisfied:

$$|t_{i-2^k+1} - t_{i+2^k-1}| < \mu. \tag{6.3}$$

This yields the following outlier detection algorithm for irregularly spaced data:

Algorithm 6.3.

(1) Start with $k := 0$.

(2) Define a vector z *by*

$$z_i = \operatorname{med}\{y_{i-2^k+1}, y_{i-2^k+2}, \dots, y_{i+2^k-1}\}$$

where $i \in \{2^k, \dots, n - 2^k + 1\}$.

(3) Determine for $i \in \{2^k, \dots, n - 2^k + 1\}$ *the local median*

$$\hat{z}_i = \operatorname{med}\{z_{i-2^{k+2}+2}, z_{i-2^{k+1}+1}, z_i, z_{i+2^{k+1}-1}, z_{i+2^{k+2}-2}\}.$$

(4) For all i *which satisfy* $|z_i - \hat{z}_i| > \lambda_k$ *and the conditions* (6.2) *and* (6.3) *mark* $y_{i-2^k+1}, y_{i-2^k+2}, \dots, y_{i+2^k-1}$ *as suspected outliers.*

(5) If both conditions (6.2) *and* (6.3) *are satisfied for at least one index* i*, then increase* $k := k + 1$ *and proceed with step (2) else remove all marked data points.*

While the choice of the outlier thresholds λ_k is discussed in the next section, the constants κ and μ depend on the particular application: κ depends on assumptions about the variation of the underlying signal and μ determines the largest outlier patch we want to check for. In general, however, it seems quite natural to choose the parameters such that $\kappa = 5\mu$ because running medians of 5 are used.

Before we end this section we introduce shortly a simplification of the algorithms that were developed in this section: Instead of calculating $\hat{z}_i$ as a median of several local medians, we simply choose $\hat{z}_i$ as the median

$$\hat{z}_i = \operatorname{med}\{y_{i-5\cdot 2^k+3}, \ldots, y_{i+5\cdot 2^k-3}\}.$$

We analyse some implications of this simplifications in the next section among other things.

6.6 The choice of the outlier thresholds

In the last two sections we have introduced several outlier tests that have in common that one data point is suspected to be an outlier if its distance to a local median is larger than a threshold, and consecutive outliers are treated by comparing a local median with either a median of five local medians or with another local median that is calculated of more points. The question arises how the thresholds should be chosen.

Bruce et al. (see [15]) set λ in their algorithm such that an average of $100 \times p$ percent outliers are "detected" in the Gaussian case where p is set to some small value (e.g. .01). An approximation is obtained by simulation based on the Gaussian model. We adopt their idea, but calculate the exact distribution of each difference mentioned above and use the upper $p/2$ quantile as the threshold.

Everything is based on the following theorem 6.1 where the binomial coefficients $\binom{n}{k}$ shall be defined as 0 if $n < k$.

Theorem 6.1. *Let j and n be odd integers and $X_1, \ldots, X_n$ i.i.d. random variables with distribution function F_X and a symmetric density function f_X, so that $f_X(s-x) = f_X(s+x)$ for some $s \in \mathbb{R}$ and all $x \in \mathbb{R}$. Furthermore denote $m = (n+1)/2$. Then the distribution function F^Y of*

$$Y = \operatorname{med}(X_1, \ldots, X_j) - \operatorname{med}(X_1, \ldots, X_n)$$

is given by

$$F^Y(q) = \sum_{k=1}^{n} \frac{\binom{k-1}{(j-1)/2} \cdot \binom{n-k}{(j-1)/2}}{\binom{n}{j}} \cdot F_k(q)$$

where

$$F_m(q) = 1_{\{q \geq 0\}},$$

for $k > m$

$$F_k(q) = \begin{cases} 0, & \text{if } q < 0 \\ \frac{n!}{(m-1)!(k-m-1)!(n-k)!} \sum_{l=0}^{k-m-1} \binom{k-m-1}{l} \int_{-\infty}^{\infty} F_X(x)^{k-m-1-l} \cdot & \\ \cdot\, (1-F_X(x))^{n-k} \cdot \frac{(-1)^l}{l+m} \cdot \left(F_X(x)^{l+m} - F_X(x-q)^{l+m}\right) f_X(x)dx & \text{if } q \geq 0. \end{cases} \tag{6.4}$$

and for $k < m$

$$F_k(q) = 1 - F_{2m-k}(-q).$$

Proof. We denote by $\mathrm{R}(\operatorname{med}(X_1, \ldots, X_j))$ the rank of the submedian $\operatorname{med}(X_1, \ldots, X_j)$ within the entire sequence $X_1, \ldots, X_n$. Note that R is well-defined because we assume that j is odd and hence $\operatorname{med}(X_1, \ldots, X_j)$ is an element of the sequence $X_1, \ldots, X_n$.

We can now write the distribution function of Y as

$$F^Y(q) = \mathbb{P}(\operatorname{med}(X_1, \ldots, X_j) - \operatorname{med}(X_1, \ldots, X_n) \leq q) = \sum_{k=1}^{n} \alpha_k \cdot F_k(q)$$

where

$$\alpha_k = \mathbb{P}\left(\mathrm{R}\left(\mathrm{med}(X_1, \dots X_j)\right) = k\right)$$

and

$$F_k(q) = \mathbb{P}\left(\mathrm{med}(X_1, \dots, X_j) - \mathrm{med}(X_1, \dots, X_n) \le q \mid \ \mathrm{R}\left(\mathrm{med}(X_1, \dots X_j)\right) = k\right). \tag{6.5}$$

In the remainder of this proof we will have to check the expressions for α_k and $F_k(q)$.

Let us start with α_k. There are obviously $\binom{n}{j}$ different combinations of ranks that can be occupied by $X_1, \dots X_j$, and each one can occur with the same probability. We have to count for how many combinations the submedian has rank k. Obviously, this requires that one member of the subsequence $X_1, \dots, X_j$ has rank k and exactly half of the ranks of the remaining elements must be smaller than k. There are only $k-1$ ranks available lower than k and $n-k$ ranks greater than k. Therefore, the condition cannot be satisfied if either $(j-1)/2 > k-1$ or $(j-1)/2 > n-k$. Otherwise, we can choose between $\binom{k-1}{(j-1)/2}$ combinations of ranks lower than k and $\binom{n-k}{(j-1)/2}$ combinations of ranks greater than k. Thus, we can arrange the ranks of the subsequence $X_1, \dots, X_j$ in

$$\binom{k-1}{(j-1)/2} \cdot \binom{n-k}{(j-1)/2}$$

ways, so that that the middle element is at position k. Dividing by $\binom{n}{j}$ yields the required formula for α_k.

Let us turn now to the computation of $F_k(q)$. We can get rid of the conditional probability in (6.5) by writing $F_k(q)$ as

$$F_k(q) = \mathbb{P}(X_{(k)} - X_{(m)} \le q). \tag{6.6}$$

If $k = m$ then, of course, $F_m(q) = 1_{\{q \geq 0\}}$ is obvious. Furthermore, if $k < m$ then $X_{(k)} - X_{(m)}$ has the same distribution as $-(X_{(2m-k)} - X_{(m)})$ because the variables $X_1, \ldots, X_n$ are symmetrically distributed, and so we get the relationship

$$F_k(q) = \mathbb{P}(-(X_{(2m-k)} - X_{(m)}) \leq q) = 1 - \mathbb{P}(-(X_{(2m-k)} - X_{(m)}) > q) = 1 - F_{2m-k}(-q).$$

We consider now the case $k > m$ in more detail. The case $q < 0$ is trivial because $X_{(k)}$ is by definition always greater or equal $X_{(m)}$, so that $F_k(q) = 0$, therefore we assume from now on that $q \geq 0$. By noting that the right-hand side of (6.6) is independent of a specific order of $X_1, \ldots, X_n$ we can rewrite (6.6) as

$$F_k(q) = n! \cdot \mathbb{P}(X_k - X_m \leq q \text{ and } X_1 < X_2 < \cdots < X_n). \tag{6.7}$$

We will explore the probability on the right-hand side of (6.7) a little further and write it as a multiple integral:

$$\begin{aligned}\mathbb{P}(X_k - X_m \leq q \text{ and } X_1 < X_2 < \cdots < X_n) = \int \cdots \int \{x_1 < x_2 < \cdots < x_n\} \cdot \\ \cdot \{x_k - x_m \leq q\} \cdot f_X(x_1) \cdot f_X(x_2) \cdots f_X(x_n)\, dx_1 dx_2 \cdots dx_n.\end{aligned} \tag{6.8}$$

Most of the integrals can be eliminated by using the following simple probabilistic statement that holds for every sequence of i.i.d. random variables $Z_1, \ldots, Z_n$ with distribution function F and numbers $a, b \in \mathbb{R} \cup \{-\infty, \infty\}$ with $a < b$:

$$\mathbb{P}(a < Z_1 \leq Z_2 \leq \cdots \leq Z_n \leq b) = \frac{1}{n!}(F(b) - F(a))^n. \tag{6.9}$$

The proof is obtained very easily by rewriting the left-hand side

$$\begin{aligned}&\mathbb{P}(a < Z_1 \leq Z_2 \leq \cdots \leq Z_n \leq b) \\ &= \mathbb{P}(a < Z_1 \leq b \text{ and } a < Z_2 \leq b \text{ and } \ldots \text{ and } a < Z_n \leq b \text{ and } Z_1 \leq Z_2 \leq \cdots \leq Z_n)\end{aligned} \tag{6.10}$$

and noting that the event that each variable Z_i lies between a and b is independent from the special order $Z_1 \leq Z_2 \leq \cdots \leq Z_n$. Writing the last expression as a product of probabilities and evaluating them gives the right-hand side of (6.9). [4]

