A SECOND GENERATION WAVELET CONSTRUCTION AND APPLICATIONS TO REGRESSION AND TIME SERIES ANALYSIS

Marina Iuliana Knight

School of Mathematics

March 2006

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

Abstract

This thesis is centred on second generation wavelet constructions, and investigates some of their possible applications to regression and time series analysis. We start by proposing an adaptive lifting scheme which constructs second generation wavelets that adapt locally to the signal features. We demonstrate empirically that our proposed algorithms provide sparse wavelet representations and have competitive denoising properties for irregularly spaced datasets, when compared to both established wavelet and non-wavelet based regression techniques.

Next we address the problem of transmembrane segment prediction along a protein sequence, which we 'convert' into a regression problem. Central to our approach is the construction of inter-residue distance matrices, used for estimating the protein's hydropathy profile. In conjunction with this, we propose using two adaptive algorithms for denoising the hydropathy signal. Our approach is shown to improve prediction of transmembranar segments when compared to results obtained using classical wavelets.

Finally, we investigate the problem of estimating the evolutionary wavelet spectrum of a locally stationary wavelet process whose realizations are assumed to feature missing observations. In order to obtain a set of detail coefficients associated with each observed time location, we propose a modified lifting scheme. Based on this, we construct a second generation wavelet estimator of the wavelet spectrum and investigate some of its properties, both theoretically and computationally. The obtained results indicate that our proposed periodogram needs a bias correction, which is a direction for further research.

Acknowledgements

All my thanks go to my advisor, Guy Nason, for his guidance, patience and most of all optimism throughout these last three years. I also want thank Guy Nason and Bernard Silverman for taking me on, and having trusted me.

Matt Nunes made this journey more enjoyable, through his friendship and work collaboration, and I also feel particularly lucky for having good friends rather than office colleagues, Chris, Simon, Huw, Martin and Geoff.

Finally, I want to thank my mum and my husband for much more than words can say.

Thank you all!

Author's Declaration

I declare that the work in this thesis was carried out in accordance with the Regulations of the University of Bristol. The work is original except where indicated by special reference in the text and no part of the dissertation has been submitted for any other degree. Any views expressed in the dissertation are those of the author and do not necessarily represent those of the University of Bristol. The thesis has not been presented to any other university for examination either in the United Kingdom or overseas.

Marina Iuliana Knight

Date: 13th March 2006

Cu toată dragostea, pentru tine Olivia

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Simply described, wavelets are localised functions that resemble a 'small wave', as their name also suggests. Due to their ability of providing multiscale representations of objects (of the type zoom in–zoom out) and to their good compression properties, wavelets are found in applications in various fields from signal processing and time series to proteomics or genetics.

Classical, or first generation wavelets as they are also known, suffer from some limitations: usually they can only work on data that is regularly spaced and of dyadic length. In this thesis we will concentrate on second generation wavelet constructions based on the lifting scheme that removes one coefficient at a time of Jansen *et al.* (2001, 2004), which allow for greater generality in applications: second generation wavelets can work on irregularly spaced data, irrespective of their length.

This thesis introduces a novel adaptive construction of second generation wavelets, whose properties are investigated in the context of nonparametric regression. A problem that appears in proteomics is then addressed, and a solution that involves a combination between a new approach for modelling the hydropathy profile of a protein and the adaptive wavelet construction previously investigated is proposed. Finally, we make use of second generation wavelets for spectral estimation in the context of a locally stationary wavelet time series that features missing observations.

Chapter 2 reviews the current literature on both first and second generation wavelets, as well as the use of wavelets in nonparametric regression problems. Since chapter 5 is inter-disciplinary, it starts with a short introduction to proteins tailored for the needs of the problem addressed there, that of transmembrane segment prediction. Chapter 6 is concerned with the problem of spectral estimation from a realization with missing observations of a process belonging to a certain class of nonstationary processes introduced by Nason *et al.* (2000). Therefore, a small literature review on (locally stationary) time series appears in its beginning.

Successful noise removal and function estimation in nonparametric regression problems are fundamentally dependent on choosing a wavelet basis of appropriate smoothness. The work of chapters 3 and 4 is motivated by the interesting problem of trying to overcome the user-directed choice of a wavelet basis in nonparametric regression problems. In chapter 3 we propose an adaptive second generation wavelet approach that allows the circumvention of this problem. In our construction, wavelet functions are built such that they adjust locally to the signal smoothness. Taking such an approach relieves the user from having to make a subjective choice, which should be based on information such as signal smoothness that is not normally available. Our method also naturally allows for handling datasets with multiple observations at the same location. The detailed simulation study of chapter 4 shows that our adaptive methods perform well when compared to established wavelet and non-wavelet regression techniques also designed to work on irregular data.

In chapter 5 we address the problem of predicting hydrophobic segments along the sequence of a transmembrane protein. This is done by estimating the hydropathy level along a protein as a function of the protein's amino acid composition and shape. Classical wavelet-based methods have already been used for this problem, and the residues modelled as equally spaced. Our approach is based on two main ingredients: (i) we model the protein as a (straight) chain with irregularly spaced residues and (ii) for analysing its associated hydropa-

thy signal, we employ two of the adaptive algorithms from chapters 3 and 4. In order to estimate the distance between consecutive residues we derive family-oriented dissimilarity matrices that use the resolved three-dimensional information contained in similar proteins. We show that by incorporating the information contained in similar proteins and introducing irregular amino acid distances the prediction of transmembranar segments is improved, both in terms of predicted segments compared to experimentally determined ones, and also the proportion of correctly predicted segments.

In chapter 6 we investigate the problem of estimating the evolutionary wavelet spectrum associated with a class of locally stationary wavelet processes, whose realizations are assumed to feature missing observations. To estimate the corresponding spectrum, we propose a second generation wavelet-based periodogram. Its construction is based on first devising a 'nondecimated' lifting scheme which ensures that a set of empirical wavelet coefficients is available at each observed time point. We then theoretically investigate some statistical properties of the proposed periodogram, and make a first step towards constructing a corrected wavelet periodogram, that would unbiasedly estimate the spectrum. We also explore our proposed method computationally, and the obtained results are promising.

The datasets used throughout the thesis can be found at

`http://www.stats.bris.ac.uk/~maxmp/`.

Chapter 2

An Overview of Wavelet Theory

Wavelet theory is a relatively new mathematical tool which has undergone extensive research in the last 20 years. From the 1980's when wavelets were first introduced, they have emerged as strong tools, useful in a variety of applications from signal processing and data compression to astronomy. Wavelets can act not only on uni-dimensional objects, such as signals, but also on multidimensional ones, such as images and even n-dimensional functions.

In this thesis we will only be concerned with the use of wavelets in statistical problems, more exactly in signal denoising, compression and spectral estimation. Also, all applications will involve one-dimensional functions, although they can be adapted for higher dimensions.

But what are wavelets?

There are many ways of introducing the notion of wavelet, and the reader interested in a detailed mathematical presentation of wavelets should refer to the excellent monograph of Daubechies (1992), while Vidakovic (1999) provides a comprehensive review of the subject.

Since wavelets are functions themselves, and since most commonly they are used to represent functions, in this thesis we start by introducing nicely behaved spaces of functions, Hilbert spaces, although not every Hilbert space is a space of functions. Then we will formally introduce wavelets.

2.1 Bases on Hilbert spaces

In its most general sense, a Hilbert space H is a complete vector space endowed with a scalar product, usually denoted by $<\cdot\,,\cdot>$ (Rudin (1973)). Every scalar product induces a norm, denoted by $\|\cdot\|$, on the space. Completeness means that every Cauchy sequence in H converges in norm to an element of H, the scalar product provides a measure of association between two elements of H and the norm gives a measure of size of an element. As such, we can think of Hilbert spaces as being a generalisation of Euclidian spaces, with whose geometry we are familiar.

Hilbert spaces are most interesting when they are of infinite dimension, and such situations commonly occur when their elements are functions or infinite sequences.

Two common examples of infinite dimension Hilbert spaces are:

The space of square integrable functions, usually denoted by L^2.

For functions $f : \mathbb{R} \to \mathbb{C}$ we say that $f \in L^2(\mathbb{R}) \quad \Leftrightarrow \quad \int_{-\infty}^{\infty} |f(x)|^2 dx < \infty$.

The inner product defined on this space is given by $< f, g >_{L^2} = \int_{-\infty}^{\infty} f(x)\overline{g}(x)dx$, so the induced norm is $\|f\|_{L^2} = (\int_{-\infty}^{\infty} |f(x)|^2 dx)^{1/2}$.

The space of square summable sequences, usually denoted by l^2.

For sequences $\underline{x} = \{x_k\}_{k\in\mathbb{Z}}$ we say that $\underline{x} \in l^2(\mathbb{Z}) \quad \Leftrightarrow \quad \sum_{k\in\mathbb{Z}} |x_k|^2 < \infty$. The inner product defined on this space is $< \underline{x}, \underline{y} >_{l^2} = \sum_{k\in\mathbb{Z}} x_k \overline{y}_k$, and the associated norm is $\|x\|_{l^2} = (\sum_{k\in\mathbb{Z}} |x_k|^2)^{1/2}$.

Convergence in a Hilbert space is to be understood as convergence in the associated norm.

An essential property for function representation is that any (separable) Hilbert space admits a (countable) orthonormal basis.

Definition 2.1.1. *We say that a family $\{f_k\}_{k\in B} \subset H$ is an orthonormal basis for the Hilbert space H if $< f_i, f_j >= \delta_{i,j}$, $\forall i, j \in B$ and $\overline{span}\{f_k / \, k \in B\} = H$, i.e. the family $\{f_k\}_{k\in B} \subset H$ has orthogonal elements of norm 1 and is dense in H.*

Therefore any element $f \in H$ can be written as $f = \sum_{k \in B} < f, f_k > f_k$.

It would be instructive at this point to introduce one of the most widely used orthonormal bases in a Hilbert space, from which the famous Fourier series derive.

2.1.1 Example of an orthonormal basis

Let us take the space of square integrable functions defined on $[-\pi, \pi)$, $L^2([-\pi, \pi))$. It can be checked that an orthonormal basis for this space is given by the family of functions $\{f_n(\cdot)\}_{n \in \mathbb{Z}}$, where

$$f_n(x) = \frac{1}{\sqrt{2\pi}} e^{inx}, \text{ for } x \in [-\pi, \pi). \tag{2.1}$$

Consequently, any $f \in L^2([-\pi, \pi))$ can be written as the limit of a superposition of sinusoidal functions $\{f_n(\cdot)\}_{n \in \mathbb{Z}}$, i.e. $f(x) = \sum_{n=-\infty}^{\infty} < f, f_n > f_n(x)$, with $< f, f_n >= \int_{-\pi}^{\pi} \frac{1}{\sqrt{2\pi}} f(x) e^{-inx} dx$, and the convergence of the series takes place in the L^2-norm.

It follows that f can be decomposed as

$$f(x) = \sum_{n=-\infty}^{\infty} F_n e^{inx}, \tag{2.2}$$

where $F_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-inx} dx$. This result is known as the *Fourier series decomposition of* f.

The magnitude of the coefficients $\{F_n\}_n$ reveals the frequency content of the signal f. However, due to the nature of the decomposing blocks, which span the whole real axis, we are unable to (directly) extract localised information, linking the frequency to the location on the x-axis.

Fourier series are often used in practice because of feasibility of computing the coefficients F_n, but ignoring the conditions under which the series converges. For example, it can be proved that any periodic function (say g) with period $2T$, absolutely integrable over the interval $(-T, T)$ (i.e.

$\int_{-T}^{T} |g(x)|dx < \infty$) and with a finite number of discontinuities and a finite number of extremes in the interval $(-T, T)$, has a Fourier series which converges pointwise to g, except at the points of discontinuity (where the Fourier series will converge to the average of the left and right limit). For functions with discontinuities, this result leads to what in the literature is known under the name of 'Gibbs phenomenon'— a ringing in the Fourier series approximation at the points at which the signal has sudden jumps. Hence many basis functions are needed for representing a discontinuity more accurately.

The decomposition of a periodic function into its corresponding Fourier series can be further extended to non-periodic absolutely integrable functions, $f \in L^1(\mathbb{R})$ (i.e. $\int_{-\infty}^{\infty} |f(x)|dx < \infty$), which under certain assumptions can be represented at any continuity point, say x, as

$$f(x) = \int_{-\infty}^{\infty} \frac{1}{2\pi} \hat{f}(\omega) e^{i\omega x} d\omega, \tag{2.3}$$

where $\hat{f}(\omega) =< f(\cdot), e^{i\omega \cdot} >_{L^2} = \int_{-\infty}^{\infty} f(x) e^{-i\omega x} dx$ is called the *Fourier transform of* f.

The above formula shows how f can be represented at all its continuity points as a 'sum' of sinusoids with complex coefficients describing the amplitude associated to each frequency ω. Note that unlike in the Fourier decomposition of a periodic function, the frequency now varies on a continuous, rather than discrete scale.

Here we have only introduced aspects of the Fourier transform and of the Fourier series tailored for the needs of this thesis. For more details, Priestley (1981) offers a beautiful discussion of the subject.

2.1.2 Other basis constructions

When approximating a function by a superposition of basis functions it is desirable for the decomposition to use few basis functions (for example, if many coefficients F_n in equation (2.2) are null). This would facilitate a bet-

ter understanding of the signal structure, and it would also ensure a sparse representation of the signal on the chosen basis, hence better compression. Such representations can be achieved if the basis functions are mirroring as much as possible the properties of the function to be decomposed, such as its smoothness degree and periodicity structure. Due to the nature of sinusoids, the Fourier series approximation should be used for smooth functions, which also exhibit a degree of periodicity.

Most of the time, the functions we deal with in practice do not have the same degree of smoothness throughout their trajectory and may exhibit sudden jumps. Such signals cannot be successfully decomposed through a Fourier approach, and hence other bases have been investigated. We are only briefly mentioning here alternative approaches, and then we concentrate on introducing bases of wavelet functions.

Polynomial bases— the elements of such bases are monomials of various degrees, $\{(x-a)^k\}_{k\in\mathbb{N}}$— have been investigated and widely used. However, they are also poor in representing local features of the signal, and usually present problems in providing a good fit in the tails of the data (Ramsay and Silverman (1997)).

Spline bases have been constructed in the quest of obtaining adaptiveness to the local characteristics of the signal. Put simply, a spline is formed of more polynomials of the same degree, which are smoothly joined at points called 'knots'. One of the most common choices are cubic splines, i.e. piecewise cubic functions of class C^2 (twice derivable functions with continuous second derivative)— the condition of having a continuous second derivative ensures the smoothness of the cubic spline. Polynomial splines have the capacity of adapting themselves to the variable smoothness of the signal, but they are influenced by the positioning of the knots. For details on this topic, the reader can refer to Green and Silverman (1994).

Another basis construction, that proved to have remarkable properties in representing many functions (including functions with discontinuities), is the

one that led to wavelet bases, which we will now introduce.

2.2 Wavelet bases

The construction of a wavelet basis is centred on the *mother wavelet*, a function usually denoted by ψ, assumed to have an oscillatory behaviour, hence the name of wavelet, and to be square integrable. Mathematically, we express these properties as $\psi \in L^2(\mathbb{R})$ and $\int_{-\infty}^{\infty} \psi(x)dx = 0$. Standard references in the wavelet literature are the monographs of Meyer (1992), Daubechies (1992), Mallat (1999) and Vidakovic (1999).

2.2.1 Wavelet families

Wavelet families are built using *translations* ($\psi(x) \to \psi(x+b), b \in \mathbb{R}$) and *re-scalings* ($\psi(x) \to \psi(ax), a \neq 0$) of the mother wavelet, ψ. The re-scaling is most commonly termed as 'dilation', but it has to be understood as shrinking as well as stretching. The dilation and translation parameters may be allowed to vary either in a continuous or in a discrete manner, depending on their ultimate use.

The mother wavelet ψ and the dilation and translation parameters $a, b \in \mathbb{R}$, $a \neq 0$ can generate a *wavelet family*, $\{\psi_{a,b}(x)\}_{a,b}$, where each element is nothing else but a re-scaled and translated version of the mother wavelet:

$$\psi_{a,b}(x) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{x-b}{a}\right). \tag{2.4}$$

The wavelet family is indexed by two subscripts— the first one indicates the wavelet scale, a^{-1}, and the second one its location. The normalisation parameter $\frac{1}{\sqrt{|a|}}$ appears in order to preserve the L^2-norm of ψ: $\|\psi_{a,b}\|_{L^2} = \|\psi\|_{L^2}$.

Any $f \in L^2(\mathbb{R})$ can be well approximated by a continuous linear superposition of wavelet functions, result known under the name of Calderón's repro-

ducing identity (see Daubechies (1992) for details):

$$f(x) = C_\psi^{-1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{a^2} < f, \psi_{a,b} >_{L^2} \psi_{a,b}(x) da db, \tag{2.5}$$

where $C_\psi = \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega$ is assumed to be finite for the above formula to have sense.

There is an obvious similarity between decomposition (2.5) and the function representation (2.3) using a Fourier transform. We should note though that while the coefficients in the Fourier decomposition only provide information about the amplitude associated with each frequency, in the wavelet decomposition above the coefficients provide information on the amplitude of the wavelet at both a given scale and location.

In what follows we will mainly concentrate on discrete wavelet constructions. When the scale and location parameters a, b *are discrete*, first of all a *choice of values* has to be made.

Possibly the most commonly used choice is $a = 2^{-j}$ and $b = 2^{-j}k$ with $j, k \in \mathbb{Z}$, which leads to obtaining the *discrete (decimated) wavelet family* $\{\psi_{j,k}(\cdot)\}_{j,k\in\mathbb{Z}}$, where $\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k)$. Information about the function being decomposed with this family will be available at locations of the form $2^{-j}k$ within each scale 2^j. Note that with increasing j, the wavelet scale also increases and the sampling gets finer— see figure 2.1. If ψ has compact support, then with increasing j, $\psi_{j,k}(\cdot)$ becomes 'taller' and narrower.

Another choice in applications is $a = 2^{-j}$ and $b = k$ with $j, k \in \mathbb{Z}$, which leads to the construction of the *nondecimated wavelet family* $\{\psi_{j,k}(\cdot)\}_{j,k\in\mathbb{Z}}$, where $\psi_{j,k}(x) = 2^{j/2}\psi(2^j(x - k))$. Here wavelets exist at all integer locations, rather than at a dyadic pyramid.

So far we have only constructed families of wavelets, and we have seen that when the scale and location parameters vary continuously, any square integrable function can be represented as a linear continuous superposition of elements from the wavelet family $\{\psi_{a,b}(\cdot)\}_{a,b}$ (formula (2.5)).