We apply now statement (6.9) three times to equation (6.8). First to the subsequence $X_1, \ldots, X_{m-1}$ with $a = -\infty$ and $b = x_m$, then to $X_{m+1}, \ldots, X_{k-1}$ with $a = x_m$ and $b = x_k$ and finally to $X_{k+1}, \ldots, X_n$ with $a = x_k$ and $b = \infty$. Doing this we obtain

$$\begin{aligned}
&\mathbb{P}(X_k - X_m \leq q \text{ and } X_1 < X_2 < \cdots < X_n) \\
&\quad = \iint \tfrac{1}{(m-1)!(k-m-1)!(n-k)!} F_X(x_m)^{m-1} \cdot (F_X(x_k) - F_X(x_m))^{k-m-1} \cdot \\
&\qquad \cdot (1 - F_X(x_k))^{n-k} \cdot \{x_k - q < x_m < x_k\} \cdot f_X(x_k) f_X(x_m) dx_k dx_m \\
&\quad = \tfrac{1}{(m-1)!(k-m-1)!(n-k)!} \sum_{l=0}^{k-m-1} \textstyle\int_{-\infty}^{\infty} (1 - F_X(x_k))^{n-k} \binom{k-m-1}{l} \cdot F_X(x_k)^{k-m-1-l} \cdot \\
&\qquad \cdot (-1)^l \cdot \left(\int_{x_k - q}^{x_k} F_X(x_m)^{l+m-1} f_X(x_m) dx_m \right) \cdot f_X(x_k) dx_k \qquad (6.11) \\
&\quad = \tfrac{1}{(m-1)!(k-m-1)!(n-k)!} \sum_{l=0}^{k-m-1} \binom{k-m-1}{l} \textstyle\int_{-\infty}^{\infty} (1 - F_X(x_k))^{n-k} \cdot F_X(x_k)^{k-m-1-l} \cdot \\
&\qquad \cdot \tfrac{(-1)^l}{l+m} \cdot (F_X(x_k)^{l+m} - F_X(x_k - q)^{l+m}) \cdot f_X(x_k) dx_k.
\end{aligned}$$

Finally, we can substitute this expression in equation (6.7) and receive the desired formula. This proves the theorem. □

The distribution of the differences $X_{(k)} - X_{(m)}$ which is determined in the proof has also been examined by David who derived (see [28]) an expression for the density function which integrates nicely to equation (6.4) for $F_k(q)$.

As long as the value of n is quite small, the result of this theorem can be used directly to determine suitable thresholds for algorithm 6.1 where outlier rejection is based on

[4] If F has a density function with respect to the Lebesgue measure, then alternatively a technical proof by induction is possible where the probability is expressed by a multiple integral and the number of integral signs is successively reduced with each induction step.

j	1	1	1	1	1	1	1	1	1
n	3	5	7	9	11	13	15	17	19
p=0.05	1.956	1.984	1.985	1.982	1.980	1.977	1.976	1.974	1.973
p=0.01	2.737	2.699	2.670	2.653	2.640	2.631	2.625	2.619	2.615
j	1	1	1	1	1	1	3	3	7
n	21	23	25	27	29	31	15	21	35
p=0.05	1.972	1.971	1.970	1.970	1.969	1.969	1.298	1.305	0.870
p=0.01	2.611	2.607	2.605	2.604	2.602	2.601	1.760	1.755	1.176

Table 6.1: Thresholds for algorithm 6.1 and the simplified version of algorithm 6.3

the difference between a data point and its local median. It can also be used for the simplification of algorithm 6.3 where two local medians are compared (see the end of the previous section). The integral in (6.4) was discretized, and approximations of the $0.975-$ and 0.995-quantiles were determined in S-Plus (see table 6.1) for the Gaussian model where F_X and f_X are distribution and density functions of the standard normal distribution.

The problem with this method is that the accuracy decreases rapidly for larger values of n. This is due to the computation of $F_k(q)$ where the expression for

$$\gamma_{(k:n)} = \frac{n!}{(m-1)!(k-m-1)!(n-k)!}$$

increases faster than exponentially because, for example,

$$\frac{\gamma_{(n+2:n+2)}}{\gamma_{(n:n)}} = \frac{(n+1)\cdot(n+2)}{\frac{n+1}{2}\cdot n - 12} = 4\cdot\left(1+\frac{5}{n-1}\right),$$

Consequently the sum in (6.4) must decrease very fast because $F_k(q)$ lies between 0 and

j	1	1	2	2	3	3
n	5	7	5	7	5	7
p=0.05	1.330	1.330	0.911	0.911	0.633	0.633
p=0.01	1.813	1.795	1.241	1.228	0.862	0.853
j	4	4	5	5	6	6
n	5	7	5	7	5	7
p=0.05	0.444	0.444	0.312	0.312	0.220	0.220
p=0.01	0.604	0.597	0.425	0.421	0.302	0.306

Table 6.2: Outlier threshold values which are used by algorithm 6.3 and its simplification. The median of j neighboured data points is compared with the median of a supset of size n. The difference between the medians is larger than the values in the third and fourth row with probability 0.05 and 0.01.

1. On the other hand the binomial coefficients are also increasing, so that a lot of large numbers cancel each other nearly out, leading to a numerically ill-posed algorithm.

We will discuss an alternative possibility for the computation of $F_k(q)$ further below, but before we do that we describe how thresholds for algorithm 6.3 can be obtained. Fortunately, the values for j and n are always 1 and 5 (or 7 if a slightly larger window is required), so that the discretizing approach works very well in this situation. In order to apply the theorem we have to know the distribution and density functions F^Y and f^Y of the median of k independently distributed standard normal random variables $Y = \operatorname{med}(X_1, \dots, X_k)$. These functions are obtained very easily: Slightly more generally, the distribution of the rth largest element $X_{(r)}$ of k random variables with distribution function F is given by

$$\begin{aligned}\mathbb{P}(X_{(r)} \le s) &= \mathbb{P}(\text{at least } r \text{ of the } X_i \text{ are less or equal to } s) \\ &= \sum_{i=r}^{k} \mathbb{P}(\text{exactly } i \text{ of the } X_i \text{ are less or equal to } s) \qquad (6.12) \\ &= \sum_{i=r}^{k} \binom{k}{i} F(s)^i \cdot (1 - F(s))^{k-i}.\end{aligned}$$

In particular, of course, the distribution of the median is obtained by setting $r = \frac{k+1}{2}$, assuming that k is odd. Moreover, it is not difficult to calculate the quantiles that are needed for algorithm 6.3 (see table 6.2).

We develop now a procedure for calculating $F_k(q)$ and hence suitable thresholds for algorithm 6.1 and the simplification of algorithm 6.3 which works well even if n is quite large. It was inspired by ideas of Boncelet (see [11]) who developed an recursive algorithm for computing the distribution function of $X_{(k_1)}, X_{(k_2)} \ldots, X_{(k_n)}$.

Let $X_{(k:n)}$ be the kth largest element of random variables $X_1, \ldots, X_n$. Furthermore we use the notation

$$F^{(n)}(s,t;\ k,l) = \begin{cases} \mathbb{P}(X_{(k:n)} \le s \wedge X_{(l:n)} \le t) & \text{if } 0 < k < l \le n \\ \mathbb{P}(X_{(l:n)} \le t) & \text{if } 0 = k < l \le n \\ 0 & \text{if } 0 < k \le n < l \\ \mathbb{P}(X_{(k:n)} \le s) & \text{if } 0 < k = l \le n. \end{cases}$$

If we assume that the X_i are independently distributed, then the relation

$$\begin{aligned}F^{(n+1)}(s,t;\ k,l) &= F^{(n)}(s,t;\ k-1,l-1)\mathbb{P}(X_{n+1} \le s) \\ &\quad + F^{(n)}(s,t;\ k,l-1)\mathbb{P}(X_{n+1} \le s) \qquad (6.13) \\ &\quad + F^{(n)}(s,t;\ k,l)\mathbb{P}(X_{n+1} \le s)\end{aligned}$$

holds for all integers $1 \leq k < l \leq n+1$ and real numbers $s < t$ and is essentially proved by distinguishing the several cases that can occur. Moreover it is a special case of a more general theorem of Boncelet (see [11]). Although we assume from now on that the variables X_i are identically distributed with distribution function F it is worth noticing that the proof of the statement only requires independence.

Equation (6.13) can obviously be used to determine $F^{(n)}(s,t;\ k,l)$ recursively. The (univariate) distributions of the order statistics that occur for $0 = k < l \leq n$ and $0 < k = l \leq n$ can either be calculated from equation (6.12) or, somewhat faster, by another two recurrence relations of David and Shu (see [29]) which hold for $r \in \{1, \ldots, n-1\}$:

$$\mathbb{P}(X_{(r:n)} \leq s) = \mathbb{P}(X_{(r+1:n)} \leq s) + \binom{n}{r} F^r(s)(1-F(s))^{n-r}$$
$$\mathbb{P}(X_{(r:n)} \leq s) = \mathbb{P}(X_{(r:n-1)} \leq s) + \binom{n-1}{r-1} F^r(s)(1-F(s))^{n-r}.$$

Finally, suitable start values are, for example, given by

$$F^{(2)}(s,t;\ 0,1) = 2F(t)\cdot(1-F(t)) + F(t)^2$$
$$F^{(2)}(s,t;\ 0,2) = F(t)^2$$
$$F^{(2)}(s,t;\ 1,1) = 2F(s)\cdot(1-F(s)) + F(s)^2$$
$$F^{(2)}(s,t;\ 1,2) = 2F(s)\cdot F(t) - F(s)^2$$
$$F^{(2)}(s,t;\ 2,2) = F(s)^2$$
$$F^{(2)}(s,t;\ 1,3) = F^{(2)}(s,t;\ 2,3) = 0.$$

j	7	15	15	31
n	49	75	105	155
p=0.05	0.880	0.595	0.606	0.405
p=0.01	1.182	0.802	0.812	0.561

Table 6.3: Further outlier threshold values which are used by the simplification of algorithm 6.3. The median of j neighboured data points is compared with the median of a supset of size n. The difference between the medians is larger than the values in the third and fourth row with probability 0.05 and 0.01.