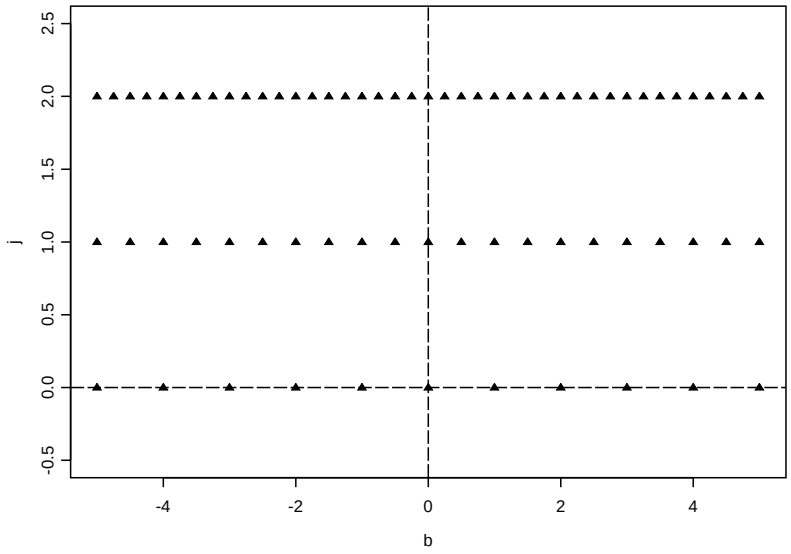

Figure 2.1: Critical sampling: wavelet functions exist at dyadic rational locations within each scale.

A similar expansion can be obtained in the discrete setting. More exactly, for certain choices of $\psi \in L^2(\mathbb{R})$, the decimated wavelet family $\{\psi_{j,k}(\cdot)\}_{j,k\in\mathbb{Z}}$ forms an orthonormal basis for $L^2(\mathbb{R})$. This ensures that any $f \in L^2(\mathbb{R})$ can be represented as

$$f(x) = \sum_{j\in\mathbb{Z}} \sum_{k\in\mathbb{Z}} < f, \psi_{j,k} > \psi_{j,k}(x), \tag{2.6}$$

i.e. any square integrable function can be written as the limit of a linear combination of wavelet functions at different scales and locations. As in the case of Fourier series, we can regard the function f as being converted into a set of coefficients, only that this time they are doubly indexed.

A common way of introducing wavelet bases and highlighting their properties is by constructing them within the versatile framework of a multiresolution analysis (MRA), introduced by Mallat (1989a,b). Essentially, a MRA of $L^2(\mathbb{R})$ allows for the approximation of any $f \in L^2(\mathbb{R})$ at different resolutions by projecting f on a sequence of approximation spaces. It is useful to think of the

approximations at different resolution levels as a 'zooming' process: a higher resolution level is equivalent to zooming in and obtaining a fine, detailed representation of f, while a lower resolution level is equivalent to zooming out and getting a coarse representation of f.

2.2.2 Multiresolution analysis (MRA) construction

Definition 2.2.1. *An* **(orthogonal) multiresolution analysis (MRA) of** $L^2(\mathbb{R})$ *consists of a sequence of successive approximation vector spaces* V_j *(note that each* V_j *is a space of functions), with the following properties:*

1. V_j *is a closed subspace of* $L^2(\mathbb{R})$, $\forall j \in \mathbb{Z}$,
2. $\bigcap_{j\in\mathbb{Z}} V_j = \{0_{L^2(\mathbb{R})}\}$,
3. $\overline{\bigcup_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R})$,
4. $V_j \subset V_{j+1}, \quad \forall j \in \mathbb{Z}$, *hence* $\{0_{L^2(\mathbb{R})}\} \subset \cdots \subset V_{-1} \subset V_0 \subset V_1 \subset \cdots \subset L^2(\mathbb{R})$,
5. $f(\cdot) \in V_j \Leftrightarrow f(2\cdot) \in V_{j+1}$, $\forall j \in \mathbb{Z}$,
6. *There exists* $\varphi \in V_0$ *a scaling function with* $\int_{-\infty}^{\infty} \varphi(x)dx = 1$, *such that* $\{\varphi(\cdot - k)\}_{k\in\mathbb{Z}}$ *is an orthonormal basis of* V_0.

We will refer to the index j as the resolution level, or just briefly the level or scale in the MRA ladder, although strictly speaking, functions at level j have scale 2^j. Larger j corresponds to a finer scale and a finer approximation space V_j.

The MRA construction has important implications, as follows:

(i) Conditions 3 and 4 imply that $\lim_{j\to+\infty} \mathrm{Proj}_{V_j} f = f$, and the spaces $\{V_j\}_{j\in\mathbb{Z}}$ can be used for approximating functions— *any function* f *can be gradually approximated by its projections on the* $\{V_j\}_{j\in\mathbb{Z}}$ *spaces,* $\{Proj_{V_j} f\}_{j\in\mathbb{Z}}$,

(ii) Condition 5 gives the multiresolution property of the ladder: *any* V_j *is a scaled version of* V_0— $f(\cdot) \in V_0 \Leftrightarrow f(2^j\cdot) \in V_j$, $\forall j \in \mathbb{Z}$,

(iii) The multiresolution property of the V_j's together with the fact that $\{\varphi_{0,k}(\cdot)\}_{k\in\mathbb{Z}}$ is an orthonormal basis of V_0, ensures that $\{\varphi_{j,k}(\cdot)\}_{k\in\mathbb{Z}}$ is an orthonormal basis of V_j, $\forall j \in \mathbb{Z}$, where $\varphi_{j,k}(x) = 2^{j/2}\varphi(2^j x - k)$.

As $\varphi \in V_0 \subset V_1$ and $\{\varphi_{1,k}(\cdot)\}_{k\in\mathbb{Z}}$ is an orthonormal basis of V_1, we can write

$$\varphi(x) = \sum_{k\in\mathbb{Z}} h_k \sqrt{2}\varphi(2x - k). \tag{2.7}$$

The above relation is known under the name of the *scaling or refinement equation*, and its (unique) coefficients $\{h_k\}_{k\in\mathbb{Z}} \in l^2(\mathbb{Z})$ form a vector that is often referred to as a *low-pass filter*, for reasons that will soon become obvious. We will see that this filter has an averaging or smoothing effect by preserving the low frequencies and suppressing the high ones.

We note here that the MRA conditions are sometimes re-written into the Fourier domain. For instance, fundamental properties of the filter taps h_k ($\sum_{k\in\mathbb{Z}} h_{2k} = \sum_{k\in\mathbb{Z}} h_{2k+1} = 1/\sqrt{2}$ and $\sum_{k\in\mathbb{Z}} h_k h_{k-2l} = \delta_{l,0}$) are obtained by following a Fourier approach. For a presentation of the multiresolution approach in the Fourier domain, the reader can consult Vidakovic (1999).

The key property of the multiresolution approach is (Daubechies (1992)) that *whenever a sequence of closed subspaces of $L^2(\mathbb{R})$ satisfies assumptions 1-6, there exists (but is not unique) an orthonormal wavelet basis $\{\psi_{j,k}(\cdot)\}_{j,k\in\mathbb{Z}}$ of $L^2(\mathbb{R})$ with $\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k)$ such that $Proj_{V_{j+1}} f = Proj_{V_j} f + \sum_{k\in\mathbb{Z}} < f, \psi_{j,k} > \psi_{j,k}$, and the mother wavelet ψ can be constructed explicitly.*

Wavelets would not have become the important tool that they are today if it had not been for the realisation that many wavelet bases, with various properties, can be constructed by starting from different mother wavelet functions.

Constructing the mother wavelet.

The construction of a mother wavelet is centred on the idea of representing the information lost when moving from a finer approximation space to the next coarser one (say from V_{j+1} to V_j). Take W_j to be the orthogonal complement of V_j in V_{j+1}, hence $V_{j+1} = V_j \oplus W_j$, $\forall j \in \mathbb{Z}$ (where '$\oplus$' denotes direct sum

of spaces) and any function in V_{j+1} can be uniquely represented as a sum of a function in V_j and a function in W_j. Therefore the space W_j contains the lost information when moving from a representation on V_{j+1} to a representation on V_j.

Condition 4 of the MRA and the way of construction for each W_j, imply that $W_j \perp W_{j'}$, $\forall j \neq j'$ and for any fixed $j_0 \in \mathbb{Z}$

$$V_J = V_{j_0} \oplus (\oplus_{j=j_0}^{j=J-1} W_j),\ \forall J \geq j_0 + 1.$$

This shows that any function in V_J can be 'recovered' by taking the sum of its approximation at a lower resolution (j_0) and the functions that represent the detail lost between levels J and j_0.

In the limit ($J \to \infty$), using property 3 of the MRA, the above formula becomes

$$L^2(\mathbb{R}) = V_{j_0} \oplus (\oplus_{j=j_0}^{\infty} W_j), \tag{2.8}$$

and together with property 2 we further obtain

$$L^2(\mathbb{R}) = \oplus_{j \in \mathbb{Z}} W_j. \tag{2.9}$$

These are equivalent ways of representing the space of square integrable functions by using a multiresolution decomposition.

Due to the orthogonal nature of these decompositions, in order to obtain an orthonormal basis for $L^2(\mathbb{R})$ it is now enough to find $\{\psi_{j,k}(\cdot)\}_{k \in \mathbb{Z}}$, an orthonormal basis for W_j, $\forall j \in \mathbb{Z}$. Furthermore, because the spaces W_j inherit the scaling property (5) from the V_j, *it suffices to construct* ψ *such that* $\{\psi_{0,k}(\cdot)\}_{k \in \mathbb{Z}}$ *is an orthonormal basis for* W_0.

It was proved (see Daubechies (1992) for the comprehensive construction) that a possible choice that yields *a desired mother wavelet,* ψ is

$$\psi(x) = \sum_{k \in \mathbb{Z}} (-1)^k h_{1-k} \sqrt{2} \varphi(2x - k). \tag{2.10}$$

Essentially, the decomposition in (2.10) follows the same rationale as the refinement relation for the scaling function: decompose the mother wavelet $\psi \in W_0 \subset V_1$ on the orthonormal basis of the space V_1 and obtain a (unique) set of coefficients $\{g_k\}_{k\in\mathbb{Z}} \in l^2(\mathbb{Z})$. The desired orthonormality of $\{\psi_{0,k}(\cdot)\}_{k\in\mathbb{Z}}$, together with the properties of the MRA construction, yield a possible solution $g_k = (-1)^k h_{1-k},\ k \in \mathbb{Z}$, which corresponds to the mother wavelet ψ in the above formula. When filters $\{h_k\}_{k\in\mathbb{Z}}$ and $\{g_k\}_{k\in\mathbb{Z}}$ are linked by the previous formula, they are said to be *quadrature mirror filters*. We will see that the set $\{g_k\}_{k\in\mathbb{Z}}$ is a high-pass filter, only preserving the high frequencies in the signal.

The refinement relations can be written at any scale and translation as

$$\varphi_{j,l}(x) = \sum_k h_{k-2l}\varphi_{j+1,k}(x) \text{ and } \psi_{j+1,l}(x) = \sum_k g_{k-2l}\varphi_{j+1,l}(x). \tag{2.11}$$

As a consequence, we can express the filters as

$$h_{k-2l} =< \varphi_{j,l}, \varphi_{j+1,k} > \text{ and } g_{k-2l} =< \psi_{j,l}, \varphi_{j+1,k} >,\ \forall j,k,l. \tag{2.12}$$

If the analytical form of the scaling function is available (note that this is not always the case), by taking $j = l = 0$ in formulas (2.12) we can obtain the low- and high-pass filters $h_k =< \varphi, \varphi_{1,k} >$, respectively $g_k =< \psi, \varphi_{1,k} >$.

Relations (2.12) also highlight that if the scaling function has compact support, then the low-pass filter is finite (i.e. it has a finite number of non-zero filter coefficients), and consequently so is the high-pass filter. By using the refinement relation (2.10) it follows that the wavelet function can be expressed as a finite linear combination of compactly supported functions, and consequently ψ is also compactly supported. We will see this principle illustrated in the following example.

Examples of wavelet bases.

The Haar wavelet basis. The oldest wavelet basis is the Haar basis, introduced by Alfred Haar in 1910. It also has the simplest construction, and

most introductory texts on wavelets would start with a presentation of this basis due to its pedagogical value.

The construction starts with the simple scaling function

$$\varphi(x) = \mathbb{1}_{[0,1)}(x), \tag{2.13}$$

where $\mathbb{1}_X(x) = \begin{cases} 1, & \text{if } x \in X \\ 0, & \text{otherwise} \end{cases}$, for a given set X.

Since $h_k =< \varphi, \varphi_{1,k} >$, it follows that the low-pass filters are defined by $h_0 = h_1 = 2^{-1/2}$ and $h_k = 0$, $\forall k \in \mathbb{Z} \setminus \{0, 1\}$.

Consequently, due to the quadrature mirror relation, the high-pass filters are given by $g_0 = -g_1 = 2^{-1/2}$ and $g_k = 0$, $\forall k \in \mathbb{Z} \setminus \{0, 1\}$. Hence $\psi(x) = 2^{-1/2}\varphi_{1,0}(x) - 2^{-1/2}\varphi_{1,1}(x)$, which translates into the following expression for the Haar mother wavelet function

$$\psi(x) = \mathbb{1}_{[0,1/2)}(x) - \mathbb{1}_{[1/2,1)}(x), \tag{2.14}$$

which generates the Haar wavelet family

$$\psi_{j,k}(x) = 2^{j/2}\{\mathbb{1}_{[2^{-j}k, 2^{-j}(k+\frac{1}{2}))}(x) - \mathbb{1}_{[2^{-j}(k+\frac{1}{2}), 2^{-j}(k+1))}(x)\}. \tag{2.15}$$

By construction, wavelets in the Haar family have compact support— $\text{supp}(\psi_{j,k}) = [2^{-j}k, 2^{-j}(k+1)]$. Haar wavelets at the same scale do not overlap (unless they are the same), and if they are at different scales, they either have non-overlapping supports, or the support of one of them is included in a region where the other wavelet is constant.

Figure 2.2 shows the Haar scaling function and a few members of the Haar wavelet family, corresponding to $j = 0$, $j = 1$ with $k = 0$ and 1. Note how the change in scale affects the length of the wavelet support— the finer the scale, the more wavelet functions are needed in order to cover the same interval.

It can be checked that the Haar wavelet family forms an orthonormal basis

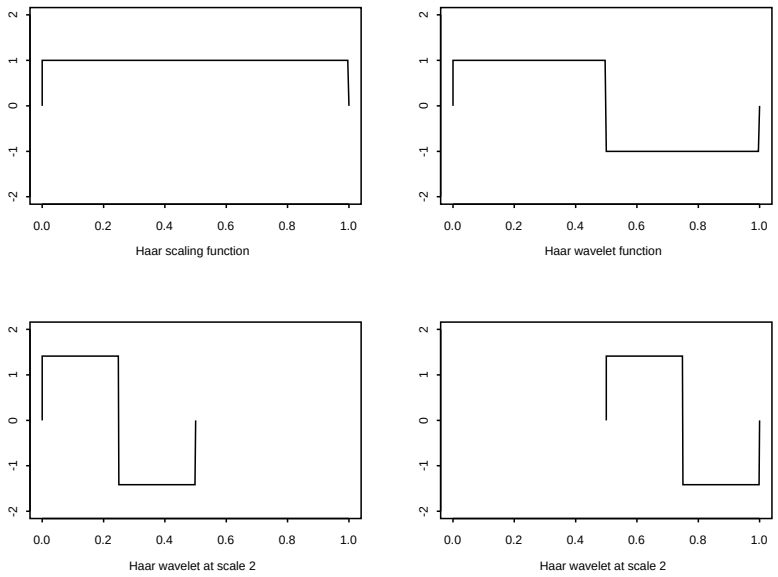

Figure 2.2: The Haar scaling function, and scaled and translated Haar wavelets.

on which any square integrable function can be decomposed (for proof see chapter 2 of Kovac (1998)).

We previously mentioned that a good basis for decomposing a signal is one that would match the signal smoothness. This would ensure that few basis functions are needed to describe the signal. Since the components of the Haar basis are discontinuous functions, it easy to imagine that this wavelet basis does not do a good job when representing a smooth function, and many wavelet coefficients are needed to describe the signal.

It is therefore apparent the need of constructing smooth wavelet bases.

The Shannon wavelet basis. While Haar wavelets have compact support in time domain, the Shannon wavelet family is based on a mother wavelet with compact support in the frequency domain. Here we only outline its construction, and the reader is directed to the monograph of Vidakovic (1999) (page 63) for details.

The construction starts with the scaling function defined in the Fourier domain by $\hat{\varphi}(\omega) = \mathbb{1}_{[-\pi,\pi)}(\omega)$. In the time domain, this translates into having the

infinitely differentiable scaling function $\varphi(x) = \frac{\sin(\pi x)}{\pi x}$, with infinite support. The wavelet function can be expressed as $\psi(x) = \varphi(x - \frac{1}{2}) - 2\varphi(2x - 1)$, hence it is also smooth. However, the consequent infinite length filters induce bad localization properties, which is not desirable.

So far, two issues arise when designing wavelet bases: ideally, they would have some degree of smoothness and finite filters. We will soon see that due to Ingrid Daubechies such wavelet constructions are possible.

2.2.3 Function representation on a wavelet basis

Let us now come back to the issue of representing a function on a wavelet basis. Using the orthogonal multiresolution representation (2.8) for the space of square integrable functions, and since $\{\varphi_{j,k}(\cdot)\}_k$, $\{\psi_{j,k}(\cdot)\}_k$ are orthonormal bases for V_j, respectively W_j, it follows that $\{\varphi_{j_0,k}(\cdot)/k \in \mathbb{Z}\} \bigcup \{\psi_{j,k}(\cdot)/j \geq j_0, k \in \mathbb{Z}\}$ is an orthonormal basis for $L^2(\mathbb{R})$. Consequently, one can write

$$f(x) = \sum_{k \in \mathbb{Z}} < f, \varphi_{j_0,k} > \varphi_{j_0,k}(x) + \sum_{j \geq j_0} \sum_{k \in \mathbb{Z}} < f, \psi_{j,k} > \psi_{j,k}(x), \quad \forall f \in L^2(\mathbb{R}). \tag{2.16}$$

The first component in the above sum gives the overall large-scale behaviour of f at the coarse scale j_0 (and is in fact $\text{Proj}_{V_{j_0}} f(x)$), while the second component represents the fine-scale (detail) features of f, collected when moving to the coarse scale.

Since $\{\psi_{j,k}(\cdot)\}_{j,k}$ is an orthonormal basis for $L^2(\mathbb{R})$, we obtain an equivalent representation to Calderón's reproducing identity, that motivated this section. Any $f \in L^2(\mathbb{R})$ can be represented as

$$f(x) = \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} < f, \psi_{j,k} > \psi_{j,k}(x), \tag{2.17}$$

which shows that any square integrable function can be written in the limit as a linear combination of wavelet functions at different scales and locations.

The wavelet coefficients $\{< f, \psi_{j,k} >\}_{j,k \in \mathbb{Z}}$ provide information about f at

scale 2^j near position $2^{-j}k$. Since together with the wavelet basis used for decomposing the function, the wavelet coefficients completely characterize f, we will be interested in computing them. The framework in which we set up the computations of the scaling and wavelet coefficients is known under the name of **the discrete wavelet transform** (DWT).