We can use this for the calculation of $F_k(q)$ by noting that for $k > m$ and $q > 0$

$$\begin{aligned}
F_k(q) &= \mathbb{P}(X_{(k:n)} - X_{(m:n)} \leq q) \\
&= \sum_{\mu \in \mathbb{Z}} \mathbb{P}\left(X_{(k:n)} - X_{(m:n)} \leq q \wedge X_{(m)} \in (-q + \mu\varepsilon, -q + (\mu+1)\varepsilon]\right) \\
&\leq \sum_{\mu \in \mathbb{Z}} \mathbb{P}\left(X_{(k:n)} \leq (\mu+1)\varepsilon \wedge X_{(m)} \in (-q + \mu\varepsilon, -q + (\mu+1)\varepsilon]\right) \\
&\leq \sum_{\mu \in \mathbb{Z}} \left(F^{(n)}(m, k;\ (\mu+1)\varepsilon - q, (\mu+1)\varepsilon - F^{(n)}(m, k;\ (\mu+1)\varepsilon - q, \mu\varepsilon\right)
\end{aligned}$$

where the right side converges to $F_k(q)$ if the distribution F of X_i has a density function with respect to the Lebesgue measure. In a similar manner a lower bound for $F_k(q)$ can be obtained and the average of both bounds can be used as an approximation of $F_k(q)$ while the distance between upper and lower bound can be regarded as a measure of the goodness of the approximation.

The entire procedure was applied to the Gaussian model with $\varepsilon = 0.01$ and $\mu \in \{-600, \ldots, 600\}$. It was combined with Theorem 6.1 to calculate the $0.975-$ and $0.995-$quantiles of the distributions of $\text{med}(X_1, \ldots, X_j) - \text{med}(X_1, \ldots, X_n)$ for sev-

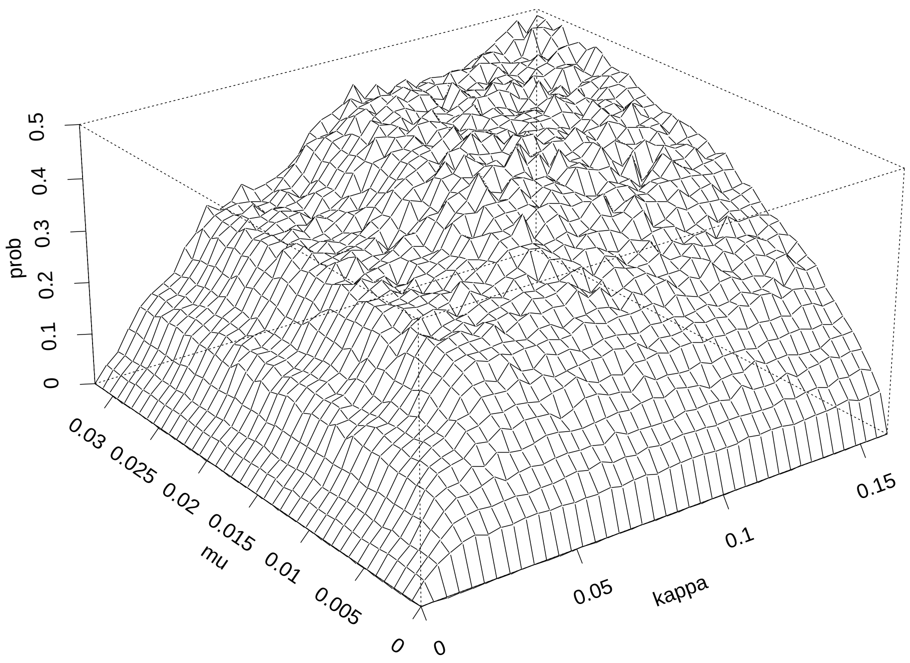

Figure 6.4: Number of detected outliers when applying algorithm 6.3 to pure noise for different parameter choices

eral values of j and n, including those combinations which are shown in table 6.3 where some quantiles can be found that are used by the simplification of algorithm 6.3. The complete results are stored in a file which can be obtained from the web page

`http://feat.mathematik.uni-essen.de/~kovac/thesis.html`

Let us finally take a short glimpse on how many data points are removed in practice: Figure 6.4 shows the influence of κ and μ on the amount of outliers found in normal noise. For each value of κ and μ 20 samples of size 1000 from a normal distribution were drawn. For each replication a further sample of size 1000 from a uniform distribution was drawn, algorithm 6.1 was applied and the number of detected outliers was determined.

The figure shows the average number of detected outliers over all replications and demonstrates that the algorithm tends to remove a lot of data points even for small values of κ or μ. It can be seen that less than a quarter of of the data is removed if either μ is smaller than 0.0064 or κ smaller than 0.36.

6.7 Wavelet thresholding of median-filtered data

We introduce now another method which differs from the other approaches in that no outlier detection step, followed by removals or replacements of suspicious data points, is performed. Moreover, the assumption that the noise is normally distributed can be relaxed—just continuity of its distribution function will turn out to be helpful for numerical purposes.

Median filters have played an important role in the preceding sections. We have introduced some of their properties in Section 6.3 and one result was that they usually fit the data very well even when the data contain outliers, but that they produce relatively rough curves. It seems reasonable to use a smoothing technique like wavelet thresholding to smooth the output of the median filter, so we propose a two-stage algorithm where first a median filter and then wavelet thresholding is applied to the data. In particular we consider the following algorithm for some integer J and window width γ.

Algorithm 6.4.

(1) Transform the given data to a grid of size 2^J by determining for each new time point $\tilde{t}_i = (i - 0.5)/2^J$ the index set

$$I_i = \{i \in \{1, \ldots, n\} : |t_j - \tilde{t}_i| \leq \gamma\}.$$

If I_i is empty for some integer i, then replace it by $I_i = \{k\}$ where k is the index of the time point with minimum distance to $\tilde{t}_i$.

If the size of the ith index set I_i is even, then replace it by $I_i \setminus \{k\}$ where k is the index of the time point with the largest distance to $\tilde{t}_i$.

Finally calculate for each $i \in \{1, \ldots, 2^J\}$ the local median

$$m_i = \operatorname{med}\{y_j : j \in I_i\}.$$

(2) Apply wavelet thresholding to the local medians m_i with coefficient-dependent thresholds $\tau_{j,k}$.

The part of the first step where each even-sized index set is turned into an odd-sized set by removing one index is only necessary to perform the threshold selection mechanism which is described below and makes use of Theorem 6.2 which requires windows that consists of an odd number of data points.

As in the previous sections we have to discuss the choice of several parameters. The size of the window of the median filter depends on the same aspects as for the other algorithms: The size must be large enough to ensure that outlier patches have no influence on the estimation, but should be as small as possible, in order not to produce unnecessary bias.

The selection of the thresholds is much more difficult. One problem is that the wavelet coefficients are not as nicely distributed as in the situation that was discussed in previous chapters where they simply followed a multivariate normal distribution. Virtually no published work has been concerned with the distribution of a linear transformation that is applied to a median filter, although Velleman suggested (see [117]) as early as 1980 to smooth the result of the median filter by a linear averaging method. An exact, but very long and complicated formula for the multivariate output distribution of the median filter has been derived by Nodes and Gallagher (see [86]), and Boncelet has recently extended (see [12])) the recursive algorithm which was mentioned in the last section to deal with median filters. Unfortunately, it is still a long way to obtain, for example, exact quantiles for a linear transformation of running medians, and in the remainder we will use simulation to determine suitable thresholds.

The algorithm for the choice of the thresholds was inspired by some recent work of Peterson (see [89]). He considers a wavelet-like (but robust) translation-invariant transform and suggests to apply the transform to simulated Gaussian noise and to determine for each replication the maximum in each level. Level-dependent thresholds are then chosen by determining suitable quantiles of the maxima in each level.

Our approach is quite similar—the main difference is that we take the different variances of the coefficients into account that occur because each new grid point possibly depends on a changing number of data points if the time data are irregularly spaced. In particular we propose to perform the following steps:

Algorithm 6.5.

(1) Generate m independently $\mathcal{N}(0,1)$ distributed samples of size n and apply the median filter and the wavelet transform to it, getting wavelet coefficients $w_{j,k}^{(l)}$.

(2) Calculate the variance $c_{j,k}$ of each coefficient in the way described further below.

(3) Determine for each level j and replication l the maximum $m_j^{(l)}$ of the normalised

coefficients in that particular level and replication:

$$m_j^{(l)} = \max_k w_{j,k}^{(l)} / \sqrt{c_{j,k}}.$$

(4) *Find for each level j the 0.9-quantile of the maxima $m_j^{(l)}$ where $l \in \{1, \ldots, m\}$.*

Note that the normal distribution in the first step can easily be replaced by any other distribution which is considered to be a better model for the noise. Moreover, the 0.9-quantile in step (4) was chosen to get a similar performance as Donoho and Johnstone's VisuShrink: Note that for uncorrelated Gaussian noise:

$$\mathbb{P}(\max(X_1, \ldots, X_n) \leq \lambda) = \Phi(\lambda)^n$$

and that the special choice $\lambda = \sqrt{2 \log(n)}$ yields probabilities around 0.9 for sample sizes between 256 and 2048. Thus, the estimator that is based on the threshold choice given in algorithm 6.5 coincides essentially with Donoho and Johnstone's VisuShrink for equally spaced data and filter width $\gamma < 1/n$ such that the median filter is the identity.

We turn now to the computation of the variances of the wavelet coefficients. Obviously, it is sufficient to restrict ourselves to the calculation of the covariance matrix of the filter output because this matrix can be combined with our fast algorithm of Section 3 to give the required variances. First we derive an exact formula for the bivariate output distribution of a median filter.