2.2.4 The discrete wavelet transform

The discrete wavelet transform makes use of the nested structure of the multiresolution analysis, and it provides a fast scheme for recursively computing the scaling and wavelet coefficients, without having to evaluate the inner products $< f, \varphi_{j,k} >$ and $< f, \psi_{j,k} >$. Mallat (1989a,b) made the connection between the MRA and derived a fast algorithm for calculating the coefficients that arise in the decomposition of f.

Let $f \in L^2(\mathbb{R})$ and for a fixed level j, take its approximation on the space V_j: $\text{Proj}_{V_j} f(x) = \sum_{k\in\mathbb{Z}} c_{j,k}\varphi_{j,k}(x)$, where the set $\{c_{j,k} =< f, \varphi_{j,k} >\}_k$ gives the scaling coefficients. The detail produced by moving between two consecutive approximation spaces V_j and V_{j+1} is given by $\text{Proj}_{W_j} f(x) = \sum_{k\in\mathbb{Z}} d_{j,k}\psi_{j,k}(x)$, where $\{d_{j,k} =< f, \psi_{j,k} >\}_k$ are the corresponding wavelet coefficients at level j.

Since $\text{Proj}_{V_{j+1}} f(x) = \text{Proj}_{V_j} f(x) + \text{Proj}_{W_j} f(x)$, we obtain

$\sum_k c_{j+1,k}\varphi_{j+1,k}(x) = \sum_l c_{j,l}\varphi_{j,l}(x) + \sum_l d_{j,l}\psi_{j,l}(x)$.

Relations in (2.12) and the orthogonality of the V_j and W_j spaces imply

$$c_{j,l} = \sum_k h_{k-2l} c_{j+1,k}, \tag{2.18}$$

$$d_{j,l} = \sum_k g_{k-2l} c_{j+1,k}. \tag{2.19}$$

In the literature, these formulas are known as the DWT of the function f and they give a recursive way for computing the scaling, respectively wavelet coefficients at each coarser scale $j \in \mathbb{Z}$ associated to the multiresolution de-

composition of f.

The DWT can also be recursively reversed— the scaling coefficients at each finer scale can be obtained from the scaling and wavelet coefficients at the previous (coarser) scale:

$$c_{j+1,k} = \sum_l h_{k-2l} c_{j,l} + \sum_l g_{k-2l} d_{j,l}. \tag{2.20}$$

For practical problems, the role of the DWT is invaluable, hence it is worth presenting how the above methodology is adapted to an observed set of data.

2.2.5 The DWT for discrete data

In practice, we do not have a continuous function f to decompose on the approximation spaces V_j, but merely a collection of discrete observations of the function values, $f(x_i)$, at equally spaced locations of the form $x_i = x_0 + i\Delta$, for some fixed x_0 and Δ. The number of observations is assumed to be of the form 2^J for some fixed J, and for simplicity it is usually assumed that $x_0 = 0$ and $\Delta = 2^{-J}$, hence $x_i = 2^{-J} i$.

Therefore, we start with a sequence of observations $(x_i, f(x_i))_{i \in \overline{0,2^J-1}}$. In order to apply the above algorithm, first interpolate the observations by using the basis of scaling functions from the space V_J: denote $c_{J,i} = f(x_i)$ and construct the function $\tilde{f} \in V_J$ by taking $\tilde{f}(x) = \sum_{i \in \mathbb{Z}} c_{J,i} \varphi_{J,i}(x)$ (a treatment of the data outside the boundaries must be established, but we postpone this issue). Use the function $\tilde{f}$ as an approximation for the observed function f. It is important to realise that in consequence the wavelet coefficients of $\tilde{f}$ represent an approximation for the wavelet coefficients $d_{j,k} =< f, \psi_{j,k} >$ of f.

However, by an abuse of notation, we will still denote by $c_{j,k}$ and $d_{j,k}$ the scaling, respectively wavelet coefficients of $\tilde{f}$. Often, they are referred to as the *empirical* scaling and wavelet coefficients of f, and mathematically they are related to their continuous equivalents through a proportionality factor (see Abramovich *et al.* (2000) for instance).

We start the DWT with the finest scaling coefficients $\underline{c}^J = \{c_{J,k}\}_k$, and by applying (2.18) and (2.19) obtain the sequences of scaling and wavelet coefficients at the next coarser level, $\underline{c}^{J-1} = \{c_{J-1,k}\}_k$ and $\underline{d}^{J-1} = \{d_{J-1,k}\}_k$. Repeat the procedure for $\underline{c}^{J-1}$, and re-iterate until the desired coarse level, say j_0, has been reached. At the end of this process, the initial sequence $\underline{c}^J$ in time domain has been replaced by the sequence $(\underline{c}^{j_0}, \underline{d}^{j_0}, \underline{d}^{j_0+1}, \ldots, \underline{d}^{J-1})$ in the wavelet domain.

For any level $j < J$, the sequence $\underline{c}^j$ is referred to as *smooth*, since it provides a coarser description of the initial signal, while $\underline{d}^j$ is often referred to as *detail*, since it extracts the features lost when representing the signal in a coarser version.

The inverse DWT (2.20) can be applied to revert from the scaling and wavelet coefficients decomposition $(\underline{c}^{j_0}, \underline{d}^{j_0}, \underline{d}^{j_0+1}, \ldots, \underline{d}^{J-1})$ to the initial set of values, $\underline{c}^J$.

When periodic handling of the boundaries is enforced (see next set of remarks, boundary issues), at each step of the DWT the sequence of scaling coefficients is halved, and as such at each level j, the sequences $\underline{c}^j$ and $\underline{d}^j$ have length 2^j (see Nason and Silverman (1994) for details). Conversely, in the inverse DWT, the number of scaling coefficients is doubled at each step. The two sequences in time and wavelet domain have therefore the same number of elements, 2^J.

Figure 2.3 (top right hand side) shows the decomposition of the *HeaviSine* signal (see section 4.1 for its description) sampled at $n = 2^{10}$ regularly spaced locations, into smooth and detail coefficients, using the Haar filters. The coefficients correspond to Haar wavelets at 6 scales, and are represented in a layered plot: the top row contains the finest details (corresponding to $\underline{d}^9$ in our notation) and the last row the smooth coefficients ($\underline{c}^4$ in our notation). Within level j, each coefficient $d_{j,k}$ is plotted at the location corresponding to the midpoint of the support of the wavelet function $\psi_{j,k}$, i.e. $2^{-j}(k + \frac{1}{2})$. Note how the magnitude of the coefficients increases with coarsening the scale.

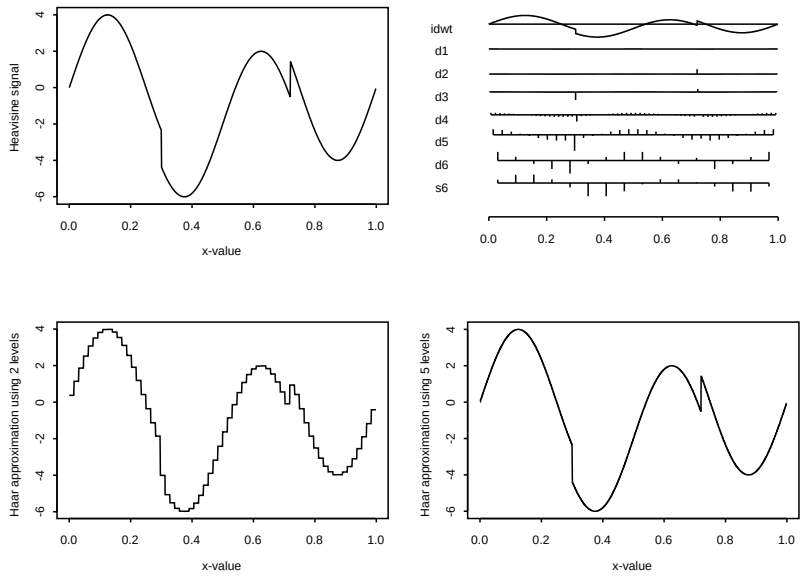

Figure 2.3: On the top row, (left) the *HeaviSine* signal sampled at 1024 locations and (right) its empirical Haar scaling and wavelet coefficients; on the bottom row, two possible approximations of *HeaviSine* based on Haar wavelets.

In the second row of figure 2.3 we show two approximations of the *HeaviSine* signal. The approximation on the left is obtained by using only the (empirical) scaling coefficients ($\underline{c}^4$) and the lowest two levels of detail ($\underline{d}^4, \underline{d}^5$). As $V_6 = V_4 \oplus W_4 \oplus W_5$ it follows that the approximation is $\text{Proj}_{V_6} f$. The approximation on the right is obtained by using all levels of detail except for the last one, $(\underline{c}^4, \underline{d}^4, \cdots, \underline{d}^8)$, hence the approximation is $\text{Proj}_{V_9} f$. Looking upwards from the bottom row, each row of detail coefficients brings extra information on the signal, and contributes to a finer representation of it— the more levels of detail we include, the better the signal is represented. However, the quality of an approximation depends not only on the level of projection, but also on the wavelet basis used for decomposition: sparser wavelet coefficients will generally ensure better signal representation at the same scale.

Let us make a few remarks now.

Remarks

1. **Filters**. The DWT formulas (2.18) and (2.19) justify the terminology of

low-pass, high-pass filters for $\{h_k\}_k$, respectively $\{g_k\}_k$: the filter $\{h_k\}_k$ is used for smoothing the initial signal, while the $\{g_k\}_k$ is used for extracting the detail lost by moving from one level to the next coarser one.

2. **Filter notation of the DWT**. The DWT formulas can be also viewed as the result of filter operations on the initial sequence $\underline{c}^J$. The construction is based on the use of low- and high-pass filters, and on decimation, defined next. The action of a filter H on a sequence $\underline{c} = \{c_n\}_n$ is defined by $(H\underline{c})_j = \sum_n h_{n-j}c_n$. The decimation operator acts on the sequence $\underline{c}$ by retaining only its values on the even positions, $(\mathscr{D}_0\underline{c})_j = c_{2j}$. In this notation, the DWT formulas can be re-written as $\underline{c}^j = \mathscr{D}_0 H \underline{c}^{j+1}$ and $\underline{d}^j = \mathscr{D}_0 G \underline{c}^{j+1}$, where H and G denote the low-, high-pass filters respectively.

3. **Boundary issues**. When filters longer than Haar are used, problems may appear at the boundaries when the range of data indices does not overlap the range of filter indices. Various approaches to this problem have been taken, ranging from periodizing the signal or padding it out with zeroes (see Nason and Silverman (1994) for a detailed discussion) to an entirely new construction by Cohen et al. (1993), designed to build wavelets on an interval.

4. **Matrix representation of the DWT**. Another way of writing the DWT comes from the realisation that at every step of the DWT, we are in effect representing the signal on two different bases. In the first instance f is represented on the basis $\{\varphi_{J,k}(\cdot)\}_k$ of the space V_J. By moving to the next coarser level, $J-1$, we represent f on the basis $\{\{\varphi_{J-1,k}(\cdot)\}_k, \{\psi_{J-1,k}(\cdot)\}_k\}$, and so on. Since any change of basis of this type can be represented by matrix multiplication, it follows that *the DWT itself can be represented by matrix multiplication.* Hence if we denote the vector obtained by the DWT of $\underline{c}^J$ by $\underline{d} = (\underline{c}^{j_0}, \underline{d}^{j_0}, \underline{d}^{j_0+1}, \ldots, \underline{d}^{J-1})$, then $\underline{d} = W\underline{c}^J$, where $W \in \mathscr{M}_{2^J,2^J}$ is the matrix built in the DWT pro-

cess. As the bases used for representing the signal at each step are orthonormal, it follows that W is an orthogonal matrix ($WW^T = I_{2^J}$, where I_n is the identity matrix of order n). The inverse DWT follows as $\underline{c}^J = W^T \underline{d}$. The matrix repesentation of the DWT is particularly useful when tackling nonparametric regression problems, which we will investigate later.

We end this section by noting that the DWT through its recursive way of computing the smooth and detail (see (2.18), (2.19) and (2.20)) highlights how desirable filters of finite support are, as they ensure finite summations.

2.2.6 Vanishing moments of wavelet functions

One of the most attractive features of wavelets is the property of *vanishing moments*.

Definition 2.2.2. *We say that a wavelet function ψ has $m+1$ vanishing moments if*

$$\int_{-\infty}^{\infty} x^l \psi(x) dx = 0, \; \forall l \in \overline{0, m}. \tag{2.21}$$

Let us now examine the implications of this property. The most obvious one, since $< x^l, \psi(x) >= 0$, $\forall l \in \overline{0, m}$, is that the *wavelet coefficients of any polynomial of degree at most m are annihilated in a decomposition on such a wavelet basis*. Or, in other words, any polynomial of degree at most m can be written as a linear combination of integer translates of the scaling function, $\{\varphi_{0,k}(\cdot)\}_{k \in \mathbb{Z}}$.

The Haar wavelet basis for example, has only 1 vanishing moment (arising from the admissibility condition), which means that it produces exactly zero detail only for constant functions.

Daubechies (1992) established a connection between the number of vanishing moments of a wavelet function and its smoothness. More exactly, she proved that any wavelet with the first m derivatives continuous and bounded,

and which decays like $|\psi(x)| \leq \frac{C}{(1+|x|)^{\alpha}}$, $\alpha > m+1$, has $m+1$ vanishing moments.

The first constructions of wavelet bases with compact support and with a specified number of vanishing moments are due to Daubechies (1988). These wavelet bases are the famous families *Daubechies extremal phase* and *Daubechies least asymmetric* (the second family consists of wavelets which are by construction closer to being symmetrical, hence their name). Wavelets in these families are indexed by a number which indicates the number of vanishing moments the wavelet is designed to have (D1– Haar wavelet, D2– Daubechies extremal phase wavelet with compact support and 2 vanishing moments, and so on up to D10; the least asymmetric wavelets are usually denoted by S2 up to S10). Such a wavelet with say $m+1$ vanishing moments, has finite filters of length $2m+2$, and the support of its scaling function is $[0, 2m+1]$, while the support of the wavelet function is $[-m, m+1]$.

These wavelets can be designed by re-writing their properties, including that of having a specific number of vanishing moments, in terms of their filter coefficients. Having finite filters ensures that a solvable system in the filter taps is obtained (see Vidakovic (1999), page 92).

Unlike the examples that we encountered so far, compactly supported Daubechies wavelets do not have an analytical form. However, using the values of the associated filters, the scaling function can be plotted by repeatedly using the scaling equation (2.7) to compute its values first at integer locations $\varphi(k)$, and then at dyadic rationals, $\varphi(2^{-j}k)$ for $j = 1, 2, \ldots$; a similar procedure can be used to plot the wavelet function. Other techniques for obtaining the values of the scaling and wavelet functions have been developed— see for example the cascade algorithm of Daubechies (1992).

It has been shown (see Daubechies (1992)) that wavelet and scaling function regularity increases with m. Hence wavelets with a higher number of vanishing moments are smoother and have longer support. The possible discontinuities in the signal will then influence the wavelet coefficients more. Figure 2.4 gives

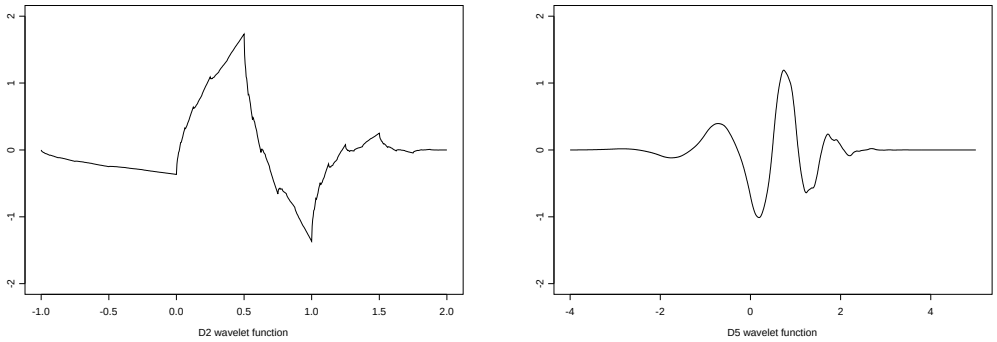

Figure 2.4: D2 and D5 mother wavelets.

two examples of members of the Daubechies extremal phase wavelet family. Note how the wavelet smoothness increases with its vanishing moments.

Let us come back to the example in figure 2.3. Although *HeaviSine* is fairly smooth, there we used Haar wavelets for decomposing it, hence a very sparse wavelet representation was not to be expected. Compare the Haar wavelet decomposition to the one in figure 2.5, where the signal is decomposed using the smooth D5 wavelet basis (which annihilates polynomials up to degree 4). Remembering that the signal was sampled at 1024 locations, note that very few wavelet coefficients are different from zero. Consequently, the approximation using only the scaling coefficients and two levels of detail is now much better than the one obtained by using Haar wavelets (see figure 2.5, right versus figure 2.3, left on the bottom row).

In general there is no rule as to which wavelet works best for providing a sparse representation of the signal. For each type of signal a different wavelet choice may be the best one, depending on the signal features, such as the number of discontinuities it presents. As a simple rule of thumb, the signal smoothness should be closely matched by the regularity of the wavelet used to decompose it. In practice though, the smoothness of the signal is seldom known, and this leaves us with a serious problem when it comes to choosing the wavelet. The best number of levels down to which we decompose the signal (j_0) is another issue which is left as a choice. We will come back to these issues

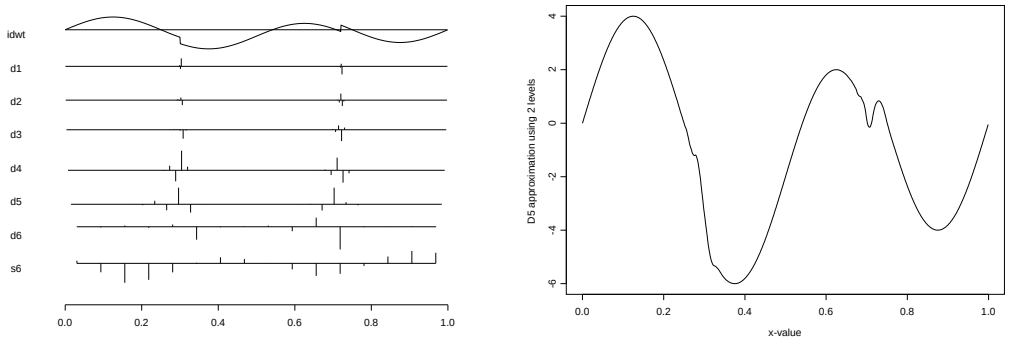

Figure 2.5: Empirical scaling and wavelet coefficients obtained by decomposing *HeaviSine* using D5 wavelet basis (left). Reconstruction of *HeaviSine* based on D5 wavelets and their corresponding details at the two coarsest scales (right).

when discussing nonparametric regression.