Theorem 6.2. *Let $X_1, \ldots, X_n$ be i.i.d. random variables with distribution function F and consider the submedians Y_i of windows of odd size n_i*

$$Y_1 = \mathrm{med}(X_1, \ldots, X_{n_1})$$

and

$$Y_2 = \mathrm{med}(X_{n-n_2+1}, \dots, X_n),$$

such that both windows overlap. Denote the number of overlapping points by k:

$$k = n_1 + n_2 - n.$$

Then the common distribution of Y_1 and Y_2 is for real numbers $y_1 \le y_2$ given by

$$\begin{aligned} F^{(Y_1,Y_2)}(y_1, y_2) = \sum_{\mu=0}^{k} \sum_{\kappa=0}^{k-\mu} & \left(\sum_{\nu=\max(\frac{n_1+1}{2}-\mu,0)}^{n_1-k} \binom{n_1-k}{\nu} F(y_1)^\nu (1-F(y_1))^{n_1-k-\nu} \right) \\ & \cdot \left(\sum_{\nu=\max(\frac{n_2+1}{2}-\mu-\kappa,0)}^{n_2-k} \binom{n_2-k}{\nu} F(y_2)^\nu (1-F(y_2))^{n_2-k-\nu} \right) \\ & \cdot \binom{k}{\mu} \cdot \binom{k-\mu}{\kappa} F(y_1)^\mu (F(y_2)-F(y_1))^\kappa (1-F(y_2))^{k-\mu-\kappa}. \end{aligned} \tag{6.14}$$

Proof. The proof is not very difficult. First note that

$$\begin{aligned} F^{(Y_1,Y_2)}(y_1, y_2) = \sum_{\mu=0}^{k} \sum_{\kappa=0}^{k-\mu} \mathbb{P}(Y_1 \le y_1, Y_2 \le y_2, & \text{exactly } \mu \text{ overlapping points are} \\ & \text{smaller than } y_1 \text{, and exactly } \kappa \text{ points lie between } y_1 \text{ and } y_2). \end{aligned} \tag{6.15}$$

The submedian Y_1 is smaller than y_1 if and only if at least $\frac{n_1+1}{2}$ variables of $X_1 \dots, X_{n_1}$ are smaller than y_1. If exactly μ of the k overlapping variables are smaller than y_1, then this condition is satisfied if and only if the ρth largest element of the sequence $X_1 \dots, X_{n_1-k}$ is smaller than y_1 where $\rho = \max(\frac{n_1+1}{2} - \mu, 0)$. Similar arguments can be found for Y_2 and the independence of the input variables shows that the equation above is equivalent to

$$
\begin{aligned}
&F^{(Y_1,Y_2)}(y_1,y_2) = \\
&\qquad \sum_{\mu=0}^{k}\sum_{\kappa=0}^{k-\mu} \mathbb{P}(X_{(\max(\frac{n_1+1}{2}-\mu,0):n_1-k)} \leq y_1)\cdot \mathbb{P}(X_{(\max(\frac{n_2+1}{2}-\mu-\kappa,0):n_2-k)} \leq y_2) \\
&\qquad \cdot \mathbb{P}(\text{exactly } \mu \text{ overlapping points are smaller than } y_1, \text{ and exactly} \\
&\qquad \kappa \text{ points lie between } y_1 \text{ and } y_2).
\end{aligned}
\tag{6.16}
$$

The probabilities in this formula are now easily evaluated using the simple statement (6.12) about order statistics and basic probability theory, then yields equation (6.14). □

Similar expressions were previously obtained by Kuhlmann and Wise (see [72]) who derived the bivariate distribution for the situation where every window is of the same size, and the theorem is of course a special case of the more generally results by Nodes and Gallagher (see [86]) where the multivariate distribution was analysed.

Theorem 6.2 can be used to calculate the covariances of the filter output. The idea is to approximate $\mathbb{E}(Y_1Y_2)$ by dividing the plane into small squares of size ϵ and determining the probability of each square with equation (6.14):

$$
\begin{aligned}
\int Y_1Y_2 d\mathbb{P} &\simeq \sum_{k\in\mathbb{Z}}\sum_{l\in\mathbb{Z}} k\varepsilon\cdot l\varepsilon\cdot \mathbb{P}(Y_1\in[\mu\varepsilon-\tfrac{1}{2}\varepsilon,\mu\varepsilon+\tfrac{1}{2}\varepsilon), Y_2\in[\kappa\varepsilon-\tfrac{1}{2}\varepsilon,\kappa\varepsilon+\tfrac{1}{2}\varepsilon)) \\
&= \sum_{k\in\mathbb{Z}}\sum_{l\in\mathbb{Z}} k\varepsilon\cdot l\varepsilon\cdot (F^{(Y_1,Y_2)}(\mu\varepsilon+\tfrac{1}{2}\varepsilon,\kappa\varepsilon+\tfrac{1}{2}\varepsilon)+F^{(Y_1,Y_2)}(\mu\varepsilon-\tfrac{1}{2}\varepsilon,\kappa\varepsilon-\tfrac{1}{2}\varepsilon) \\
&\quad - F^{(Y_1,Y_2)}(\mu\varepsilon+\tfrac{1}{2}\varepsilon,\kappa\varepsilon-\tfrac{1}{2}\varepsilon)-F^{(Y_1,Y_2)}(\mu\varepsilon-\tfrac{1}{2}\varepsilon,\kappa\varepsilon+\tfrac{1}{2}\varepsilon))
\end{aligned}
\tag{6.17}
$$

If F is continuous then the preceding procedure yields a good approximation for the covariance of Y_1 and Y_2 for sufficiently small ε. The variances of the median filter variables can, of course, be determined by applying a similar strategy in one dimension to the distribution function that is given by (6.12).

6.8 Examples

In this chapter we have introduced several robust wavelet estimators, but so far we have rarely seen how they perform in practice. Therefore we will now apply all new algorithms of this chapter to three data sets. Throughout this section Daubechies' compactly supported wavelets of extremal phase with four vanishing moments will be used.

We start with revisiting the crash data that we have already seen in Section 6.4 and to which we successfully applied algorithm 6.1. A similar result for the new wavelet basis can be seen in the top left corner of Figure 6.5. Again, the window width λ was chosen as 0.03 such that every window contained at least five data points. A similar strategy for choosing the window widths was used for the other algorithms and leads to the parameter choice $\lambda = 1.13$ and $\mu = 0.23$ for algorithm 6.4 and its simplified version and $\gamma = 0.03$ for algorithm 6.5.

The curves are obviously much less disturbed by the outliers in the crash data set. There are no high frequency phenomena and the minimum of the curve does not appear to be overestimated. However, none of the curves seems to fit the data perfectly, see for example the bump near time position 13 in the first three figures, and the interval $[12, 19]$ in the last one. Experiments showed that slight improvements are possible by adjusting the parameters manually. It is likely that further improvements are possible by using a different grid transform where some sort of local averaging takes place or by estimating the variances for each observation separately rather than globally—note that the noise level of the crash data is very small at the beginning, but increases clearly after the impact and exhibits the most variation near time point 33.

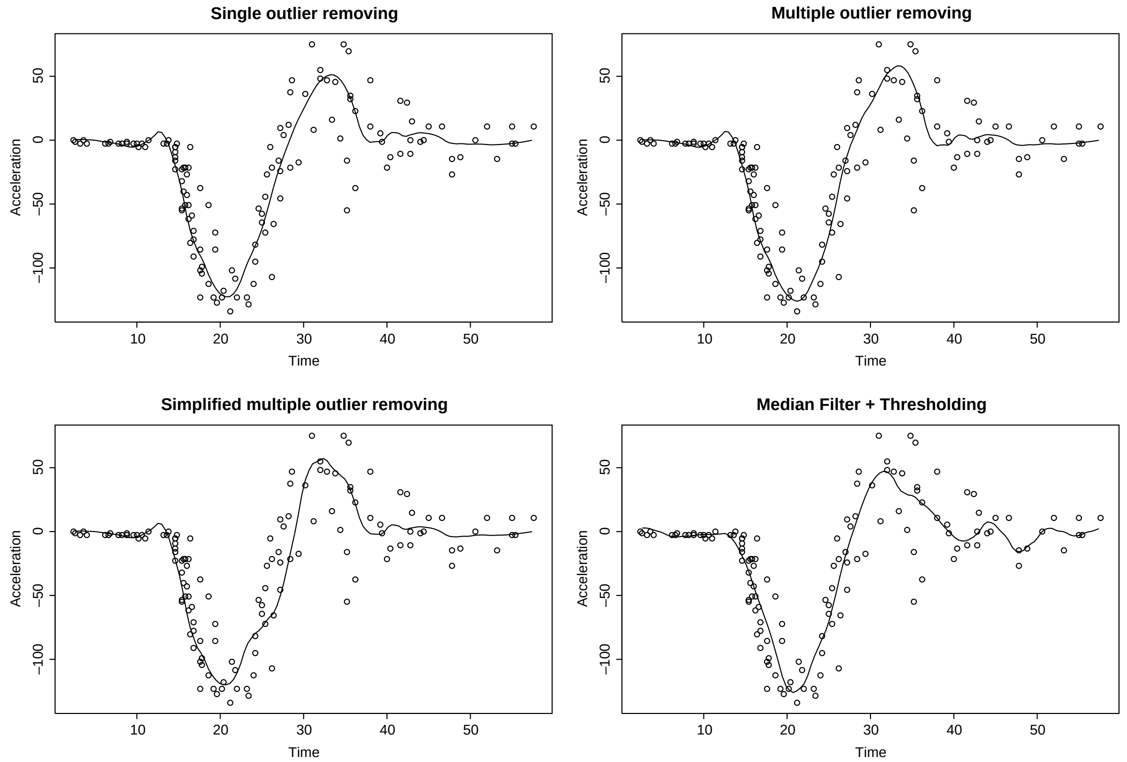

Figure 6.5: Four robust estimates for the crash data. The top left panel shows the result of wavelet thresholding after single outliers were removed with Algorithm 6.1. The wavelet estimate after multiple outlier removal with Algorithm 6.3 and its simplification can be seen in the top right and bottom left corners. Finally, the combination of median filtering with wavelet thresholding using Algorithm 6.4 and Algorithm 6.5 is shown in the bottom right figure.