2.2.7 Other wavelet constructions

Several extensions of the wavelet methodology have been constructed, such as wavelet packets, biorthogonal wavelets and periodized wavelets, which we will briefly review.

Wavelet packets were introduced by Coifman *et al.* (1989). They are generalisations of orthogonal wavelets, obtained by linear combinations similar to refinement equations. Their construction enhances the oscillatory character of the wavelet, so additionally they are indexed over an oscillation parameter; the collection of all their dilations and translations forms the 'library' of all packet functions, and any orthogonal basis selected from the library is a wavelet packet basis of $L^2(\mathbb{R})$. Such an example is the Walsh basis whose construction relies on Haar wavelets. When using a wavelet packet to decompose a signal, in effect both filters (low- and high-pass) are applied at every step of the transform not only on the scaling coefficients, but also on the detail and the resulting coefficients are grouped into 'crystals'. This representation facilitates the selection of a best orthogonal basis (best according to a specified criterion), for details see Coifman and Wickerhauser (1992).

Biorthogonal wavelet bases. Orthogonality is a strong constraint in wavelet constructions, which led to the impossibility of constructing symmetrical, compactly supported wavelets with more than 1 vanishing moments. In a construction introduced by Cohen *et al.* (1992), orthogonality has been relaxed for biorthogonality.

The biorthogonality approach allows for more flexibility in the wavelet construction, and most biorthogonal wavelets are designed to have compact support and symmetry. Biorthogonal wavelets are constructed in the framework of two MRA's, usually referred to as primal and dual. In each of these MRA's, the bases are not orthonormal anymore, and the approximation and detail spaces lose their orthogonality. However, orthogonality exists between the approximation and detail spaces each belonging to the other MRA. The pairs of primal and dual scaling, wavelet functions are linked through a set of biorthogonality relations.

In this instance, the signal is reconstructed using a basis comprising (primal) scaling and wavelet functions, and decomposed on a *different* basis of (dual) scaling and wavelet functions. For the interested reader, a good review of multiresolution analyses, wavelet families and their applications is found in Jawerth and Sweldens (1994).

Wavelets on an interval. Often we work with functions that naturally exist on an interval, rather than on the whole real line. However, the wavelet bases that we have seen so far are constructed for $L^2(\mathbb{R})$. Much effort has been put into solving this problem. Cohen *et al.* (1993) devised a construction of wavelet bases for functions defined on an interval, and also proposed a corresponding fast wavelet transform.

Another approach is that of using periodized wavelets which span $L^2([0,1])$. Essentially, for each scale j and location k, the usual scaling and wavelet functions, $\varphi_{j,k}(\cdot), \psi_{j,k}(\cdot)$, are 'wrapped' around the interval $[0,1]$. The new scaling and wavelet functions are periodic functions with period 1. This construction is equivalent to periodizing the observed (finite) sequence of observations $\underline{c}^J$,

and then applying the usual DWT with the chosen wavelet basis. The wavelet coefficients have indices restricted to scales $j \geq 0$ and locations $k \in \overline{0, 2^j - 1}$, so the approximation spaces are finite dimensional. However, unless the function is itself periodic, this approach will introduce edge effects (see Vidakovic (1999)).

Multi-dimensional wavelet bases. We end this section by noting that the MRA and the construction of wavelet bases can be generalised to spaces of higher dimensions. This is especially useful when dealing with multi-dimensional functions, such as images. Several such extensions exist, from the one due to Mallat (1989b) in which the multi-dimensional wavelet functions are simply tensor products of univariate wavelets, to more complex, 'non-separable' ones, see for instance Meyer (1992). In Mallat's construction for two-dimensions, the approximation spaces become tensor products of their unidimensional equivalents, and the wavelet functions correspond to three directions: horizontal, vertical and diagonal.

2.3 The lifting scheme

We have already seen that classical wavelets cannot provide straightforward solutions in certain situations. Such examples occur when the sampled data lives on an interval, it is irregularly spaced or when its length is not of the form 2^J, for some $J \in \mathbb{Z}$. Several constructions designed to overcome these problems are known in the literature— wavelets on an interval, wavelets on curves and using wavelets on interpolated grids are just some examples, more will be presented later in the context of nonparametric regression. Sweldens (1996, 1998) introduced a new way for building wavelets, known as *the lifting scheme*, which can handle more general settings. Wavelet functions obtained through the lifting algorithm are known in the literature under the name of *second generation wavelets*, hence often the wavelet constructions that we presented so far are referred to as 'first generation' or 'classical'.

The lifting scheme still aims to represent the contents of a signal by a decomposition on a wavelet basis that uses few non-zero coefficients, as this facilitates signal compression and structure representation.

To fix notation, let us start with a vector $\underline{x} = \{x_i\}_{i \in \overline{1,n}}$ defining a (general) location structure of length n, on which data is to be collected. The value of a function of interest is then observed at each location x_i, yielding a vector $\underline{f} = \{f(x_i)\}_{i \in \overline{1,n}}$, the sampled signal. Note that while previously the grid $\underline{x}$ was considered to be regular, we now allow for its complete irregularity.

Let us start with an informal presentation of the lifting scheme, as introduced by Sweldens (1996). The lifting construction essentially consists of iterating three steps: split, predict and update.

Split. At this step the signal is subsampled (split) into two vectors: points that correspond to the even positions in the grid, and points that correspond to the odd positions in the grid.

Predict. The next task is to predict the f-values corresponding to the odd positions by using the information contained in the f-values corresponding to the even positions. The error in prediction (the difference between the true and predicted function values on the odd positions) is then quantified in a vector, called a *detail vector*.

Update. The f-values on the even positions are now updated by using linear combinations of the current f-values on the even positions and the detail vector obtained at the previous step. The purpose of this stage is to preserve some quantity from the initial signal, such as its mean value.

After re-iterating these steps on the updated (sub)sample, the initial signal $\underline{f}$ will be replaced by the remaining updated subsample (which reproduces the coarse features of the signal) and the detail coefficients accumulated throughout the process. This is similar to the DWT which replaces $\underline{f}$ with a set of scaling and wavelet coefficients, but note that the DWT is incapable of taking into account the irregularity of $\underline{x}$.

Sweldens (1998) introduced the lifting algorithm also under the theoretical

frame of a second generation (biorthogonal) MRA. This construction parallels the orthogonal MRA for classical wavelets, but there are a few essential differences: (i) the basis functions are no longer dilations and translations of one function, (ii) the filters are now both location and scale dependent, allowing for usage on irregular data. In this context, a Fourier approach to the wavelet construction is no longer feasible.

In what follows we will give a brief presentation of the MRA for second generation wavelets. For further details, the reader can consult Sweldens (1998).

2.3.1 Second generation multiresolution analysis

Definition 2.3.1. *The* **'second generation' primal multiresolution analysis (MRA) of** $L^2(\mathbb{R})$ *is defined to be a sequence of approximation subspaces,* $(V_j)_{j\in\mathbb{Z}}$ *such that:*

1. *Conditions 1-4 from definition 2.2.1 of the previously introduced (orthogonal) MRA hold,*

2. *The approximation spaces are no longer scaled versions of each other, i.e. condition 5 of definition 2.2.1 is not a requirement anymore,*

3. *Each* V_j *has a basis* $\{\varphi_{j,k}(\cdot)/k \in S_j\}$*, where* $\varphi_{j,k}(\cdot)$ *are scaling functions at level* j *and location* x_k*, and* S_j *is an index set with the property* $S_j \subset S_{j+1}$*.*

The index j still refers to the approximation level— the higher j is, the finer the approximation. However, it should be noted that since V_j and V_{j+1} are not necessarily scaled versions of each other in the sense defined for the classical MRA ($f(\cdot) \in V_j \Leftrightarrow f(2\cdot) \in V_{j+1}$), the level index j is no longer used only on a $\log_2$ scale, and its definition depends on the way the location index spaces S_j are defined with coarser/finer j.

It is important to observe that although the notation x_k in condition 3 of the previous definition does not explicitly state the location dependency on

the level, it corresponds to the scaling points at level j— the finer the level, the finer the sampling. Also note that in the second generation setting the scaling bases have lost their orthonormality, and in this context only endowing V_0 with a basis is not sufficient.

Definition 2.3.2. *A* **dual multiresolution analysis** *of the MRA just introduced, consists of a sequence of spaces $\tilde{V}_j$ with the same properties as the spaces V_j, each endowed with a basis of (dual) scaling functions, $\{\tilde{\varphi}_{j,k}(\cdot)/k \in S_j\}$, with the property*

$$< \varphi_{j,k}, \tilde{\varphi}_{j,k'} >= \delta_{k,k'}, \forall j \in \mathbb{Z}, \forall k, k' \in S_j. \tag{2.22}$$

The scaling functions and their duals are said to be biorthogonal if relation (2.22) holds.

As in the classical MRA, here the scaling functions are also defined iteratively. More exactly, since $\{\varphi_{j,k}(\cdot)\}_{k\in S_j} \subset V_j \subset V_{j+1}$ and $\{\varphi_{j+1,l}(\cdot)\}_{l\in S_{j+1}}$ is a basis for V_{j+1}, it follows that it exists a sequence $\{h_{j,k,l}\}_{j\in\mathbb{Z},k\in S_j,l\in S_{j+1}}$ such that:

$$\varphi_{j,k}(x) = \sum_{l\in S_{j+1}} h_{j,k,l}\varphi_{j+1,l}(x), \forall j \in \mathbb{Z}, \forall k \in S_j. \tag{2.23}$$

This is the equivalent of the familiar scaling or *refinement equation*, and the coefficients $\underline{h} = \{h_{j,k,l}\}_{j\in\mathbb{Z},k\in S_j,l\in S_{j+1}}$ are the filter coefficients. The filters are assumed finite, i.e. the sets $\{l \in S_{j+1}/h_{j,k,l} \neq 0\}$ and $\{k \in S_j/h_{j,k,l} \neq 0\}$ are finite and their sizes are uniformly bounded, $\forall j, k, l$, hence the summation in (2.23) is well defined.

Similarly, dual finite filters exist and the *dual refinement equation* is given by:

$$\tilde{\varphi}_{j,k}(x) = \sum_{l\in S_{j+1}} \tilde{h}_{j,k,l}\tilde{\varphi}_{j+1,l}(x), \forall j \in \mathbb{Z}, \forall k \in S_j. \tag{2.24}$$

Remarks.

1. **Equivalence to dyadic intervals.** In the classical construction, each

$\varphi_{j,k}$ is a scaled (in the $\log_2$ sense) and translated version of φ. This induces a bijective correspondence with a dyadic interval $[2^{-j}k, 2^{-j}(k+1)]$: at each level j, the real line is covered by disjoint dyadic intervals which get finer with larger j. Similarly, in the case of the second generation multiresolution, we also need to partition $\mathbb{R}$. In practical situations, a set of partitionings is constructed based on the set of n initial locations, $\underline{x}$. At the finest level we start with a (disjoint) division of the initial interval into n smaller intervals, each of which contains exactly one initial grid point, x_k. The way of constructing the partition at the next coarser level is directly linked to the proposed way of subsampling the signal. If following the paradigm introduced by Sweldens (1996) of splitting the signal into odds and evens, at the next coarser level we have $[n/2]$ elements. Each of the cells at next coarser level is built such that it contains exactly one of the points on the even positions, x_{2k}. This can be also thought of as fusioning two neighbouring intervals at the finer level, or re-distributing the intervals associated to the removed points. The procedure is then re-iterated until reaching the desired coarse level.

2. **Synthetising the primal and dual scaling functions.** Sweldens (1998) shows that by starting with the index sets $\{S_j\}_{j\in\mathbb{Z}}$, the filter $\underline{h}$ and a set of partitionings, a set of scaling functions $\{\varphi_{j,k}(\cdot)\}_{j\in\mathbb{Z},k\in S_j}$ satisfying (2.23) can be synthetised through a cascade algorithm, provided that the algorithm converges for all j, k. It is worth stressing here that consequently not every filter leads to a set of scaling functions. Similarly, the dual scaling functions $\{\tilde{\varphi}_{j,k}(\cdot)\}_{j,k}$ can be obtained starting from a dual finite filter $\underline{\tilde{h}}$ and the same set of partitionings, again conditional on the convergence of the cascade algorithm $\forall j, k$.

3. **Functions defined on other domains.** The MRA construction in Sweldens (1998) is more general than it appears from our presentation, and it includes square integrable functions defined on any spatial domain

$X \subseteq \mathbb{R}^n$. So a second generation MRA can be naturally formulated for functions defined on intervals, or other bounded domains.

Construction of second generation wavelet bases.

We can now **introduce second generation wavelets**, in a similar manner to first generation wavelets.

First construct a set of spaces $(W_j)_{j\in\mathbb{Z}}$, with W_j the (non-orthogonal) complement of V_j in V_{j+1} and $W_j \perp \tilde{V}_j$. This permits a telescopic decomposition of any approximation space— $V_J = V_{j_0} \oplus (\oplus_{j=j_0}^{J-1} W_j)$, $\forall J \geq j_0 + 1$.

For each j, the signal is approximated in a coarser manner if projected on the space V_j, than if projected on V_{j+1}, and the space W_j contains the 'difference' in detail between the two approximations. This entirely parallels the classical case.

Definition 2.3.3. *We say that the set of functions $\{\psi_{j,m}(\cdot)/j \in \mathbb{Z}, m \in D_j\}$, where $D_j = S_{j+1} \setminus S_j$, is a set of wavelet functions if the set $\{\psi_{j,m}(\cdot)/m \in D_j\}$ forms a basis for W_j, $\forall j \in \mathbb{Z}$.*

In the dual MRA, for each $j \in \mathbb{Z}$, take the space $\tilde{W}_j$ which complements $\tilde{V}_j$ in $\tilde{V}_{j+1}$ such that $\tilde{W}_j \perp V_j$.

Definition 2.3.4. *We say that the set $\{\tilde{\psi}_{j,m}(\cdot)/m \in D_j\}$ is a set of dual wavelets at level j if they form a basis for $\tilde{W}_j$ and are biorthogonal to the primal wavelets, i.e.*

$$< \psi_{j,m}, \tilde{\psi}_{j,m'} >= \delta_{m,m'}, \ \forall j \in \mathbb{Z}, \ \forall m, m' \in D_j. \tag{2.25}$$

The construction of the primal and dual wavelet spaces induces orthogonality of the primal, dual scaling and wavelet functions, respectively. More exactly,

$$< \psi_{j,m}, \tilde{\varphi}_{j,k} >= 0, \quad < \tilde{\psi}_{j,m}, \varphi_{j,k} >= 0, \ \forall j, k, m \tag{2.26}$$

Definition 2.3.5. *If relations (2.22), (2.25) and (2.26) hold, then the primal and dual scaling and wavelet functions are said to be a biorthogonal set of scaling and wavelet functions.*

We now only briefly note the existence of refinement relations for the primal and dual wavelet functions. Because $\psi_{j,m} \in W_j \subset V_{j+1}$ and $\{\varphi_{j+1,l}(\cdot)\}_l$ is a basis for V_{j+1}, there exists the filter $\underline{g} = \{g_{j,m,l}\}_{j\in\mathbb{Z},m\in D_j,l\in S_{j+1}}$ such that the following refinement relation holds

$$\psi_{j,m}(x) = \sum_{l\in S_{j+1}} g_{j,m,l}\varphi_{j+1,l}(x),\ \forall j \in \mathbb{Z},\ \forall m \in D_j. \tag{2.27}$$

For dual wavelets, with dual filter $\underline{\tilde{g}} = \{\tilde{g}_{j,m,l}\}_{j\in\mathbb{Z},m\in D_j,l\in S_{j+1}}$, we have

$$\tilde{\psi}_{j,m}(x) = \sum_{l\in S_{j+1}} \tilde{g}_{j,m,l}\tilde{\varphi}_{j+1,l}(x),\ \forall j \in \mathbb{Z},\ \forall m \in D_j. \tag{2.28}$$

The filters $\underline{g}$ and $\underline{\tilde{g}}$ are also assumed to be finite, and as such the previous sums are well defined.

2.3.2 Function representation on second generation wavelet bases

As bases for $L^2(\mathbb{R})$ we may now use $\{\psi_{j,m}(\cdot)/j \in \mathbb{Z}, m \in D_j\}$ or $\{\varphi_{j_0,k}(\cdot)/k \in S_{j_0}\} \bigcup \{\psi_{j,m}(\cdot)/j \geq j_0, m \in D_j\}$. Due to the biorthogonality of the scaling and wavelet functions, for any $f \in L^2(\mathbb{R})$ we obtain equivalent wavelet decompositions to (2.16), respectively (2.17):

$$f(x) = \sum_{k\in S_{j_0}} < f, \tilde{\varphi}_{j_0,k} > \varphi_{j_0,k}(x) + \sum_{j\geq j_0}\sum_{m\in D_j} < f, \tilde{\psi}_{j,m} > \psi_{j,m}(x), \tag{2.29}$$

$$f(x) = \sum_{j\in\mathbb{Z}}\sum_{m\in D_j} < f, \tilde{\psi}_{j,m} > \psi_{j,m}(x). \tag{2.30}$$

Note that the sparsity of the wavelet decomposition is dictated by the choice of dual, rather than primal wavelets.

Implications of the second generation MRA construction on filters.

Since $< \varphi_{j,k}, \tilde{\varphi}_{j,k'} >= \delta_{k,k'}$, from the refinement relations for the primal and dual scaling functions, it follows that

$$\sum_{l \in S_{j+1}} h_{j,k,l} \tilde{h}_{j,k',l} = \delta_{k,k'}, \ \forall j \in \mathbb{Z}, \ \forall k, k' \in S_j, \tag{2.31}$$

where the filters can be obtained as $h_{j,k,l} =< \varphi_{j,k}, \tilde{\varphi}_{j+1,l} >$ and $\tilde{h}_{j,k,l} =< \tilde{\varphi}_{j,k}, \varphi_{j+1,l} >$.

Also, the biorthogonality of the primal and dual wavelet functions and their corresponding refinement relations give

$$\sum_{l \in S_{j+1}} g_{j,m,l} \tilde{g}_{j,m',l} = \delta_{m,m'}, \ \forall j \in \mathbb{Z}, \ \forall m, m' \in D_j, \tag{2.32}$$

where the filters are given by $g_{j,m,l} =< \psi_{j,m}, \tilde{\varphi}_{j+1,l} >$ and $\tilde{g}_{j,m,l} =< \tilde{\psi}_{j,m}, \varphi_{j+1,l} >$.