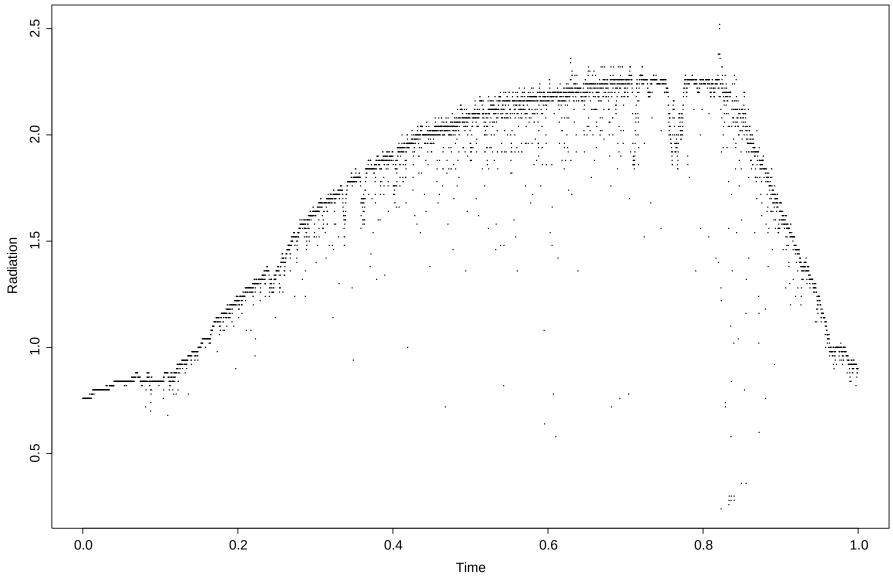

Figure 6.6: The balloon data set

Let us now turn to a different data set which is shown in Figure 6.6 and was already mentioned for several times. The data are described by Davies and Gather (see [30]) and they are taken from a balloon which took measurements of radiation from the sun. Unfortunately, it happened occasionally that the measurement device was cut off from the sun causing large outlier patches.

The data set that we analyse here is relatively short and consists of 4984 observations, but there are similar sets from other balloon flights available with more than 30000 data points. Therefore, it may be possible to use the small data set to adjust the parameters for the smoothing procedures.

We have already pointed out that the choice of the parameters largely depends on the largest outlier patch. It is not clear whether the unusual small observations around time point 0.75 are caused by the rope or actually belong to the signal. We consider here the first possibility and regard them as outliers. This means that the procedures must be able to remove 150 consecutive outliers. There is no information about the time structure available, in particular no time data set. Therefore we used here equally spaced time points on a grid of size 4984 in the unit interval, and tried to adjust the parameters such that outlier patches of size $150/4984$ could be removed by the algorithms. In particular, we chose λ for algorithm 6.4 and γ for algorithm 6.5 both as $150/4984$. Because of the spiky shape of the outlier patches the window size could be decreased for algorithm 6.4 and its simplified version where $\kappa = 200/4984$ and $\mu = 40/4984$ turned out to be large enough.

It has to be mentioned that the noise level of the data was estimated in a different way because the data are sampled with a precision of two digits and also appear to be highly correlated, both occasionally causing 70 identical consecutive observations, so that the estimation method as given by equation 6.1 does not seem to be suitable. Therefore, we estimated the noise level in two different ways and obtained approximately $\sigma = 0.06$ in both cases: The first method was to replace each difference $y_{2j} - y_{2j-1}$ in 6.1 by $y_i - y_{i-600}$. The second approach was to choose the interval $[0.6;\ 0.8]$ where the underlying signal looked nearly constant and to calculate the normalised median absolute deviation from the observations in this interval.

The estimated signals can be seen in figure 6.7. They are scarcely affected by the outlier patches, only the curve which is produced by algorithm 6.1 exhibits some variation in the interval $[0.2;\ 0.4]$ which is possibly due to the outliers. On the other hand, algorithm 6.4 and its simplified version tend to oversmooth the data slightly which can be noted in the interval $[0.6;\ 0.8]$ which contains the large outlier patch. Finally, the combination of

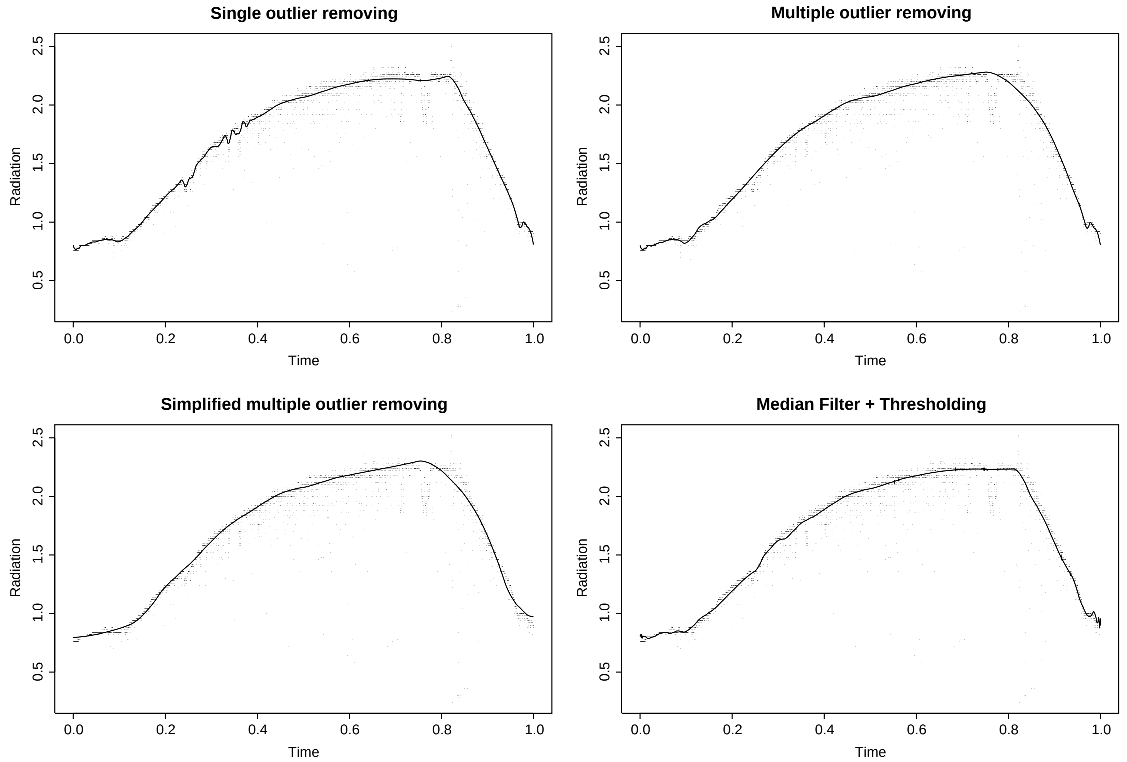

Figure 6.7: Four robust estimates for the balloon data. The top left panel shows the result of wavelet thresholding after single outliers were removed with Algorithm 6.1. The wavelet estimate after multiple outlier removal with Algorithm 6.3 and its simplification can be seen in the top right and bottom left corners. Finally, the combination of median filtering with wavelet thresholding using Algorithm 6.4 and Algorithm 6.5 is shown in the bottom right figure.

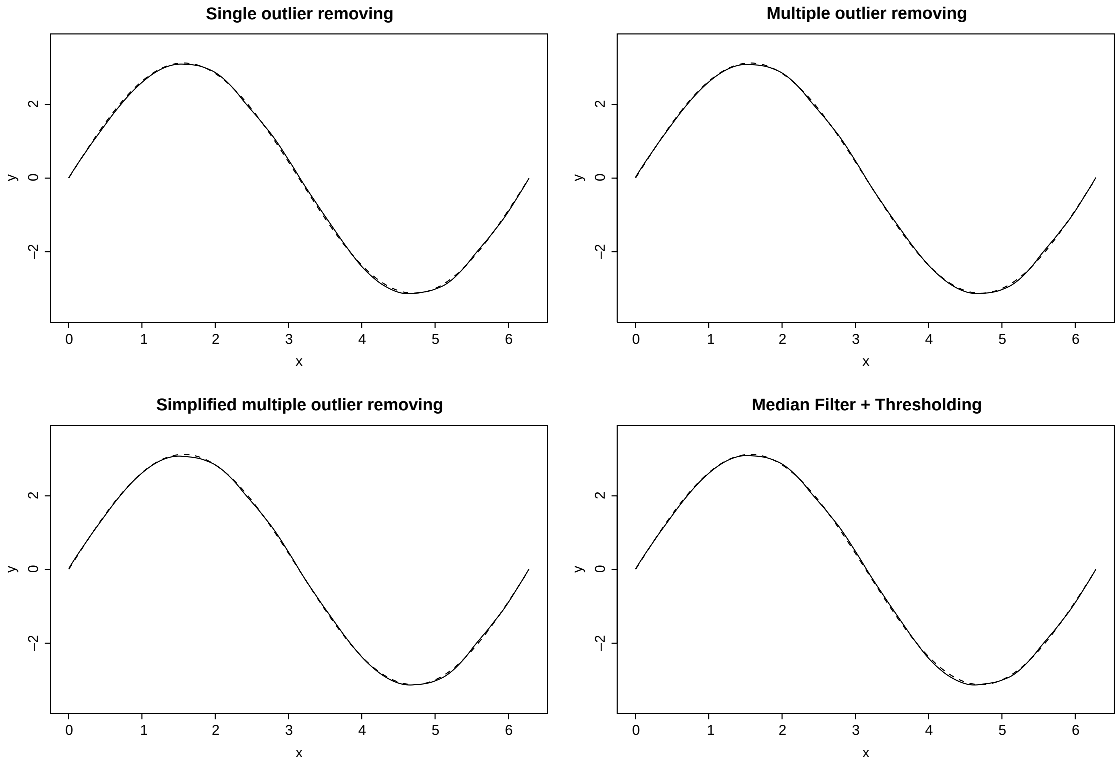

Figure 6.8: Four robust estimates for a sine curve which is disturbed by Cauchy noise. The top left panel shows the result of wavelet thresholding after single outliers were removed with Algorithm 6.1. The wavelet estimate after multiple outlier removal with Algorithm 6.3 and its simplification can be seen in the top right and bottom left corners. Finally, the combination of median filtering with wavelet thresholding using Algorithm 6.4 and Algorithm 6.5 is shown in the bottom right figure.

median filter with thresholding produces a nearly perfect fit, with exactly the right amount of smoothing.