The refinement relations for the primal and dual wavelet functions, combined with the construction of the primal and dual wavelet spaces, $V_j \perp \tilde{W}_j$ and $W_j \perp \tilde{V}_j$, generate

$$\sum_{l \in S_{j+1}} h_{j,k,l} \tilde{g}_{j,m,l} = 0, \ \sum_{l \in S_{j+1}} g_{j,m,l} \tilde{h}_{j,k,l} = 0, \quad \forall j, k, m. \tag{2.33}$$

Definition 2.3.6. *If the set of filters* $\{\underline{h}, \underline{\tilde{h}}, \underline{g}, \underline{\tilde{g}}\}$ *satisfies relations (2.31), (2.32) and (2.33), then it is said to be a set of biorthogonal filters.*

Biorthogonality of the filters is equivalent to biorthogonality of the scaling and wavelet functions provided that the cascade algorithm converges both for the primal and dual scaling functions.

2.3.3 The fast wavelet transform

An **equivalent of the discrete wavelet transform** exists for the second generation MRA as well. It also follows from the equivalent approximations of any function $f \in L^2(\mathbb{R})$ on V_{j+1}— using the basis formed only of scaling functions at level $j+1$, or using the basis of scaling and wavelet functions at the next coarser level j.

Denote the scaling coefficients at level j by $c_{j,k} =< f, \tilde{\varphi}_{j,k} >$, with $k \in S_j$, and the wavelet coefficients at the same level by $d_{j,m} =< f, \tilde{\psi}_{j,m} >$, with $m \in D_j$.

The fast wavelet transform gives a recursive way of computing coefficients at level j from the scaling coefficients at the next finer level, $j+1$. Note that at each step *dual* filters are used for decomposing the signal

$$c_{j,k} = \sum_{l \in S_{j+1}} \tilde{h}_{j,k,l} c_{j+1,l}, \; \forall j \in \mathbb{Z}, k \in S_j, \tag{2.34}$$

$$d_{j,m} = \sum_{l \in S_{j+1}} \tilde{g}_{j,m,l} c_{j+1,l}, \; \forall j \in \mathbb{Z}, m \in D_j. \tag{2.35}$$

The fast inverse transform reconstructs the scaling coefficients at a finer level from the scaling and wavelet coefficients at the next coarse level. Note that this uses *primal* filters

$$c_{j+1,l} = \sum_{k \in S_j} h_{j,k,l} c_{j,k} + \sum_{m \in D_j} g_{j,m,l} d_{j,m}. \tag{2.36}$$

At this point it would be helpful to illustrate some of the previously introduced concepts in examples.

Examples of second generation wavelets.

The Lazy wavelet. Let us start with the given set of locations $\underline{x}$ and the index sets $\{S_j\}_j$ (hence $\{D_j\}_j$ is also known). Let the filters $\underline{h}$ be defined by $h_{j,k,l} = \delta_{k,l} \, \forall k \in S_j, l \in S_{j+1}$ and $\underline{g}$ defined by $g_{j,m,l} = \delta_{m,l} \, \forall m \in D_j, l \in S_{j+1}$. Take the dual filters $\underline{\tilde{h}} := \underline{h}$ and $\underline{\tilde{g}} := \underline{g}$.

Starting from these filters, a pair of scaling functions can be synthetised through a cascade algorithm: $\varphi_{j,k}(x) = \mathbb{1}_{\{x_k\}}(x)$, with the Dirac function as their dual, $\tilde{\varphi}_{j,k}(x) = \delta(x - x_k)$. From the refinement relations we can obtain the primal and dual wavelets, $\psi_{j,m} = \varphi_{j+1,m}$, respectively $\tilde{\psi}_{j,m} = \tilde{\varphi}_{j+1,m}$.

By decomposing an initial signal with such filters we do nothing else but split the scaling coefficients into two groups at each level j, $c_{j,k} = c_{j+1,k}$, $\forall k \in S_j$ and $d_{j,m} = c_{j+1,m}$, $\forall m \in D_j$.

Interpolating scaling functions. In practical situations we start with a set of observations $\underline{f}$ on the grid $\underline{x}$. To apply the fast wavelet transform, an initial set $\underline{c}^J$ is needed, where J denotes the finest level at which the data was collected. The usual choice is $c_{J,i} = f(x_i)$, and we approximate f by $\tilde{f}(x) = \sum_i c_{J,i}\varphi_{J,i}(x)$, where $\{\varphi_{J,i}(\cdot)\}_i$ is a set of scaling functions at the finest level. Choosing the scaling functions such that $\varphi_{j,i}(x_k) = \delta_{i,k}$ ensures that $\tilde{f}$ interpolates the values $f(x_i)$. Such scaling functions are known as interpolating, their duals are the Dirac scaling functions (the same as in the case of the Lazy wavelet), and we will talk about their filters later.

2.3.4 Vanishing moments for second generation wavelets

Remember that in the classical construction we introduced wavelet functions with vanishing moments. Having a wavelet function with N vanishing moments in an orthogonal MRA meant that any polynomial of degree at most $N - 1$ can be expressed as a linear combination of integer translations of the scaling function $\{\varphi_{0,k}(\cdot)\}_k$.

Definition 2.3.7. *In the second generation context, we say that a wavelet function has N vanishing moments if there exists a level j^* such that $< x^l, \psi_{j,m}(x) >= 0$, $\forall l \in \overline{0, N-1}, j \geq j^*$.*

Similarly, let us denote by $\tilde{N}$ the number of vanishing moments of the dual wavelet functions.

Assuming $\tilde{N}$ vanishing moments for the dual wavelets, from the decomposition (2.29) on a second generation scaling and wavelet basis we obtain that any polynomial of degree at most $\tilde{N}-1$ can be written as a linear combination of $\{\varphi_{j,k}(\cdot)/k \in S_j\}$, $\forall j \geq j^*$, which defines a primal MRA of order $\tilde{N}$. Symmetrically, the number of vanishing moments of the primal wavelets gives the order of the dual MRA.

This highlights an important aspect of second generation MRA's: the dual functions are the ones that should be chosen to match the signal smoothness, as they determine the sparsity of the decomposition. Also, observe the duality of the construction: the role of the primal and dual MRA's and respectively of the scaling and wavelet functions can be interchanged.

2.3.5 'Lifting' a second generation MRA

At this point, one might wonder what is the connection between the split-predict-update description of the lifting scheme, as given at the beginning of this section, and the second generation MRA just introduced. So far, only the 'split' stage has a correspondent in the MRA frame— applying the Lazy wavelet with the index sets S_j, D_j. What about the predict and update stages? What do they correspond to?

We have already seen that various families of classical wavelets have various properties: if we want to decompose a signal on a smooth wavelet basis, we choose a wavelet from one of the Daubechies families; if we want a compact, symmetrical wavelet and do not require smoothness, we can choose Haar wavelets.

Through the lifting scheme we can design wavelet bases that have several useful properties, which is equivalent to designing a new MRA with some desired properties, such as a higher order. The key to being able to 'custom-design' MRA's and consequently wavelet functions, stays in the versatility of the filters used in the second generation setting: an initial set of filters (that

leads to a MRA) can be 'lifted' by performing some simple operations, in order to obtain a new set of filters that leads to a MRA with better properties. This is equivalent to gradually generating a new MRA satisfying the desired properties, which translates into obtaining a better signal representation in terms of the smooth and detail coefficients. For several examples on possible initial MRA's the reader is directed to Sweldens (1998) (the MRA generated by the Lazy wavelet is a possibility, and so is the one generated by the interpolating scaling functions). We shall see in what follows that each of the prediction and update stages initially presented, refers to a certain type of transform in the set of (initial) filters.

Primal lifting.

Assume we have an initial MRA, hence we start with an initial set of biorthogonal filters $\{\underline{h}, \underline{\tilde{h}}, \underline{g}, \underline{\tilde{g}}\}$.

We will first introduce the primal lifting scheme, and show that this is in fact the 'update' stage previously mentioned.

Definition 2.3.8. *The primal lifting scheme is based on the following alteration of the initial set of biorthogonal filters (where the set* $\underline{b} = \{b_{j,k,m}\}_{j\in\mathbb{Z},k\in S_j,m\in D_j} \in l_2$ *is a finite filter):*

$$h^{new}_{j,k,l} = h_{j,k,l}, \tag{2.37}$$

$$\tilde{h}^{new}_{j,k,l} = \tilde{h}_{j,k,l} + \sum_m b_{j,k,m}\tilde{g}_{j,m,l}, \tag{2.38}$$

$$g^{new}_{j,m,l} = g_{j,m,l} - \sum_k b_{j,k,m}h_{j,k,l}, \tag{2.39}$$

$$\tilde{g}^{new}_{j,m,l} = \tilde{g}_{j,m,l}. \tag{2.40}$$

It can be easily proved that the new set of filters is still biorthogonal, by using the biorthogonality of the initial set, while $\underline{h}, \underline{g}$ and $\underline{b}$ finite ensure the new filters are finite too.

Since the primal filter $\underline{h}$ is untouched, the primal scaling functions do not change. However, the dual filter $\underline{\tilde{h}}$ changes, and consequently so do the dual

scaling functions. Since the dual wavelet functions are obtained through the refinement of the dual scaling functions, they change too. As the filters $\underline{g}$ change, the primal wavelet functions also change, according to the formula

$$\psi_{j,m}^{new}(x) = \psi_{j,m}(x) - \sum_{k \in S_j} b_{j,k,m} \varphi_{j,k}(x). \tag{2.41}$$

The new set of filters generates new primal and dual MRA's (see for example Simoens and Vandewalle (2003))— the new primal MRA has the same scaling functions, while in the dual MRA both scaling and wavelet functions are modified. The new approximation and detail spaces can be defined as the closure of the span of the scaling, respectively wavelet functions.

By choosing a suitable $\{b_{j,k,m}\}_{k,m}$ we can design at each level j new wavelet functions with some desirable properties— such as a certain number of vanishing moments. Overall, this is the same as setting the order of the (new) final dual MRA. Most often, this step is used to preserve the initial average signal, which is equivalent to setting the number of vanishing moments of the primal wavelets to one.

This condition turns out to be a possible (and often used) way of designing the filters $\{b_{j,k,m}\}_{k,m}$: as the (previous) scaling and wavelet functions are known, from equation (2.41), at each level j conditions imposed on each $\psi_{j,m}^{new}$ translate into conditions on $\{b_{j,k,m}\}_k$, $m \in D_j$.

In practice, primal lifting is applied sequentially, at each level j down to the desired coarse level.

In the fast wavelet transform, dual filters are used for decomposing the finer scaling coefficients ($\underline{c}^{j+1}$) into coarser scaling coefficients ($\underline{c}^j$) and detail ($\underline{d}^j$). Since the primal lifting operates a change on $\underline{\tilde{h}}$, but not on $\underline{\tilde{g}}$, only the scaling coefficients at the level at which the change is performed are being modified. An application of the primal lifting can be described as follows

1. decompose $\underline{c}^{j+1}$ using the 'old' filters, $\underline{\tilde{h}}$, $\underline{\tilde{g}}$; obtain $(\underline{c}^j, \underline{d}^j)$,

2. 'lift' the scaling coefficients; the detail coefficients are unchanged

$$c_{j,k}^{new} := c_{j,k} + \sum_{m \in D_j} b_{j,k,m} d_{j,m}, \quad \forall k \in S_j, \tag{2.42}$$

3. for the next step in the wavelet transform, operate using $\underline{c}^j := \underline{c}^{j,new}$, $\underline{d}^j := \underline{d}^j$.

It is now obvious that primal lifting is indeed the 'update' stage as described when we informally introduced it. Note that in order to compute the updated scaling coefficients at level j there is no need to use the new filters, but only the update filter, $\underline{b}$.

Dual lifting.

Definition 2.3.9. *Symmetrically, it is possible to alter the other two filters $(\underline{h}, \underline{\tilde{g}})$ with a similar construction, known as dual lifting:*

$$h_{j,k,l}^{new} = h_{j,k,l} + \sum_{m} a_{j,k,m} g_{j,m,l}, \tag{2.43}$$

$$\tilde{h}_{j,k,l}^{new} = \tilde{h}_{j,k,l}, \tag{2.44}$$

$$g_{j,m,l}^{new} = g_{j,m,l}, \tag{2.45}$$

$$\tilde{g}_{j,m,l}^{new} = \tilde{g}_{j,m,l} - \sum_{k} a_{j,k,m} \tilde{h}_{j,k,l}. \tag{2.46}$$

The new set of filters is still biorthogonal and finite, provided the initial filter set is biorthogonal, finite and $\underline{a}$ is finite. A transform is induced in all scaling and wavelet functions, except for the dual scaling functions.

The name of dual lifting is motivated by the ability of the construction to allow the design of dual wavelets at level j through

$$\tilde{\psi}_{j,m}^{new}(x) = \tilde{\psi}_{j,m}(x) - \sum_{k \in S_j} a_{j,k,m} \tilde{\varphi}_{j,k}(x). \tag{2.47}$$

The dual wavelets will usually be built to satisfy smoothness constraints. In practice, dual lifting is applied at each level of the transform with the overall

purpose of designing a higher order primal MRA.

In the fast wavelet transform, as the only dual filter changed through dual lifting is $\underline{\tilde{g}}$, the scaling coefficients remain unchanged, and only the details are affected by such a filter transform. An application of the dual lifting at level j can be described as follows

1. decompose $\underline{c}^{j+1}$ using the 'old' filters, $\underline{\tilde{h}}$, $\underline{\tilde{g}}$; obtain $(\underline{c}^j, \underline{d}^j)$,

2. 'lift' the detail coefficients; the scaling coefficients are unchanged

$$d_{j,m}^{new} := d_{j,m} - \sum_{k \in S_j} a_{j,k,m} c_{j,k}, \quad \forall m \in D_j, \tag{2.48}$$

3. for the next step in the wavelet transform, operate using $\underline{c}^j := \underline{c}^j$, $\underline{d}^j := \underline{d}^{j,new}$.

This shows that dual lifting is in effect the 'prediction' step (here at level j). Note that only the 'prediction' weights, $\underline{a}$, are used for lifting the detail coefficients.

We previously mentioned interpolating scaling functions and their filters: they can be obtained by applying dual lifting applied to the MRA generated by the Lazy wavelet.

For details on how the scaling and wavelet functions are affected by primal and dual lifting, the reader can consult Sweldens (1998).

Alternating primal and dual lifting.

We have so far seen that through lifting transforms we can (separately) design primal and dual MRA's of higher order. One natural question appears at this point: is there any way in which we can design a MRA that has higher order both in the the primal and dual frame?

The answer is yes, by alternating the primal and dual lifting transforms. The key to successively using primal and dual lifting is that one does not alter the vanishing moments that have been established through the other one. This is easy to see if we think of the number of vanishing moments equivalently in

terms of setting the order of the MRA. Dual lifting for instance sets the order of the primal MRA, and by further using primal lifting, the primal scaling functions do not change. Hence the order of the primal MRA will not be affected, and nor will the number of moments for the (new) dual wavelets.

Alternating primal and dual lifting is known as the 'cakewalk construction', and through it we obtain a MRA with desired properties for both the primal and dual wavelets. In what follows we will see how this construction works in practice.

2.3.6 The fast 'lifted' wavelet transform

A typical application of the lifting algorithm consists in starting with the initial MRA generated by the Lazy wavelet (split the signal at each level j of the transform), and then at each level iteratively apply dual lifting followed by primal lifting. A lifting algorithm with a swap in the prediction and update steps has also been introduced, see Claypoole *et al.* (1998). Both approaches are equivalent to starting with a pair of biorthogonal filters, and then starting from the finest level of the transform, perform a dual, and then a primal change on these filters (or vice versa) at each level of the transform.

As we have already seen, naturally these filter changes induce changes in the basis of scaling and wavelet functions used for function decomposition. Consequently, new scaling and wavelet coefficients are obtained at each step of the transform.

The fast 'lifted' wavelet transform sets the recursive algorithm used for computing the (new) smooth and detail coefficients from the previous (old) ones. This amounts to collating formulas (2.42) and (2.48) in an iterative way. At each step j of the transform

1. **Split.** Split the signal $\{c_{j+1,k}\}_{k\in S_{j+1}}$: obtain $c_{j,k} := c_{j+1,k}$, for $k \in S_j$, $d_{j,m} := c_{j+1,m}$, for $m \in D_j$,

2. **Predict.** Predict the detail coefficients $d_{j,m} := d_{j,m} - \sum_{k\in S_j} a_{j,k,m} c_{j,k}$,

$m \in D_j$,

3. **Update.** Update the scaling coefficients $c_{j,k} := c_{j,k} + \sum_{m \in D_j} b_{j,k,m} d_{j,m}$, $k \in S_j$.

The first step is in fact the application of the direct fast wavelet transform using the Lazy wavelets filters. Note how prediction and update effects interlace: at each level j of the transform, we 'predict' and generate new details, which will be used in the subsequent 'update'; then the (new) updated scaling coefficients will be used in the next 'predict' step, and so on.

The above algorithm can be reversed in a straightforward manner, by reversing the order of the steps and undoing the operations at each level j

1. **Undo update.** $c_{j,k} := c_{j,k} - \sum_{m \in D_j} b_{j,k,m} d_{j,m}$, $k \in S_j$,

2. **Undo predict.** $d_{j,m} := d_{j,m} + \sum_{k \in S_j} a_{j,k,m} c_{j,k}$, $m \in D_j$,

3. **Undo split.** $c_{j+1,k} := c_{j,k}$ for $k \in S_j$, $c_{j+1,m} := d_{j,m}$, for $m \in D_j$.

The last step amounts to applying the fast inverse wavelet transform using the Lazy wavelet filters.

We end this section by pointing out that the 'split' into odds and evens is just one possible choice, which mostly resembles first generation constructions. This split poses problems in higher dimensions, where it is no longer clear what the odds or evens are. Jansen *et al.* (2001) introduces the concept of lifting just one coefficient at each step, which has higher flexibility and can be used in any dimension. We will investigate in detail this type of lifting transform in the next chapter.

So far we have essentially discussed possible ways to represent functions on various wavelet (and some non-wavelet) bases. We pointed out that a desirable feature of the wavelet coefficients is sparsity, which would enable data compression. We now turn towards regression, an important problem in statistics, which we will connect with the concepts previously introduced.

2.4 Nonparametric regression

Let us now put a probabilistic structure on the sequence $f_1, \ldots, f_n$, which we assume to be the noisy observations of an unknown function g, taken at possibly irregularly spaced locations $x_1, \ldots, x_n$. Model this as

$$f_i = g(x_i) + \varepsilon_i, \tag{2.49}$$

where $\varepsilon_i, \ldots, \varepsilon_n$ are random variables that denote the noise, usually assumed to be independently distributed, with zero mean and finite variance, σ^2. The locations $\underline{x}$ are assumed fixed and they usually span the interval [0,1], so $g : [0,1] \longrightarrow \mathbb{R}$. The regression problem consists in estimating g.

We measure the quality of the estimator $\hat{g}$ through its mean integrated square error,

$$\mathrm{MISE}(\hat{g}, g) = E\left[\int_0^1 \{\hat{g}(x) - g(x)\}^2 \, dx\right].$$

Since usually in practice $\hat{g}$ is estimated at the grid points $x_1, \ldots, x_n$, we use the average mean square error (also known as risk) to assess its performance

$$\mathrm{AMSE}(\hat{g}, g) = E\left[n^{-1} \sum_{i=1}^{n} \{\hat{g}(x_i) - g(x_i)\}^2\right].$$

In this section we will address the problem of *nonparametric* regression, i.e. we do not assume any functional form for g to assist us in its estimation. Although both linear and nonlinear techniques have been developed for estimating g, here we only briefly review some linear smoothing methods, and then focus on nonlinear ones, primarily on those involving wavelets.