Finally, we have a look at the sine curve which was disturbed with Cauchy noise at the beginning of this chapter. Since the noise is independently distributed there is no need to adjust for large outlier patches and small window sizes can be chosen. We chose the window sizes λ and γ for algorithms 6.1 and 6.5 both as $3.5/2048$, so that there were seven observations in each window. Equivalently, we used $\kappa = 7.5/2048$ and $\mu = 1.5/2048$ for the other algorithms. Moreover, it should be pointed out that for the determination of the thresholds for algorithm 6.5 advantage was taken from the possibility to use models for the noise distribution other than a normal distribution. In particular, Theorem 6.2 was used with F being the distribution function of a Cauchy distribution. The reconstructed sine curves can be seen in figure 6.8 as solid lines and the original signal is plotted in the same figure as a dashed line. However, differences between the estimates and the sine curve are hardly visible because all estimates are very close to the original curve.

6.9 Summary

In this chapter we have examined various ways for the robustification of wavelet thresholding which is in its usual form very sensitive to outliers. The new methods were designed to cope with unequally spaced time-points and made use of median filters which have special robustness properties that were briefly discussed. These median filters were combined with wavelet thresholding in two different ways:

- as a tool for outlier recognition

- as a prefilter

Outlier recognition was essentially performed by the computation of the differences between the data points and the median filter: Data points with a large distance to the median filter were suspected to be an outlier and therefore removed from the data set. Afterwards wavelet thresholding was applied to the cleaned data set. The simple outlier test was generalised such that outlier patches could be recognised more easily. Although this procedure is very reliable in removing consecutive outlying observations, it has the disadvantage that it removes at the same time too many data points which are not contaminated by outliers. It might be interesting to adapt other tests for multiple outliers for this situation.

The second approach that was described in this chapter used median filters in a different way. A two-stage algorithm was suggested where the data were preprocessed by a median filter and the filter output was treated by wavelet thresholding. The interesting problem was here to determine suitable thresholds since the wavelet coefficients are no longer normally distributed. A procedure for the computation of the thresholds was described which tried to remove all the noise like VisuShrink in the usual setting. Unfortunately, this method is currently based on simulation and therefore lacks the speed of other wavelet methods although only few replications are necessary. Therefore, an important problem for future research is to derive a fast algorithm for the calculation of suitable thresholds. Another interesting feature of this approach is that the model for the noise is not very specific. Therefore it might be interesting to study the behaviour of this method under several models of the noise.

The robust methods were applied to three data sets where classical thresholding does not work very well. In these examples all methods performed reasonably well. In practice the decision which method should be used depends on the given data set. If most of the data points are disturbed by Gaussian noise and only a few points are contaminated by large outliers, then outlier testing procedures should be employed. However, with many data sets it is difficult to decide whether a data point is an outlier or not. In those cases it might be better to model the noise by a distribution with heavy tails such as the Cauchy and to use the second approach where the task of the median filter is to prefilter the data.

Chapter 7

Conclusions and suggestions for future research

"The point of philosophy is to start with something

so simple as not to seem worth stating, and to end with something

so paradoxical that no one will believe it."

—Bertrand Russell, *The Philosophy of Logical Atomism*

7.1 Summary

Wavelet thresholding belongs to the more recent approaches to nonparametric regression. This method has many advantages: in particular it is of order n and adjusts automatically to the smoothness of f. Present methods require however that the time points are equally spaced and the number of points is a power of two. Furthermore, the variance of the noise is assumed to be constant and the method is sensitive to outliers and distributions with heavy tails.

Several suggestions have been made of how to overcome the limitations of $n = 2^m$, but the other problems have not been considered in much detail. In thesis we have presented a framework which permits the relaxation of all these restrictions, although we were mostly concerned with unequally spaced time points and (in Chapter 6) with the sensitivity to outliers.

The most important feature of the algorithms that were described was computational effiency: speed is one of the most striking properties of wavelet methods and should be retained in all applications and generalisations. The methods we developed are of order n, so that they can easily deal with large data sets. This was essentially achieved by the use of an efficient method for computing the variances of wavelet coefficients that are obtained by applying the DWT to correlated, non-stationary noise.

Another important idea of this thesis was to apply coefficient-dependent thresholding. Previous work has mostly focussed on situations where either the variances of all wavelet coefficients are equal or where at least the wavelet coefficients within one level have the same variance. In those cases it is common to use either one global threshold or level-dependent thresholds. Our work, however, was concerned with wavelet coefficients that could have different variances, so it was reasonable to employ an approach with coefficient-dependent thresholding, where for each wavelet coefficient, a threshold was computed which depended on the variance of the coefficient.

These two ideas were used to generalise wavelet thresholding for unequally spaced data: The given data were transformed on a grid and their wavelet decomposition was calculated. Then the variance of each coefficient was determined and coefficient-dependent thresholding was performed. Finally the inverse wavelet transform was applied to the modified coefficient vector.

The performance of this method was assessed by an analysis of the expected mean squared error. An exact formula for the risk was derived and, for example, used to calculate optimal thresholds which minimise the risk. One surprising result was that reduced VisuShrink thresholds of the form $\sqrt{2\log(n)}\sigma/3$ were nearly optimal for Donoho and Johnstone's four test signals in terms of the L^2-error.

Finally we relaxed the assumption that the noise is normally distributed and to develop methods that can cope with outliers and distributions with heavy tails. Two approaches were suggested, both based on median filters. The first method tried to recognise outliers by an evaluation of the differences between the given data and a median filter: Data points with a large distance to the local median were removed from the data set, the others kept. Subsequently, wavelet thresholding was applied to the cleaned data. This procedure is most effective for data sets where it is easy to separate data points that are contaminated by outliers from data which are only disturbed by normal noise.

Our second approach was to preprocess the given data with a median filter and to threshold the filter output afterwards. The choice of suitable thresholds turned out to be very difficult because the wavelet coefficients are no longer normally distributed even if the noise of the observations is Gaussian. A thresholding mechanism was suggested which is based on simulation and therefore not very fast, but had the advantage that the noise of the observations is not assumed to be normally distributed and can rather be specified, for example, as a Cauchy distribution.

7.2 Suggestions for future research

We finish this thesis by giving the prospects for possible future research.

Grid transformations We have already noted that other grid transformations might be useful because some of the data points are not used for the estimation procedure unless the distance between two grid points is smaller than the minimum distance between each two time points. Therefore it might turn out to be a good idea to replace the simple grid transform by some sort of local averaging. The computational speed of the algorithm could still be retained. This seems to be very promising for data sets that exhibit both large gaps and regions where data points accumulate.

The development of a different grid transform can also be motivated by the results of the risk analysis in Chapter 5 which showed that a considerable amount of error was due to the bias of the grid transform. This bias might be reduced by interpolation with higher-order polynomials.

Using covariances In our application of the algorithm we have almost entirely concentrated on the variances of the individual wavelet coefficients, while the algorithm itself also yields a great deal of information about covariance. Even though the wavelet transform often has a decorrelating effect (see [67]) it would be interesting to devise ways of processing the coefficients in a way that makes use of our knowledge of their correlation structure. This may well be more burdensome computationally, but would possibly produce more accurate statistical estimates.

Two dimensions We have concentrated on the one-dimensional case, but wavelets are of growing importance in the analysis of image data. The basic principles of our method can be easily extended to deal with two-dimensional wavelet transforms of data showing short-range correlation. Of course, the operational details are likely to depend on the specific field of application, but the need for efficient algorithms is likely to be even more crucial than in the one-dimension al case.

Cross-validation A very popular approach for choosing a smoothing parameter in non-parametric regression is cross-validation. Conventional wavelet approaches can not use this procedure directly for the threshold choice since it is not easily possible to drop a point from the data. Our new algorithm for unequally spaced data, however, does permit this and therefore it would be interesting to see how cross-validation works in this situation.

Heteroscedastic data Although we assumed throughout most parts of the thesis that the noise is uncorrelated and identically distributed the new frameworks allows for taking extra knowledge into account. For example, we mentioned already in Section 4.6 that one might estimate the noise level of each data point and take this into account when computing the variances of the wavelet coefficients.

Robust procedures We have studied a robust wavelet approach where thresholding is applied to the output of a median filter. Unfortunately the algorithm that determined suitable thresholds was based on simulation and somewhat slow. It would be very helpful if the thresholds could be found without simulation by a fast algorithm of order n. Moreover an extensive analysis of its performance for several noise models like normal distribution, Cauchy distribution, etc. would be of great interest.

Bibliography

[1] F. Abramovich and Y. Benjamini. Thresholding of wavelet coefficients as multiple hypotheses testing Procedure. In A. Antoniadis and G. Oppenheim, editors, *Wavelets and Statistics*, pages 6–14. Springer, New York, 1995.

[2] F. Abramovich and Y. Benjamini. Adaptive thresholding of wavelet coefficients. *Computational Statistics & Data Analysis*, 22:351–361, 1996.

[3] F. Abramovich, T. Sapatinas, and B.W. Silverman. Wavelet thresholding via a Bayesian approach. Technical report, Department of Mathematics, University of Bristol, 1996.

[4] F. Abramovich and B. W. Silverman. Wavelet decomposition approaches to statistical inverse problems. *Biometrika*, 84, 1995. (to appear).

[5] D. F. Andrews, P. J. Bickel, F. R. Hampel, P. J. Huber, J. W. Tukey, and W. H. Rogers. *Robust estimates of location: survey and advances*. Princeton University Press, Princeton, New Jersey, 1972.

[6] A. Antoniadis and D. T. Pham. Wavelet regression for random or irregular design. Technical report, Laboratory of Modelling and Computation, IMAG, B.P. 53, 38041 Grenoble cedex 09, France, 1997.

[7] T.C. Bailey, T. Sapatinas, K.J. Powell, and W.J. Krzanowski. Signal detection in underwater sounds using wavelets. *Journal of the American Statistical Association*, 93, 1998. (to appear).

[8] V. Barnett and T. Lewis. *Outliers in Statistical Data*. Wiley, Chichester, 3rd edition, 1972.

[9] O. Bertrand, J. Bohorquez, and J. Pernier. Time-frequency digital filtering based on an invertible wavelet transform: an application to evoked potentials. *IEEE Transactions on Biomedical Engineering*, 41:77–88, 1994.

[10] C. Blatter. *Analysis 2*. Springer, New York, 1991.

[11] C. G. Boncelet. Algorithms to compute order statistic distributions. *SIAM Journal on Scientific and Statistical Computing*, 8:868–876, 1987.

[12] C. G. Boncelet. Order statistic distributions with multiple windows. *IEEE Transactions on Information Theory*, 37:436–442, 1991.

[13] J. Bradley, C. Brislawn, and T. Hopper. The FBI wavelet/scalar quantization standard for gray-scale fingerprint image compression. Technical Report LA-UR-93-1659, Los Alamos National Laboratory, Los Alamos, 1993.

[14] N. D. Brinkman. Ethanol—A single-cylinder engine study of efficiency and exhaust emissions. *SAE Transactions*, 90:1410–1424, 1981.

[15] A. Bruce, D. L. Donoho, H.-Y. Gao, and R. Martin. Smoothing and robust wavelet analysis. In *Proceedings CompStat*, Vienna, Austria, 1994.

[16] A. Bruce and H.-Y. Gao. S+Wavelets: an object-oriented toolkit for wavelet analysis. Technical Report 38, Statsci Division, MathSoft, Seattle, 1995.

[17] J. B. Buckheit and D. L. Donoho. WaveLab and reproducible research. In A. Antoniadis and G. Oppenheim, editors, *Wavelets and Statistics*, pages 55–81. Springer, New York, 1995.

[18] V. J. Carey, E. E. Walters, C. G. Wager, and B. A. Rosner. Resistant and test-based outlier rejection: Effects on Gaussian one-and two-sample inference. *Technometrics*, 39:320–330, 1997.

[19] D. Chaudhuri, B. B. Chaudhuri, and C. A. Murthy. A data-driven procedure for density-estimation with some applications. *Pattern Recognition*, 29:1719–1736, 1996.

[20] W. C. Chen, H. Cui, and Y. Z. Liang. A new principal component analysis method based on robust diagnosis. *Analytical Letters*, 29:1647–1667, 1996.

[21] H. A. Chipman, E. D. Kolaczyk, and R. E. McCulloch. Adaptive Bayesian wavelet shrinkage. *Journal of the American Statistical Association*, 92, 1997. (to appear).

[22] C. K. Chui. *An Introduction to Wavelets*. Academic Press, Inc., San Diego, 1992.

[23] L. P. Clarke, M. Kallegri, W. Qian, H. D. Li, R. A. Clark, and M. L. Silbiger. Tree-structured nonlinear filter and wavelet transform for microcalcification segmentation in digital mammography. *Cancer Letters*, 77:173–181, 1994.

[24] W. S. Cleveland, E. Grosse, and W. M. Shyu. Local regression models. In J. M. Chambers and T. J. Hastie, editors, *Statistical Models in S*, pages 309–376. Wadsworth and Brooks, Pacific Grove, California, 1992.

[25] M. Clyde, G. Parmigiani, and B. Vidakovic. Multiple Shrinkage and subset selection in wavelets. *Biometrika*, 84, 1997. (to appear).

[26] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41:909–996, 1988.

[27] I. Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, 1992.

[28] H. A. David. *Order Statistics*. Wiley, New York, 1981.

[29] H. A. David and V. S. Shu. Robustness of location estimators in the presence of an outlier. In H. A. David, editor, *Contributions to Survey Sampling and Applied Statistics: Paper in Honor of H. O. Hartley*, pages 235–250. Academic Press, New York, 1978.

[30] L. Davies and U. Gather. The identification of multiple outliers (with discussion). *Journal of the American Statistical Association*, 88:782–801, 1993.

[31] P. L. Davies. Data features. *Statistica Neerlandica*, 49:185–245, 1995.

[32] B. Delyon and A. Juditsky. Estimating wavelet coefficients. In A. Antoniadis and G. Oppenheim, editors, *Wavelets and Statistics*, pages 151–167. Springer, New York, 1995.

[33] D. L. Donoho. Interpolating wavelet transforms. Technical Report 408, Department of Statistics, Stanford University, 1992.

[34] D. L. Donoho. Nonlinear wavelet methods for recovery of signals, images, and densities from noisy and incomplete data. In I. Daubechies, editor, *Different Perspectives on Wavelets*, pages 173–205. American Mathematical Society, 1993.

[35] D. L. Donoho. Wavelet shrinkage and W.V.D. – a ten-minute tour. In Y. Meyer and S. Roques, editors, *Progress in Wavelet Analysis and Applications*, pages 55–81. Éditions Frontières, Gif-sur-Yvette, 1993.

[36] D. L. Donoho. De-noising via soft-thresholding. *IEEE Transactions on Information Theory*, 41:613–627, 1995.

[37] D. L. Donoho and P. J. Huber. The notion of breakdown point. In P. J. Bickel, K. A. Doksum, and J. L. Hodges, Jr., editors, *A festschrift for Erich L. Lehmann in honor of his 65th birthday*, pages 157–184. Wadsworth, Belmont, California, 1983.

[38] D. L. Donoho and I. M. Johnstone. Neo-classical minimax theorems, thresholding, and adaptation. Technical report, Department of Statistics, Stanford University, 1992.

[39] D. L. Donoho and I. M. Johnstone. Ideal spatial adaption by wavelet shrinkage. *Biometrika*, 81:425–455, 1994.

[40] D. L. Donoho and I. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90:1200–1224, 1995.

[41] D. L. Donoho, I. M. Johnstone, G. Kerkyacharian, and D. Picard. Wavelet shrinkage: asymptopia? *Journal of the Royal Statistical Society*, Series B, 57:301–370, 1995.

[42] R. M. Dudley. *Real Analysis and Probability*. Chapman & Hall, New York, 1989.

[43] R. L. Eubank. *Spline Smoothing and Nonparametric Regression*. Marcel Dekker Inc., New York, 1988.

[44] J. Fan and I. Gijbels. Variable bandwidth and local linear regression smoothers. *Annals of Statistics*, 20:2008–2036, 1992.

[45] J. Fan and I. Gijbels. Data-driven bandwidth selection in local polynomial fitting: variable bandwidth and spatial adaption. *Journal of the Royal Statistical Society*, Series B, 57:371–394, 1995.

[46] J. Fan and I. Gijbels. *Local Polynomial Modelling and Its Applications*. Chapman and Hall, London, 1996.

[47] N. C. Gallagher, Jr. and G. L. Wise. A theoretical analysis of the properties of median filters. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29:1136–1141, 1981.

[48] P. J. Green and B. W. Silverman. *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*. Chapman and Hall, London, 1994.

[49] A. Haar. Zur Theorie der orthogonalen Funktionensysteme. *Math. Ann.*, 69:331–371, 1910.

[50] P. Hall and G. Nason. On choosing a non-integer resolution level when using wavelet methods. *Statistics & Probability Letters*, 34:5–11, 1997.

[51] P. Hall and P. Patil. On the choice of smoothing parameter, threshold and truncation in nonparametric regression by nonlinear wavelet methods. *Journal of the Royal Statistical Society*, Series B, 58:361–377, 1996.

[52] F. R. Hampel. *Contributions to the theory of robust estimation*. PhD thesis, University of California, Berkeley, 1968.

[53] F. R. Hampel. A general qualitative definition of robustness. *Annals of Mathematical Statistics*, 42:1887–1896, 1971.

[54] F. R. Hampel. The breakdown point of the mean combined with some rejection rules. *Technometrics*, 27:95–107, 1985.

[55] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: The Approach based on Influence Functions*. Wiley, New York, 1986.

[56] W. Härdle. *Applied Nonparametric Regression*. Cambridge University Press, Cambridge, 1994.

[57] T. J. Hastie. Generalized Additive Models. In J. M. Chambers and T. J. Hastie, editors, *Statistical Models in S*, pages 195–247. Wadsworth and Brooks, Pacific Grove, California, 1992.

[58] T. J. Hastie and R. Tibshirani. *Generalized Additive Models*. Chapman & Hall, London, 1987.

[59] D. M. Hawkins. *The identification of outliers*. Chapman and Hall, New York, 1980.

[60] J. L. Hodges, Jr. Efficiency in normal samples and tolerance of extreme values for some estimates of location. In L. M. Le Cam and J. Neyman, editors, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 163–168, Berkeley, 1967. University of California Press.

[61] B. B. Hubbard. *The World According to Wavelets: The Story of a Mathematical Technique in the Making*. A K Peters, Wellesley, 1996.

[62] P. J. Huber. *Robust Statistics*. Wiley, New York, 1981.

[63] R. Ihaka and R. Gentleman. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5:299–314, 1996.

[64] M. Jansen and A. Bultheel. Multiple wavelet threshold estimation by generalized cross validation for data with correlated noise. Technical report, Department of Computer Science, Katholieke Universiteit Leuven, Belgium, 1996.