2.4.1 Linear smoothing methods

A comprehensive treatment of smoothing methods in nonparametric regression can be found in Simonoff (1996). A popular approach consists in using linear smoothers to estimate g, i.e. estimate the value $g(x)$ by using a weighted

average of the noisy data in a window around x

$$\hat{g}(x) = \sum_{i=1}^{n} f_i w_i(x), \tag{2.50}$$

where $w_i(x)$ are weight functions that are non-zero only for those i values such that x_i is 'close' to x. The 'closeness' is defined by the window width, which tunes the smoothness of the resulting function. Its choice is crucial: narrow windows produce wiggly curves, and hence estimates with high variability, while too large windows may oversmooth the data, and consequently produce highly biased estimates. The weights are usually constructed by using scaled and translated transforms of a kernel function— kernel functions have their mass concentrated at 0 and compact support, or rapid decay outside the interval $[-1, 1]$. For a classical example of a kernel estimator see Watson (1964).

Another example of a linear estimator consists in using basis expansions (see Ramsay and Silverman (1997) for a review), where the quality of the estimator $\hat{g}(x) = \sum_{k=1}^{K} c_k \varphi_k(x)$ is tuned by the number K of basis functions $\{\varphi_k(\cdot)\}_k$ in the expansion, and by the chosen type of basis. The choice of basis (polynomial, spline, Fourier, wavelet) should be based on knowledge about the function g properties, such as smoothness, if this is available. The coefficients in the basis expansion can be estimated by using a least squares approach, i.e.

$$\min_{\underline{c}} \sum_{i=1}^{n} \left(f_i - \sum_{k=1}^{K} c_k \varphi_k(x_i) \right)^2. \tag{2.51}$$

A different way of looking at the regression problem is to find a function that ensures both goodness-of-fit to the noisy data and also a certain degree of smoothness. This is known as the smoothing spline approach (see for instance Green and Silverman (1994), Silverman (1985)), and most commonly $\hat{g}$

is obtained as the solution to

$$\min_{g \in C^2} \left(\sum_{i=1}^{n} (f_i - g(x_i))^2 + \lambda \|g''\|_{L^2}^2 \right). \tag{2.52}$$

It can be shown (Green and Silverman (1994)) that the solution to this problem is a cubic spline with knots at $\{x_i\}_{i \in \overline{1,n}}$. The parameter λ controls the balance between the smoothness of $\hat{g}$ and its fidelity to the data (the smaller λ is, less penalty is paid by roughness and we get closer to the usual linear regression context, hence the curve becomes wigglier).

The strength of these methods relies in employing them when estimating a smooth function g, but their performance decreases when g displays discontinuities. For this reason, other estimation methods have been developed.

2.4.2 Nonlinear smoothing using wavelets

Over the last decade, nonlinear smoothing using wavelets has been introduced and became increasingly popular, due to its excellent properties. For excellent reviews the reader can consult Vidakovic (1999), Abramovich *et al.* (2000).

We start by first noting some limitations imposed by the wavelet construction, which force us to make the following assumptions throughout the presentation of wavelet smoothing:

- The number of observations is assumed to be of the form 2^J, for some $J \in \mathbb{Z}$.
- The observation locations are assumed to be regularly spaced, i.e. $x_i = i/n$.
- For each i there is one (and only one) f_i.

It is only fair to stress at this point that the nonparametric regression techniques we presented so far are not as restrictive, they work on any type of grids, of any length and can handle multiple data. However, as we already

pointed out, their performance is limited when estimating functions which, e.g. present discontinuities.

Wavelet shrinkage.

Donoho and Johnstone (1994) introduced the wavelet shrinkage, a nonlinear wavelet-based technique for estimating the true function in the regression problem. Their method can be algorithmically described in a very simple manner, assuming we adhere to the conditions presented in the beginning of section 2.4.2 :

1. Apply the DWT to the data $\underline{f} = \{f_i\}_{i\in\overline{1,n}}$. Assume we stop the decomposition at primary level j_0.

2. Modify the obtained empirical wavelet coefficients, $\{d_{j,k}\}_{j\geq j_0, k\in\mathbb{Z}}$, with the purpose of removing the noise.

3. Invert the DWT by using the untouched scaling coefficients $\{c_{j_0,k}\}_{k\in\mathbb{Z}}$ and the new (modified) wavelet coefficients. The obtained signal is an estimate of g, obtained at the (regularly spaced) grid points $\{x_i\}_{i\in\overline{1,n}}$.

In what follows we will explore the rationale behind this algorithm, mostly concentrating on step 2.

Step 1 is equivalent to transforming the model (2.49) into $W\underline{f} = W\underline{g} + W\underline{\varepsilon}$, where $\underline{g} = \{g(x_i)\}_{i\in\overline{1,n}}$, $\underline{\varepsilon} = \{\varepsilon_i\}_{i\in\overline{1,n}}$ and W is the matrix associated to the DWT. Equivalently, this can be written as

$$\underline{d} = \underline{d}^* + \underline{e}, \tag{2.53}$$

where $\underline{d}$ ($\underline{d}^*$, respectively $\underline{e}$) denotes the DWT of $\underline{f}$ ($\underline{g}$, respectively $\underline{\varepsilon}$). The choice of coarsest level j_0 was shown to influence the performance of the final estimator (see for example Abramovich and Benjamini (1995), Hall and Nason (1997)), with small values for j_0 suitable for smooth signals, and large values for discontinuous ones. Hall and Nason (1997) suggest choosing j_0 on a continuous scale, using cross-validation.

In step 2 we obtain an estimate of $\underline{d}^*$, denote it by $\hat{\underline{d}}^*$, by 'somehow' removing the noise. The logic behind 'somehow' is unveiled by considering some fundamental consequences of the orthogonality of the DWT transform:

- If $\underline{\varepsilon}$ is independent Gaussian noise, then the DWT maps it into independent Gaussian noise, $\underline{e}$, with the same variance σ^2. Hence, from representation (2.53), all the detail coefficients $\underline{d}$ are equally contaminated by noise, and the regression problem (2.49) can now be formulated for the wavelet coefficients: estimate $\underline{d}^*$ using $\underline{d}$, under the assumption that $\underline{e}$ is independent, Gaussian noise.

- As many (noiseless) functions have sparse wavelet representations, in the new regression (2.53) we have in fact an important piece of supplementary information. Sparsity of the (true) details $\underline{d}^*$ means that most of them will be zero, with a few large ones that essentially represent the signal g. In this scenario, small values of $\underline{d}$ represent the noise. In a nutshell, the idea of Donoho and Johnstone (1994) consists of estimating the true details $\underline{d}^*$ by using a *thresholding scheme* that sets the small *observed* wavelet coefficients to zero, and keeps or shrinks the large ones. Setting the level of the *threshold* which separates true signal from noise is of course essential, as a threshold that is too high can effectively eliminate features of the signal, while a too low one can allow for noise to pass in the reconstruction.

Once the details have been estimated, through step 3 we obtain an estimate of the true function g at locations $\underline{x}$, $\hat{\underline{g}} = W^T \hat{\underline{d}}^*$. Orthogonality of the matrix W (i.e. $WW^T = I_n$) implies that Parseval's relation holds: $\|\hat{\underline{g}} - \underline{g}\|_{l^2}^2 = \|W^T(\hat{\underline{d}}^* - \underline{d}^*)\|_{l^2}^2 = \|\hat{\underline{d}}^* - \underline{d}^*\|_{l^2}^2$, which in its turn means that the estimators in the time and wavelet domains have the same risk.

We will now review some of the most popular thresholding rules and thresholds, which are essential in completing step 2 of the wavelet shrinkage procedure.

Possible thresholding schemes and thresholds.

Thresholding schemes. In their paper, Donoho and Johnstone (1994) propose two thresholding schemes, known as *hard* and *soft thresholding*

$$\eta_H(d_{j,k},\lambda) = d_{j,k}\mathbb{1}_{(-\infty,-\lambda)\bigcup(\lambda,\infty)}(d_{j,k}), \tag{2.54}$$

$$\eta_S(d_{j,k},\lambda) = \mathrm{sgn}(d_{j,k})\max(0,|d_{j,k}|-\lambda), \tag{2.55}$$

where λ is a set threshold.

Through the hard thresholding scheme, the details above (in absolute value) the threshold λ are kept, and all the others are set to zero ('killed'). Soft thresholding shrinks by λ the details above the threshold (in absolute value), while setting to zero the rest of them. By using one of these thresholding rules with a set threshold λ, one obtains estimated details $\underline{\hat{d}}^*$, and the DWT can then be inverted as described in step 3. It has been noted in the literature that hard threshoding is better at reproducing discontinuities in the signal, while soft thresholding produces visually smoother results (the function estimate has smaller variance), but it has larger bias (e.g. Johnstone and Silverman (1997)).

Let us now review some of the most popular thresholds.

Universal threshold. In the same paper, Donoho and Johnstone (1994) introduced the famous *universal threshold*, $\lambda = \sigma\sqrt{2\log n}$. This was motivated by the following result

$$\lim_{n\to\infty} P(\{\max_i|\varepsilon_i| > \sigma\sqrt{2\log n}\}) = 0,$$

where $\{\varepsilon_i\}_{i\in\overline{1,n}}$ are independent random variables, distributed as $N(0,\sigma^2)$. If a $d_{j,k}$ is above $\sigma\sqrt{2\log n}$, then most likely its corresponding $d^*_{j,k}$ is not zero, and that coefficient contains signal. Note that the universal threshold is only related to the data through σ.

With high probability, both hard and soft thresholding with the universal threshold, set to zero the observed wavelet coefficients which are entirely due

to noise (Donoho and Johnstone (1994)). It has been noted though, e.g. Nason (1996), that in doing so, some features of g are usually lost and the estimate obtained in practice is oversmoothed.

The estimator produced by using the soft thresholding rule with the universal threshold is known under the name of *VisuShrink*, and its risk was proved (Donoho and Johnstone (1994)) to be close to the ideal one. The ideal (or oracular) risk is the risk that would be attained had we known which empirical wavelet coefficients we should keep or kill based on the instructions of an oracle.

SURE threshold. Another popular choice of threshold was introduced by Donoho and Johnstone (1995), and is based on a result proved by Stein (1981) on estimating the mean of a multivariate normal distribution. In his paper, Stein introduced a way to obtain an unbiased estimator for the average mean square error incurred by the use of a certain estimate for the mean of a multivariate normal distribution.

More exactly, assume we have an n dimensional vector of observations $\underline{d} \sim N_n(\underline{d}^*, \Sigma)$ based on which we want to obtain an estimate for $\underline{d}^*$. If an estimator of the form $\hat{\underline{d}}^* = \underline{d} + h(\underline{d})$ is constructed, where $h : \mathbb{R}^n \to \mathbb{R}^n$ is a weakly differentiable function, then Stein proved that the average mean square error (over all $\underline{d}^*$) generated by the use of this estimator can be unbiasedly estimated by

$$S(h, \underline{d}) = \text{Trace}(\Sigma) + \|h(\underline{d})\|_{l^2}^2 + 2\text{Trace}(\Sigma\, Dh(\underline{d})), \tag{2.56}$$

where $Dh(\underline{d})$ is an $n \times n$ matrix with entries $\partial h_{j,k}(\underline{d})/\partial d_{j,k}$.

The result above states in fact that $E_{\underline{d}^*}(\|\hat{\underline{d}}^* - \underline{d}^*\|_{l^2}) = E(S(h, \underline{d}))$.

When using the soft thresholding rule (2.55) (with a set λ) to obtain an estimate for the true wavelet coefficients, we are in effect in the above situation

with the function h given by $h(d_{j,k}) = -\text{sgn}(d_{j,k})\min(|d_{j,k}|, \lambda)$. Consequently,

$$S(\lambda, \underline{d}) = n\sigma^2 + \sum_{j,k} \min(d_{j,k}^2, \lambda^2) - 2\sigma^2 \sum_{j,k} \mathbb{1}_{[-\lambda,\lambda]}(d_{j,k}). \tag{2.57}$$

This justifies the threshold choice

$$\lambda = \text{argmin}_{0 \le \lambda \le \sigma\sqrt{2\log n}} S(\lambda, \underline{d}). \tag{2.58}$$

The threshold given by (2.58) is known in the field as the *SURE threshold* (*S*tein's *U*nbiased *R*isk *E*stimator), and it usually yields smaller values than the universal one. However, it exhibits problems when the wavelet coefficients are very sparse, and consequently Donoho and Johnstone (1995) proposed a hybrid scheme for obtaining an estimator $\hat{g}$, named by its authors *SureShrink*.

SureShrink is the estimator obtained by performing soft thresholding within each scale of the wavelet coefficients, and using the universal threshold when the data at that level is sparse or a scale-dependent SURE threshold otherwise. Donoho and Johnstone (1995) have shown that SureShrink is asymptotically minimax over a range of Besov spaces. We say that an estimator $\hat{g}^*$ is minimax in a certain function class $\mathscr{F}$ (such as the smoothness spaces Hölder or Besov) if $\sup_{g \in \mathscr{F}} \text{AMSE}(\hat{g}^*, g)$ coincides with the minimax risk, $\inf_{\hat{g}} \sup_{g \in \mathscr{F}} \text{AMSE}(\hat{g}, g)$.

Note that the above theory has been developed for noise assumed independent and to follow a normal distribution. When the noise is stationary and correlated, Johnstone and Silverman (1997) show that the wavelet transform essentially decorrelates the data, and the variance of the details depends on their scale, but not on location. Therefore the authors propose using a scale-dependent threshold within the overall usage of the same thresholding scheme.

Estimating the noise variance. In practice the standard deviation of the noise is almost never known and needs to be estimated. For Gaussian noise, Donoho and Johnstone (1994) proposed using the sparsity of the true details

at the finest scale, which ensures that the observed details at the finest scale mostly represent the noise. A robust estimate for σ is given by the median of the absolute values of the finest observed details, divided by 0.6745, i.e.

$$\hat{\sigma} = \text{median}(|d_{J-1,k}|/k \in \overline{0, 2^{J-1}})/0.6745. \tag{2.59}$$

The above quantity is known as median absolute deviation from zero, in short 'mad'.

Other shrinkage approaches.

Nason (1996) proposed using a twofold ('leave-half-out') cross-validation technique for selecting the threshold subject to minimising the mean square error. Nason (2002) points out that the threshold choice is just one of the parameters that heavily influence the smoothing performance, and therefore addresses the problem of choosing simultaneously the wavelet smoothness and primary level in the DWT to minimise the average mean square error. A cross-validation approach is proposed to yield good combinations of threshold, number of vanishing moments and primary level (most often there is no unique solution). This technique uses the development of Kovac and Silverman (2000), and as such it is capable of working with irregular data sets of any length. Other cross-validation approaches have been developed, for a comprehensive review see Vidakovic (1999).

Thresholding has also been looked at as a multiple hypothesis testing problem (i.e. test $H_0 : \{d^*_{j,k} = 0\}$ versus the alternative $H_1 : \{d^*_{j,k} \neq 0\}$ and retain in the model only the coefficients for which H_0 has been rejected), and such approaches can be found in Abramovich and Benjamini (1995) and Ogden and Parzen (1996a,b). Since a large number of hypotheses is being tested simultaneously, there is an issue as how to control the error— classical approaches either control it individually or simultaneously, with a Bonferroni type correction. Abramovich and Benjamini (1995) take a 'false discovery rate' (FDR) approach to this problem. The FDR of coefficients is defined as the expected

value of the proportion (Q) of the coefficients erroneously kept in the representation, and the proposal of Abramovich and Benjamini (1995) is to maximize the number of wavelet coefficients kept in the model, subject to controlling the FDR ($E(Q) \leq \alpha$, where α is the significance level of the test). In Ogden and Parzen (1996a,b), at every scale the wavelet coefficients are (statistically) tested to determine whether significant signal is present or they are due to noise. This enables separating the details within each level into a set of 'large' coefficients considered to contain signal, and a set of 'small' ones, considered to be due to the noise. The 'small' coefficients are then 'killed', and the large ones retained. In obtaining this split, statistical tests that take into account the magnitude of the coefficients as well as their location are employed. The significance level α of the tests tunes the final smoothness of the estimate (the larger the α, the wigglier the estimate).

Thresholding and shrinkage techniques have also been built under a Bayesian framework, and they have proved to work well. In this approach, prior distributions are imposed on the true wavelet coefficients, and based on the observed details (assumed to follow a normal distribution), the corresponding posterior distribuitions are obtained. The true details are then estimated using a chosen Bayesian rule. Typically, the posterior mean has been the usual choice in the literature, and we mention here that this is in fact a shrinkage, and not a thresholding rule. Abramovich *et al.* (1998) introduced the use of posterior medians in wavelet thresholding, and showed that this corresponds to obtaining a thresholding rule. Having estimated the true details, the DWT transform is inverted, and an estimate for g is obtained.

The prior distributions are designed to capture the sparsity property of the true details, and therefore they are usually mixtures of a point mass and some other distribution— the normal distribution is used in Abramovich *et al.* (1998), the 'quasi-Cauchy' or Laplace distributions are proposed in Johnstone and Silverman (2004a, 2005)— although mixtures of normal distributions have also been used, see for example Chipman *et al.* (1997). Section 3.6.2 contains

a description of the empirical Bayesian method of Johnstone and Silverman (2004a, 2005). A comprehensive review on Bayesian wavelet techniques used for nonparametric regression appears in Vidakovic (1999).

Chapter 3

Adaptive Lifting

In this chapter we will introduce and investigate a novel adaptive lifting construction. The proposed method is the result of joint work with Matthew Nunes and Guy Nason, and a description can be found in Nunes *et al.* (2004).