[65] M. Jansen, M. Malfait, and A. Bultheel. A generalized cross validation for wavelet thresholding. *Signal Processing*, 56:33–44, 1997.

[66] B. Jawerth and W. Sweldens. An overview of wavelet based multiresolution analyses. *SIAM Review*, 36:377–412, 1993.

[67] I. M. Johnstone and B. W. Silverman. Wavelet threshold estimators for data with correlated noise. *Journal of the Royal Statistical Society*, Series B, 59:319–351, 1997.

[68] M. Juhola, J. Katajainen, and T. Raita. Comparison of algorithms for standard median filtering. *IEEE Transactions on Signal Processing*, 39:204–208, 1991.

[69] B. I. Justusson. Median filtering: statistical properties. In T. S. Huang, editor, *Two-Dimensional Digital Signal Processing (II)*, volume 43 of *Topics in Applied Physics*, pages 161–196. Springer, Berlin, 1981.

[70] G. Kaiser. *A Friendly Guide to Wavelets*. Birkhäuser, Boston, 1994.

[71] A. Kovac. *Robuste nicht-parametrische Regression mit Wavelets*. Master's thesis, Universität GH Essen, Essen, 1996.

[72] F. Kuhlmann and G. L. Wise. On second moment properties of median filtered sequenes of independent data. *IEEE Transactions on Communications*, 29:1374–1380, 1981.

[73] M. K. Kwong and P. T. P. Tang. W-matrices, nonorthogonal multiresolution analysis, and finite signals of arbitrary length. Technical Report MCS-P449-0794, Argonne National Laboratory, 1994.

[74] M. R. Leadbetter, G. Lindgren, and H. Rootzen. *Extremes and Related Properties of Random Sequences and Processes*. Springer, New York, 1983.

[75] L. Lenarduzzi. Denoising not equispaced data with wavelets. Technical Report IAMI 97.1, Istituto Applicazioni Matematica ed Informatica C.N.R., via Ampere 56, 20131 Milano, Italy, 1997.

[76] T. P. Lesbas, D. C. Mason, and N. C. Millard. TOBI image-processing – the state of the art. *IEEE Journal of Oceanic Engineering*, 20:85–93, 1995.

[77] S. G. Mallat. Multiresolution approximations and wavelet orthonormal bases of $L^2(\mathbb{R})$. *Transactions of the American Mathematical Society*, 315:69–89, 1989.

[78] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:674–693, 1989.

[79] L. Metzner. *Facettierte nichtparametrische Regression*. PhD thesis, Department of Mathematics, University of Essen, Essen, Germany, 1997.

[80] Y. Meyer. Principle of uncertainty, Hilbert bases and algebras of operators. *Astérisque*, 145:209–223, 1987.

[81] Y. Meyer. *Wavelets and Operators*. Cambridge University Press, Cambridge, 1992.

[82] E. A. Nadaraya. On estimating regression. *Theory of Probability and its Applications*, 10:186–190, 1964.

[83] G. P. Nason. Choice of the threshold parameter in wavelet function estimation. In A. Antoniadis and G. Oppenheim, editors, *Wavelets and Statistics*, pages 261–280. Springer, New York, 1995.

[84] G. P. Nason. Wavelet function estimation using cross-validation. *Journal of the Royal Statistical Society*, Series B, 58:463–479, 1996.

[85] G. P. Nason and B. W. Silverman. The discrete wavelet transform in S. *Journal of Computational and Graphical Statistics*, 3:163–191, 1994.

[86] T. A. Nodes and N. C. Gallagher, Jr. The output distribution of median type filters. *IEEE Transactions on Communications*, 32:532–541, 1984.

[87] T. Ogden and E. Parzen. Change-point approach to data analytic wavelet thresholding. *Statistics & Computing*, 6:93–99, 1996.

[88] T. Ogden and E. Parzen. Data-dependent wavelet thresholding in nonparametric regression. *Computational Statistics & Data Analysis*, 22:53–70, 1996.

[89] S. Peterson. *Filtering and Wavelet Regression Methods with Application to Exercise ECG*. PhD thesis, Lund University, Sweden, 1996.

[90] K.J. Powell, T. Sapatinas, T.C. Bailey, and W.J. Krzanowski. Application of wavelets to the pre-processing of underwater sounds. *Statistics & Computing*, 5:265–273, 1995.

[91] W. H. Press. Wavelet Transforms. Preprint 3184, Harvard-Smithsonian Center for Astrophysics, 1991.

[92] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C, the Art of Scientific Computing*. Cambridge University Press, Cambridge, 1992.

[93] P. J. Rousseuw. Least median of squares regression. *Journal of the American Statistical Association*, 79:871–888, 1984.

[94] P. J. Rousseuw and A. M. Leroy. *Robust regression and outlier detection*. Wiley, New York, 1987.

[95] W. Rudin. *Functional Analysis*. McGraw, New York, 1973.

[96] S. Sakata and H. White. An alternative definition of finite-sample breakdown point with applications to regression-model estimators. *Journal of the American Statistical Association*, 90:1099–1106, 1995.

[97] B. W. Silverman. Some aspects of the spline smoothing approach to non-parametric regression curve fitting (with discussion). *Journal of the Royal Statistical Society*, Series B, 47:1–52, 1985.

[98] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, 1986.

[99] J. S. Simonoff. *Smoothing Methods in Statisics*. Springer, New York, 1996.

[100] S. S. Sinha and D. G. Schunck. A 2-stage algorithm for discontinuing-preserving surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:36–55, 1992.

[101] M. J. T. Smith and T. P. Barnwell. Exact reconstruction techniques for tree structured subband coders. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34:434–441, 1986.

[102] M. J. T. Smith and S. L. Eddins. Analysis/synthesis techniques for subband image coding. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38:1446–1456, 1990.

[103] J. K. Song and Y. H. Lee. Linear combinations of weighted order statistic filters – canonical structure and optimal-design. *IEEE Transactions on Circuits and Systems, II Analog and Digital Signal Processing*, 43:349–362, 1996.

[104] C. Stein. Estimation of the mean of a multivariate normal distribution. *Annals of Statistics*, 9:1135–1151, 1981.

[105] C. J. Stone. Consistent nonparametric regression. *Annals of Statistics*, 5:595–645, 1977.

[106] G. Strang. Wavelet transforms versus Fourier transforms. *Bulletin (New Series) of the American Mathematical Society*, 28:288–305, 1995.

[107] A. J. Stromberg and D. Ruppert. Breakdown in nonlinear regression. *Journal of the American Statistical Association*, 87:991–997, 1992.

[108] J. O. Strömberg. A modified Franklin system and higher order spline systems on $\mathbb{R}^n$ as unconditional bases for Hardy spaces. In W. Beckner and A. P. Calderon, editors, *Conference On Harmonic Analysis In Honor Of Antoni Zygmund, University Of Chicago, March 23 - 28, 1981*, volume 2, pages 475–493, Belmont, California, 1982. Wadsworth.

[109] A. Sudeshna. *Time-dependent Spectral Analysis of Nonstationary Time Series*. PhD thesis, Stanford University, 1996.

[110] W. Sweldens. *The construction and application of wavelets in numerical analysis*. PhD thesis, Katholieke Universiteit Leuven, Leuven, 1995.

[111] W. Sweldens and R. Piessens. Wavelet sampling techniques. In *1993 Proceedings of the Statistical Computing Section*, pages 20–29. American Statistical Association, 1993.

[112] W. Sweldens and R. Piessens. Quadrature formulae and asymptotic error expansions for wavelet approximations of smooth functions. *SIAM Journal of Numerical Analysis*, 31:1240–1264, 1994.

[113] C. Taswell and K. C. McGill. Wavelet transform algorithms for finite duration discrete-time signals. *ACM Transactions on Mathematical Software*, 20:398–412, 1994.

[114] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, Reading, Massachussets, 1977.

[115] J. W. Tukey. Nonlinear (nonsuperposable) methods for smoothing data. In D. R. Brillinger, editor, *The Collected Works of John W. Tukey*, volume II. Wadsworth, Monterey, California, 1984.

[116] S. G. Tyan. Median filtering: deterministic properties. In T. S. Huang, editor, *Two-Dimensional Digital Signal Processing (II)*, volume 43 of *Topics in Applied Physics*, pages 197–217. Springer, Berlin, 1981.

[117] P. F. Velleman. Definition and comparison of robust nonlinear data smoothing algorithms). *Journal of the American Statistical Association*, 75:609–615, 1980.

[118] B. Vidakovic. Nonlinear wavelet shrinkage with Bayes rules and Bayes factors. *Journal of the American Statistical Association*, 92, 1997. (to appear).

[119] M. P. Wand and M. C. Jones. *Kernel Smoothing*. Chapman and Hall, London, 1995.

[120] Y. Wang. Function estimation via wavelet shrinkage for long-memory data. *Annals of Statistics*, 24:466–484, 1996.

[121] G. S. Watson. Smooth regression analysis. *Sankhyā*, Series A, 26:101–116, 1964.

[122] S. Weisberg. *Applied Linear Regression*, pages 231,234. Wiley, New York, 2nd edition, 1985.

[123] N. Weyrich and G. T. Warhola. De-noising using wavelets and cross-validation. In S.P. Singh, editor, *Approximation Theory, Wavelets and Applications*, volume 454 of *NATO ASI SERIES C, Mathematical and Physical Sciences*, pages 523–532. Kluwer, Dordrecht, 1995.

[124] V. Wickerhauser. Acoustic signal compression with wavelet packets. In C.K. Chui, editor, *Wavelets: a Tutorial in Theory and Applications*. Academic Press, Boston, 1992.

[125] V. Wickerhauser. *Adapted Wavelet Analysis from Theory to Software*. AK Peters, Boston, 1994.

[126] L. Yin, R. K. Yang, M. Gabbouj, and Y. Neuvo. Weighted median filters – a tutorial. *IEEE Transactions on Circuits and Systems, II Analog and Digital Signal Processing*, 43:157–192, 1996.