3.1 Motivation

Let us start with the nonparametric regression problem (see section 2.4 of the previous chapter) of estimating a true function (g) from n noisy observations, $f_1, \ldots, f_n$, observed at locations $x_1, \ldots, x_n$. We have already seen that wavelet thresholding/shrinkage methods have been developed and shown to perform well in the task of estimating g, particularly when g presents discontinuities. One of the major ingredients of wavelet shrinkage is to transform the initial signal into a set of scaling and wavelet coefficients by means of a DWT. However, classical wavelet constructions have their limitations, and consequently the application of many wavelet shrinkage methods is based on the following assumptions

1. The grid locations, $x_1, \ldots, x_n$, are assumed to be equally spaced— usually $x_i = i/n$, for $i \in \overline{1, n}$,

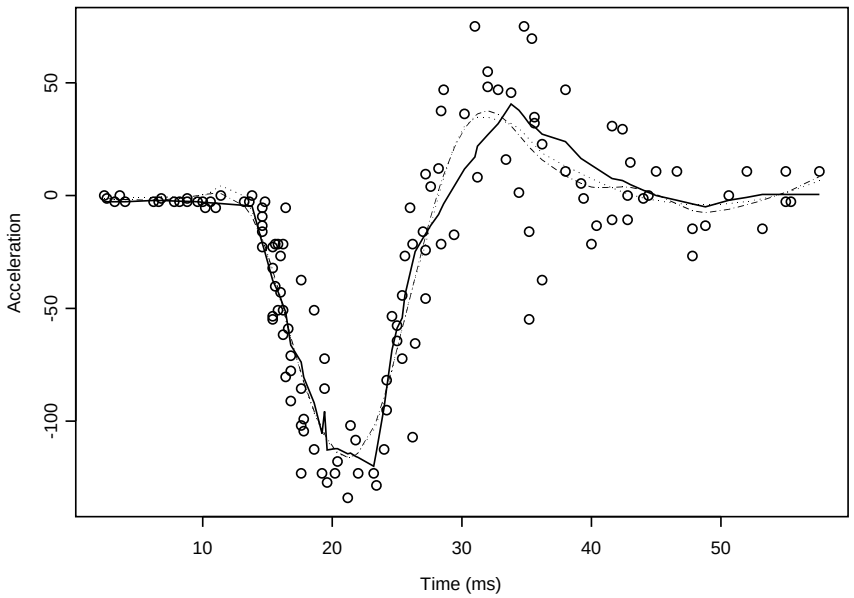

Figure 3.1: Experiment to determine efficacy of crash helmets. Plot showing 133 samples of motorcyclist's head acceleration in simulated motorcycle crashes versus time after impacts (points). The solid curve is the denoised version using our proposed method AP1S (see sections 3.6.1, 4.3.1); dotted, dot-dashed curves are the estimates using *Locfit*, `smooth.spline` (see section 4.3.1).

2. The number of observations, n, is assumed to be of the form 2^J for some $J \in \mathbb{Z}$,

3. There is one (and only one) corresponding observation f_i for each location x_i.

Further questions appear at this point, which we have already pointed out in the previous chapter: which wavelet (what smoothness) is most suitable for decomposing this particular dataset? Which primary resolution level is best?

The reality is that many real data sets do not satisfy assumptions 1–3, nor come with additional pieces of information such as the local degree of regularity of the underlying function. Such an example is the well-known `mcycle` motorcycle data (see figure 3.1) from Silverman (1985), which consists of 133 samples at 94 distinct irregularly sampled time points, and which therefore fails all of the assumptions 1–3. We will provide a detailed analysis of this dataset in the next chapter.

The wavelet literature has of course tried to provide answers to these limitations. Several methods have been proposed to adapt the usage of classical wavelet functions to irregularly spaced data, and we will review some them in section 3.2. Concerning the choice of wavelet, the general rule of thumb is to use a wavelet of a smoothness at least as high as the one of g. Nason (2002) suggests using a cross-validation approach to obtain both the wavelet smoothness and the primary resolution level that minimise the average mean square error of the estimate. These techniques are useful for situations when the underlying function possesses a constant degree of smoothness, which is not always true.

In what follows we propose an adaptive second generation wavelet construction, stemming from the lifting scheme paradigm, which automatically circumvents assumptions 1 and 2. We will base our construction on the lifting scheme which 'removes one coefficient at a time' introduced by Jansen *et al.* (2001, 2004), which we will present in detail in section 3.3. In our approach we shall exploit the flexibility of lifting and generate wavelet functions that *locally* adjust to the signal features. As a consequence, the smoothness of each wavelet function gets tuned *adaptively* and *automatically* to the varying smoothness of the given data. We introduce two degrees of adaptivity, resulting in *two adaptive lifting algorithms*, described in section 3.4. Our approach allows for a natural way of handling multiple data (situations that depart from assumption 3), and as such the only remaining issue will be an equivalent of the choice of primary resolution level. In section 3.5 we analyse the implications of using adaptiveness in the lifting scheme algorithm.

When using wavelet shrinkage methods, the noise $\varepsilon_1, \ldots, \varepsilon_n$ present in the function observations is also subjected to some assumptions— it is usually modelled as independent, identically distributed as $N(0, \sigma^2)$ random variables. Through our development we do not intend to alleviate any of these assumptions, and we will thoroughly investigate the implications of our construction on the thresholding procedure. We found that modified versions of the empir-

ical Bayesian denoising technique of Johnstone and Silverman (2004a, 2005) work well for shrinkage with our adaptive lifting transforms— in section 3.6 we will present the empirical Bayes procedure and then thoroughly analyse its implications in the context of adaptive lifting.

We will demonstrate empirically that our proposed adaptive transforms produce sparse wavelet representations and have competitive denoising properties for irregularly spaced datasets. In the next chapter, we will illustrate our methods on real data and analyse in detail the results of an extensive simulation study, including comparisons with the established wavelet and non-wavelet techniques.

3.2 Adapting classical wavelet methods to irregular designs

As we have seen before, a downside of the (classical) wavelet construction is that it can only deal with data collected on regular grids. In the context of nonparametric regression, various approaches have been proposed to transform irregularly spaced data to the equally-spaced design of the DWT. In what follows we briefly review some of these methods, and discuss their advantages and disadvantages.

Cai and Brown (1998) proposed taking into account the irregularity by using the correspondence $x_i = H^{-1}(i/n)$ between the irregular grid $\{x_i\}_{i\in\overline{1,n}}$ and a regular one, $\{i/n\}_{i\in\overline{1,n}}$. Here H is a strictly increasing function which usually needs to be estimated. This construction allows the mapping of the (unknown) function g collected on the irregular grid into the function $g \circ H^{-1}$ collected on a regular grid, and similarly for their noisy versions. Hence classical wavelet methodology can be applied for this new function, which can therefore be estimated. By composing this estimator with H, an estimator of g can be obtained. When g is much smoother than H this proves not to be a good

estimator, so for piecewise Hölder functions, an estimator of g is constructed based on the wavelet decomposition of $\text{Proj}_{V_J} n^{-1/2} \sum_{i=1}^{n} f_i \varphi_{J,i}^{per}(x)$. Then a new threshold is obtained by a generalization of the Donoho and Johnstone (1994) VisuShrink. The final estimator is within a logarithmic factor of the minimax risk over a wide range of piecewise Hölder classes.

Sardy *et al.* (1999) proposed four ways (comparable in performance) of extending the Haar wavelet transform to irregularly spaced data, and then adapted VisuShrink to the modified transform. Out of the proposed transforms, the isometric Haar wavelets are computationally simplest, and the authors point out that they can be generalized to wavelets of higher order than Haar.

In another approach, Cai and Brown (1999) showed that when the x_i are distributed independently, uniformly on [0,1], the wavelet method with universal threshold can be applied directly, as if the grid were regular. The motivation is the use of the approximation $x_{(i)} \sim E(x_{(i)}) = i/(n+1)$, and so the observations $(i/(n+1), f_i)$ are considered instead of (x_i, f_i). This estimator is also within a logarithmic factor of the minimax risk over a range of Hölder functions.

Kovac and Silverman (2000) mapped data sampled on an irregularly spaced grid to data 'collected' on a regular grid, by linearly transforming the original (noisy) function values $\underline{f}$ into new values $\underline{\tilde{f}} = R\underline{f}$. The matrix R describes the interpolation procedure. This transform allows taking into account the correlational structure of $\underline{\tilde{f}}$ by using $\text{Var}(\underline{\tilde{f}}) = R\text{Var}(\underline{f})R^T$. The usual nonparametric regression strategy can now be handled by using the DWT of $\underline{\tilde{f}}$, and then thresholding the wavelet coefficients by making use of the variance-covariance matrix of $\underline{\tilde{f}}$. To simultaneously handle the choice of wavelet, primary resolution level and threshold for estimating the true function, Nason (2002) developed a fast cross-validation algorithm, able to work on irregular grids using the Kovac-Silverman (KS) procedure.

Antoniadis and Fan (2001) approached the nonparametric regression by

formulating a penalized least squares problem in terms of the unknown wavelet coefficients of $\tilde{g} = \{g(i/n)\}_{i \in \overline{1,n}}$. They initially assumed a regular grid of length of the form $n = 2^J$, and imposed certain conditions on the penalty function. Under these restrictions, they proved the existence and uniqueness of a solution to their penalized least squares problem. The final estimators were produced by using a new universal threshold, that generated a smaller risk than the classical universal threshold. The procedure was extended to irregular data by constructing non-linear regularized Sobolev interpolators as estimators and then improving them by constructing a regularized one-step estimator.

Pensky and Vidakovic (2001) tackled the problem of nonparametric regression with data collected on any type of grid $\underline{x}$, by placing a probabilistic model on the x_i's, considered to be generated from an unknown distribution with density function say h, to be estimated. The regression function $E(f|X = x)$ is estimated by its projection on the space V_J of a multiresolution analysis, i.e. $\sum_k \hat{c}_{J,k}\varphi_{J,k}(x)$, where $\hat{c}_{J,k}$ is an estimator of $c_{J,k}$ based on $\varphi_{J,k}$ and the estimator of h. The final estimator has good properties, provided that h is reasonably smooth.

All the above methods enable wavelet shrinkage to be carried out for irregularly spaced data. However, some of them assume models for the grid values that either might not apply (e.g. uniformly distributed x_i), or require the estimation of additional quantities (e.g. the function H or the density h) that might be unreliable for small sample sizes. For interpolation methods choices need to be made, such as location and spacing of the regular grid or interpolation method, which will influence performance. For some of the other methods, unreasonable assumptions on the smoothness of g are made which might not hold in practice.

On the other hand, the lifting scheme provides a natural frame for handling irregularity in the most general data situations. Since our proposed adaptive lifting transforms are built on the lifting algorithm introduced by Jansen *et al.* (2001, 2004) we review their construction next.

3.3 MRA setting when lifting 'one coefficient at a time'

Suppose we have the values $f_1, \ldots, f_n$ of a function f, sampled at n irregularly spaced points $x_1, \ldots, x_n$ on the real line. We aim at transforming the sampled values $f_1, \ldots, f_n$ by means of lifting into a set of scaling and wavelet coefficients.

As we work on the real line, the x-values can be ordered, and to each point we will associate an interval. A possible way of doing this is by constructing intervals which have the endpoints at the midpoints between the initial grid points. Since the policy is to remove at each step one point from the set of corresponding scaling points, at each coarser level one of the intervals from the previous (finer) level will be redistributed. Once the criterion for choosing the point to be removed at each step is set, this construction defines in fact a partition of the initial interval.

Let us start with (primal) scaling functions defined to be the characteristic functions of the intervals associated to each point, hence at the finest level we have $\varphi_{n,k}(x_i) = \delta_{i,k}$, for $k, i \in \overline{1,n}$. Approximate the initial function f by

$$\tilde{f}(x) = \sum_{k=1}^{n} c_{n,k}\varphi_{n,k}(x), \tag{3.1}$$

where $c_{n,i} := f(x_i)$. From its construction $\tilde{f}$ interpolates the initial values $\{f_i\}_{i\in\overline{1,n}}$, as $\tilde{f}(x_i) = \sum_{k=1}^{n} c_{n,k}\delta_{i,k} = c_{n,i} = f(x_i)$.

A wavelet coefficient will be produced at each step of the lifting transform, and a criterion for selecting its location must be chosen. We shall refer to the initial stage where the n observations are collected by phase n. At this step the set of indices corresponding to the scaling coefficients is $S_n = \{1, \ldots, n\}$, and the set of 'wavelet' indices is empty, $D_n = \emptyset$.

At the next step, $n-1$, we choose a point to be lifted and denote its index by j_n. This will be the point that will be removed from the current set of scaling coefficients and 'converted' into a detail coefficient. The new set

of indices corresponding to the scaling coefficients is $S_{n-1} = S_n \setminus \{j_n\}$, while $D_{n-1} = \{j_n\}$ is the set of 'wavelet' indices constructed at this stage.

The point to be lifted, (x_{j_n}, c_{n,j_n}), is chosen such that $\int \varphi_{n,j_n}(x)\,dx = \min_{k\in\overline{1,n}} \int \varphi_{n,k}(x)\,dx$. Intuitively, by choosing the point with the smallest scaling function integral, we choose the point with the finest detail. The integral of a scaling function accounts for the interval length 'spanned' by that point, and therefore smaller integral values correspond to regions where the function has been densely sampled. The removal of such a point will only cause small information loss in the signal.

Once the point to be removed is chosen, we identify a set of its neighbours, I_n. Since there is a one-to-one correspondence between the point to be removed and its removal stage, we index each set of neighbours only by the stage. We use the neighbours to predict the function value at j_n by using simple regression techniques over its neighbourhood. As both neighbourhood definition and choice of regression method are essential to our installation of adaptivity, we defer their description to the next section.

The *prediction phase* yields a linear estimate in the predictors of the form $\sum_{i\in I_n} a_i^n c_{n,i}$, where a^n are the weights resulting from the chosen regression procedure over the neighbourhood. If j_n has only one neighbour, i, then the prediction is $c_{n,i}$. The detail coefficient will be obtained from

$$d_{j_n} := c_{n,j_n} - \sum_{i\in I_n} a_i^n c_{n,i}, \tag{3.2}$$

or in the one neighbour case,

$$d_{j_n} := c_{n,j_n} - c_{n,i}. \tag{3.3}$$

The *update phase* only affects the scaling coefficients of the neighbouring points

$$c_{n-1,i} := c_{n,i} + b_i^n d_{j_n},\ \forall i \in I_n, i \neq j_n. \tag{3.4}$$

For any $i \notin I_n$ ($i \neq j_n$) the scaling coefficients are unaffected, $c_{n-1,i} := c_{n,i}$.

When we initially introduced the lifting scheme in the context of a second generation MRA (see section 2.3.5 of the previous chapter), we pointed out that with each lifting step we build a new basis of (primal and dual) scaling and wavelet functions. So through the above steps, equivalent of a cakewalk construction, 'in the background' we construct new scaling and wavelet functions at each step. Therefore, due to the prediction step (dual lifting) the integrals of the new scaling functions change as well, and need to be updated.

Let us denote $I_{n,i} = \int \varphi_{n,i}(x)\, dx$ for future use (not to be confused with the set of neighbours I_n of the point indexed by j_n). The change in integrals is to account for the decreasing number of scaling points that remain to 'span' the same interval. Only the integrals corresponding to neighbours are affected

$$I_{n-1,i} = I_{n,i} + a_i^n I_{n,j_n},\ i \in I_n, \tag{3.5}$$

$$I_{n-1,i} = I_{n,i},\ i \notin I_n. \tag{3.6}$$

We embed the 'update' of integrals of the scaling functions located at the neighbouring points in the update stage, although in fact the prediction stage is responsible for their change.

The aim of the update stage is to keep $\sum_{i \in S_n} c_{n,i} I_{n,i}$ constant across scales. In other words

$$\sum_{i \in I_n} c_{n,i} I_{n,i} + c_{n,j_n} I_{n,j_n} = \sum_{i \in I_n} c_{n-1,i} I_{n-1,i}. \tag{3.7}$$

Re-writing this constraint by using the previous equations, reduces it at

$$I_{n,j_n} = \sum_{i \in I_n} b_i^n I_{n-1,i}, \tag{3.8}$$

and we will see that this is in fact equivalent to requiring the newly constructed wavelet function to have one vanishing moment. Moreover, from this require-

ment, the weights b^n will be obtained. Jansen *et al.* (2001, 2004) recommend using the minimum norm solution $b_i^n = I_{n,j_n} I_{n-1,i} / \sum_{k \in I_n} I_{n-1,k}^2$ to (3.8) for numerical stability reasons.

After obtaining the detail coefficient and updating the corresponding scaling ones, the point j_n will be removed.

Reiterate then the lifting transform, and at stage $r-1$ we start with $r = \text{card}(S_r)$ scaling coefficients (where $\text{card}(S_r)$ denotes the number of elements in the set S_r), construct the detail coefficient on position j_r, update the values of the scaling coefficients and the corresponding integrals on the neighbouring positions (with indices in I_r) and remove the index j_r from S_r— hence construct new sets $S_{r-1} = S_r \setminus \{j_r\}$ and $D_{r-1} = \{j_r\}$. The choice of index j_r is subject to minimising the integrals of the scaling functions at level r.

So at stage $r-1$, the lifting steps can be summarised as follows:

- choose a point to be removed, j_r from the set S_r,
- *predict* its function value by using regression over a chosen neighbourhood and generate the detail coefficient $d_{j_r} := c_{r,j_r} - \sum_{i \in I_r} a_i^r c_{r,i}$,
- *update* the neighbouring scaling coefficients $c_{r-1,i} := c_{r,i} + b_i^r d_{j_r}$ for any $i \in I_r$, $i \neq j_r$ and $c_{r-1,i} := c_{r,i}$ for any $i \notin I_r$, $i \neq j_r$ and update their associated integrals.

The remaining set of scaling coefficients to be used at the next stage $(r-2)$ has indices in S_{r-1}; reiterate the algorithm until the desired coarse level is reached.

The new filter coefficients, $\{h_{r-1,i,l}^{new}\}_{i \in S_{r-1}, l \in S_r}$, induced by the lifting construction after removing $n-r+1$ grid points $(j_n, j_{n-1}, \ldots, j_r)$ have the form

$$h_{r-1,i,l}^{new} = \delta_{i,l}, \ \forall l \in S_{r-1}, \tag{3.9}$$

$$h_{r-1,i,l}^{new} = a_i^r \mathbb{1}_{I_r}(i), \ l = j_r. \tag{3.10}$$

The (high-pass) filters $\{g^{new}_{r-1,j_r,l}\}_{l\in S_r}$ are given by

$$g^{new}_{r-1,j_r,l} = -b^r_l,\ \forall l \in I_r, \tag{3.11}$$

$$g^{new}_{r-1,j_r,l} = 1 - \sum_{i\in I_r} a^r_i b^r_i,\ l = j_r, \tag{3.12}$$

$$g^{new}_{r-1,j_r,l} = 0, \forall l \in S_{r-1} \setminus I_r. \tag{3.13}$$

A simple application of the refinement relations (2.23) and (2.27) reveals the progressive construction of the scaling and wavelet functions

$$\varphi_{r-1,i}(x) = \varphi_{r,i}(x) + a^r_i \varphi_{r,j_r}(x),\ i \in I_r, \tag{3.14}$$

$$\varphi_{r-1,i}(x) = \varphi_{r,i}(x),\ i \notin I_r, \tag{3.15}$$

$$\psi_{j_r}(x) = \varphi_{r,j_r}(x) - \sum_{i\in Ir} b^r_i \varphi_{r-1,i}(x). \tag{3.16}$$

As the algorithm proceeds, at each step the scaling and wavelet functions are recursively constructed from the scaling functions at the coarser level; through the prediction and update weights, as well as through the choice of point to be lifted, the scaling and wavelet functions depend on the initial grid locations, $\underline{x}$. However, for the application of the lifting algorithm itself, we do not need to explicitly construct the scaling functions at each step, but merely start with initial values for their corresponding integrals, which then need to be updated at every step throughout the algorithm.

From relation (2.29), after $j_n, j_{n-1}, \ldots, j_r$ have been removed, the initial function f can be represented as

$$f(x) = \sum_{i\in S_{r-1}} c_{r-1,i}\varphi_{r-1,i}(x) + \sum_{k\in\{n,n-1,\ldots,r\}} d_{j_k}\psi_{j_k}(x) \tag{3.17}$$

In the classical wavelet approach, the scaling and wavelet functions correspond to different scales (defined dyadically) and locations (translations); within the same scale, the support length of the scaling, respectively wavelet functions is the same. Consequently, when building the (classical) smooth and

detail coefficients they will correspond to a certain scale and location.

In the second generation wavelet approach, although the signal is still represented on a fine–coarse range, the scaling and wavelet functions at the same stage/level, say $r-1$, are not scaled versions of the same function. This is due to the second generation wavelet constructions being driven by the irregularities in the data locations. Lifting 'one coefficient at a time' highlights this aspect even more by introducing just one wavelet function at each stage. The notion of scale itself is not obvious in this approach. In our development we will use the same definition for scale as suggested in Jansen *et al.* (2004)—define the scale of the wavelet function ψ_{j_r} to be I_{r,j_r}, the length of the interval associated to the location indexed by j_r at the last stage prior to its removal. The irregularity in locations is therefore reflected in the scale, which becomes a continuous quantity.

In the dual frame, the dual scaling and wavelet functions can also be obtained recursively, as follows:

$$\begin{aligned}
\tilde{\varphi}_{r-1,i}(x) &= \tilde{\varphi}_{r,i}(x) + b_i^r \tilde{\psi}_{j_r}(x),\ i \in I_r, \\
\tilde{\varphi}_{r-1,i}(x) &= \tilde{\varphi}_{r,i}(x),\ i \notin I_r, \\
\tilde{\psi}_{j_r}(x) &= \tilde{\varphi}_{r,j_r}(x) - \sum_{i \in Ir} a_i^r \tilde{\varphi}_{r,i}(x).
\end{aligned}$$

Analogously, the corresponding (new) dual filters $\{\tilde{h}^{new}_{r-1,i,l}\}_{i \in S_{r-1}, l \in S_r}$ will have the form:

$$\tilde{h}^{new}_{r-1,i,l} = \delta_{i,l}, \quad \forall i \notin I_r, l \in S_r, \quad \tilde{h}^{new}_{r-1,i,l} = 0, \quad \forall i \in I_r, l \notin I_r,$$

$$\tilde{h}^{new}_{r-1,i,j_r} = b_i^r, \quad \forall i \in I_r, \quad \tilde{h}^{new}_{r-1,i,l} = -a_i^r b_l^r, \quad \forall i, l \in I_r, i \neq l,$$

$$\tilde{h}^{new}_{r-1,i,i} = 1 - a_i^r b_i^r, \quad \forall i = l \in I_r.$$

The new dual filters, $\{\tilde{g}^{new}_{r-1,j_r,l}\}_{l\in S_r}$ can be obtained as

$$\tilde{g}^{new}_{r-1,j_r,l} = -a^r_l,\ \forall l \in I_r, \quad \tilde{g}^{new}_{r-1,j_r,l} = 1,\ l = j_r, \quad \tilde{g}^{new}_{r-1,j_r,l} = 0,\ \forall l \in S_{r-1} \setminus I_r.$$

More details on the lifting algorithm based on removing 'one coefficient at a time', as well as its applications when tackling regression in higher dimensional spaces can be found in Jansen *et al.* (2001, 2004).

3.4 Introducing adaptivity in the lifting algorithm

Let us remember that our goal is to obtain a transform that adjusts itself to the signal characteristics, and as such produces a sparse decomposition into wavelet coefficients.

Of course, our attempt of producing an adaptive transform is not the first one known in the literature. However, most adaptive lifting constructions introduced so far have been designed for image compression, rather than for 1D signals.

Claypoole *et al.* (2003) proposed an adaptive lifting scheme for image compression. Adaptivity is constructed within the prediction step, which consists of choosing from a set of linear predictors in such a way that if an edge is detected in the image, then the wavelet is chosen such that its support does not overlap the edge. An 'update first' approach is used, to skip sending information on the predictor being chosen. The scaling coefficients are obtained through updating, and then quantised. The prediction stage is applied to the quantised coefficients, and the detail coefficients are computed, quantised and transmitted. As previously mentioned, the swap in lifting steps has first been introduced by Claypoole *et al.* (1998). In their paper they propose two adaptive algorithms: one is scale-adapted— the predictor is chosen to match the signal structure at each scale, and the other one is space-adapted— the predic-

tor is chosen (from a family) to minimise each detail value. The transforms are investigated on a small simulation study on the Donoho and Johnstone (1994) signals sampled on regular grids. The proposed algorithms give very similar results, sometimes slightly better than those obtained if using the Daubechies wavelets.

Piella and Heijmans (2002) also follow an update first strategy, but they introduce adaptiveness in the update stage. The behaviour of the algorithm is briefly investigated by denoising a few signals, and only compared to the results produced by its fixed version, no other comparisons being made.

Trappe and Liu (2000) built adaptiveness into the prediction step with the goal of minimising the l_2-norm of the signal by using Wiener filtering. This adaptive algorithm was used for decorrelating the low-pass and high-pass subbands of an AR(2) process, and then for the shrinkage of the same process, corrupted by Gaussian noise.

All of the above adaptive lifting techniques use the usual odd/even splitting, while we shall build adaptiveness in the 'one coefficient at a time' methodology of Jansen *et al.* (2001). This allows for more flexibility for an eventual extension to higher dimensions. In our development, we also explicitly analyse the case of multiple observations per grid location.

Adaptiveness in our construction will be embedded in the prediction step of the lifting algorithm. At a careful analysis of the lifting procedure, we come to realize that the prediction step provides two potential sources of adaptiveness— the *regression order* and the *neighbourhood size and configuration*.

Some questions arise: how do we choose the neighbourhood— how many neighbours, in what configuration? What order of prediction should we employ and how do we decide on it?

These are in fact inter-related issues, as well as distinct questions— inter-related in the sense that the neighbourhood size dictates the maximum possible order of regression and vice versa, distinct in the sense that informed choices should be made, based on a prior information about the signal should this be

available.

For neighbourhood configuration, we will use two possible choices: *symmetrical neighbours* (the same number of neighbours at the left and right of the removed point) or *closest neighbours* to the removed point, irrespective of the side on which they lie. However, we do not restrict the *number of neighbours*.

Next, a function value corresponding to the point to be removed is predicted by fitting a curve over its (determined) neighbourhood. Regression of up to order three is used, that is we fit either a line, a parabola or a cubic over the cloud of points that are used as predictors. In the wavelet context this is equivalent to constructing dual wavelet functions with two, three, respectively four, vanishing moments.

The lifting algorithm as introduced by Jansen *et al.* (2001, 2004) needs the specification of whether the neighbours will be chosen symmetrically or closest, their number and the type of regression to be used— linear, quadratic or cubic. Then at each step the algorithm chooses the point to be removed, assesses its neighbourhood based on these pre-specified requirements, and predicts a function value by fitting a curve of the desired (already established) degree. We fully describe such a lifting algorithm by defining its type of prediction— either linear, quadratic or cubic— with, say m neighbours in a chosen (symmetrical or closest) configuration.

Our innovation consists of *introducing two adaptive prediction steps* that will skip some/all of the prior choices that have to be made when using the lifting scheme in its version described above. There are two levels of adaptivity that can be built in the prediction stage, and these in turn induce *two adaptive lifting algorithms*. The two methods we propose are:

1. **AdaptPred.** The construction is *adaptive over the order of regression used in the prediction scheme*. The algorithm chooses at each step the type of regression (linear, quadratic or cubic, with or without an intercept) which generates the smallest detail in absolute value. The wavelet bases constructed like this adapt themselves to the smoothness of the sig-

nal, investigated within a user-specified neighbourhood configuration and size. Hence this first adaptive method surpasses the problem of having to choose the order of regression. Note though that at each step curves of likely different orders are fit and used for generating the sequence of detail coefficients. We will refer to this procedure as **AdaptPred**, and for its complete description we will only need to specify the size and configuration of neighbourhood to be used.

2. **AdaptNeigh.** The second adaptive method *minimises the detail coefficients not only over the regression schemes, but also over the neighbourhood structure.* In other words, several configurations of neighbours are tested with the first adaptive transform, and the one yielding the smallest detail coefficient in absolute value will be chosen. Hence the wavelet bases constructed through this procedure adapt themselves to the smoothness of the signal within the best predictive window at each step. This construction completely frees the user of making any choice, except for the neighbourhood size. We will refer to the lifting algorithm that embeds this type of prediction step as **AdaptNeigh**, and observe that it is completely described through the specification of the number of neighbours to be used at each step.

To illustrate how our transforms choose various orders of prediction as a function of the local signal structure, it is instructive to refer to figures 3.2 and 3.3. Note how linear prediction is predominantly employed for *Blocks* while the smoother character of *HeaviSine* induces the usage of the other two types of prediction, except at the jumps which are mostly represented using linear prediction.

Our construction starts with an initial MRA generated by compactly supported scaling functions, and through iterating the lifting steps, we gradually build a new MRA. The filters used at each stage are finite, and therefore the resulting scaling and wavelet functions (primal and dual) will be compactly

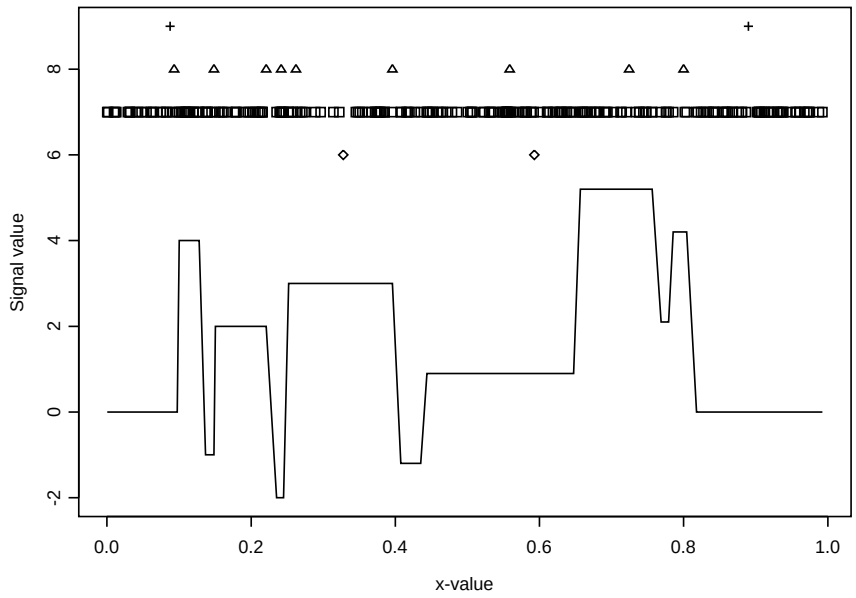

Figure 3.2: Plot showing choice of prediction scheme for the *Blocks* test signal decomposed with AN2 (see section 3.6.1) on an irregular grid. Horizontal placement of symbol indicates location of the following kinds of prediction: linear (□); quadratic (△); cubic (+); scaling functions (⋄).

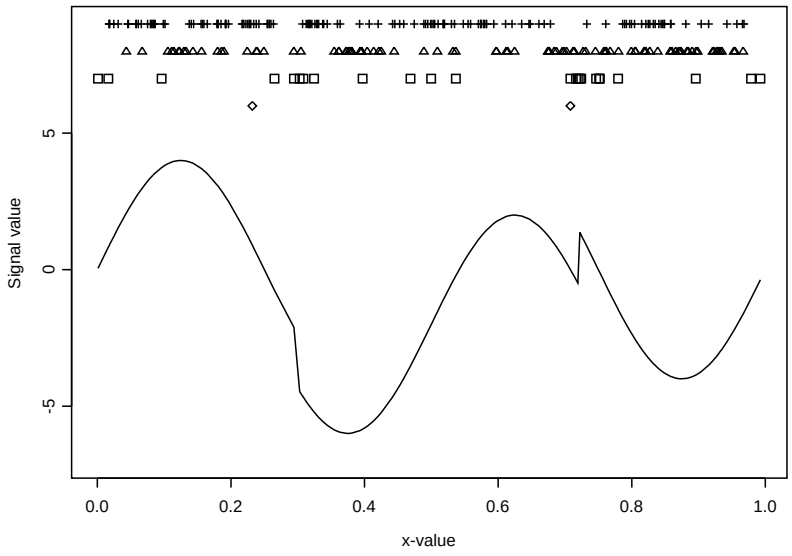

Figure 3.3: Plot showing choice of prediction scheme for the *HeaviSine* test signal decomposed with AN2 (see section 3.6.1) on an irregular grid. Horizontal placement of symbol indicates location of the following kinds of prediction: linear (□); quadratic (△); cubic (+); scaling functions (⋄).

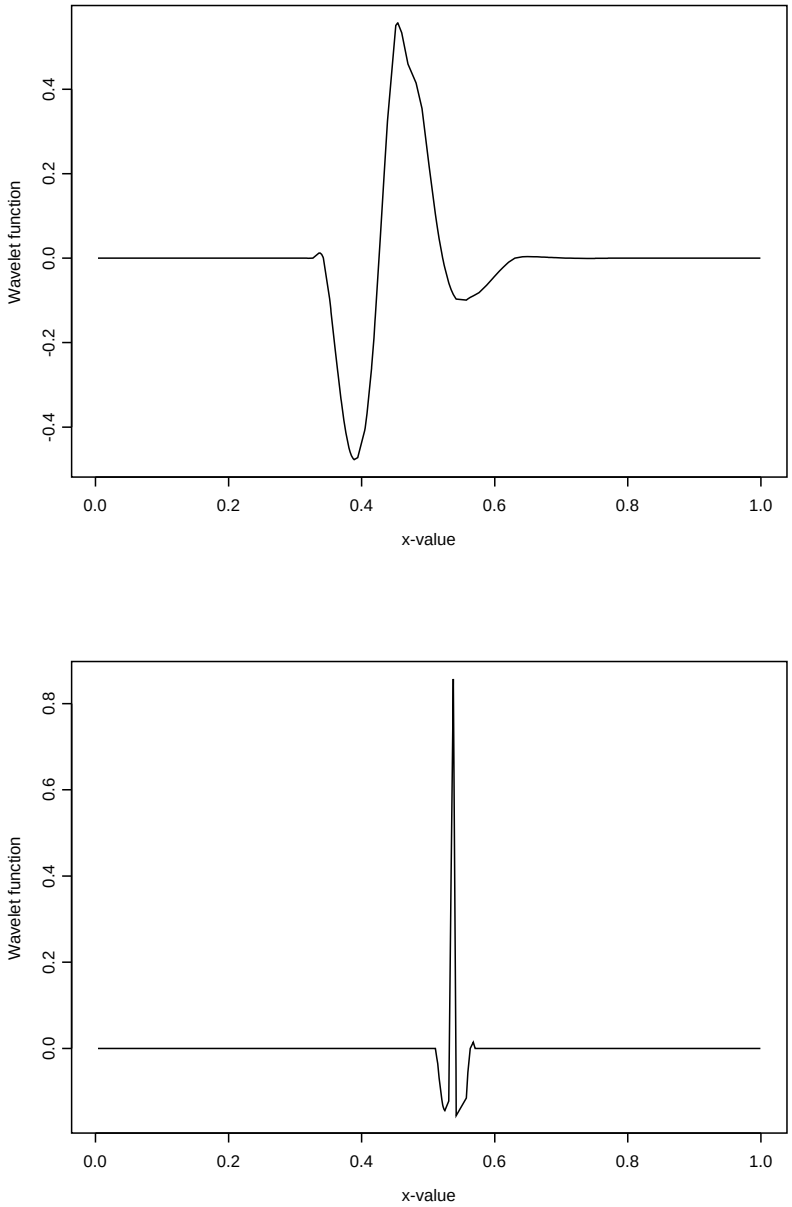

Figure 3.4: Wavelet functions at different locations (0.45, 0.54 respectively), obtained when decomposing a *Doppler* signal using AN2 (see section 3.6.1).

supported. It is possible to visualise the wavelet functions by performing a forward transform on a zero function at locations $\underline{x}$, then inserting the value 1 at the location of the wavelet coefficient whose wavelet function you want to construct and applying the inverse lifting transform. Figure 3.4 provides two examples of wavelet functions, obtained when decomposing a *Doppler* signal sampled at 256 irregular locations. The first one corresponds to the fourth observed point, which gets removed towards the end of the transform, hence the wider support of the corresponding wavelet; the second one corresponds to the 192^{nd} point, which is removed around the middle of the transform, hence its narrower support.

3.5 Implications of the adaptive construction

3.5.1 Prediction weights

Obtaining the prediction weights in the usual lifting algorithm.

Let us assume the neighbourhood definition is established.

At each step, say $r-1$ (when we remove the point j_r), we fit a curve through the cloud of points $\{(x_i, c_{r,i})\}_{i\in I_r}$ by using the least squares approach, i.e. model the data in neighbourhood I_r as

$$\{c_{r,i}\}_{i\in I_r} = X_r \underline{\beta}_r + \underline{\varepsilon}, \tag{3.18}$$

where $\{c_{r,i}\}_{i\in I_r} \in \mathscr{M}_{|I_r|,1}(\mathbb{R})$, X_r is the matrix that contains the grid values of the neighbouring points of x_{j_r} (hence it has $|I_r|$ rows) and its number of columns depends on the choice of fitting a curve through the origin or not (i.e. with or without an intercept) and on the order of the curve to be fitted to the data; $\underline{\varepsilon}$ is the random error vector with each of its components assumed to have mean zero and constant variance. The least squares estimator of the (column) vector of coefficients, $\underline{\beta}_r$, that ultimately gives the curve, is obtained through the well known formula $\hat{\underline{\beta}}_r = (X_r^T X_r)^{-1} X_r^T \{c_{r,i}\}_{i\in I_r}$.

Next assume that the point to be lifted (x_{j_r}, c_{r,j_r}) belongs to the estimated curve, so we predict its corresponding function value by $\hat{c}_{r,j_r} = X_{r,j_r}\underline{\hat{\beta}}_r$, where X_{r,j_r} is a row matrix having the same number of elements as the number of columns of the matrix X_r, filled in with the corresponding values x_{j_r}.

The matrices X_r and X_{r,j_r} reflect the type of prediction as well as the use of an intercept. For linear prediction with an intercept, the matrices will have the form $X_r = (1_{|I_r|}, \{x_i\}_{i\in I_r})$, $X_{r,j_r} = (1, x_{j_r})$, where $1_{|I_r|}$ denotes the column vector with $|I_r|$ entries, all equal to 1.

In the case of quadratic prediction with intercept, $X_r = (1_{|I_r|}, \{x_i\}_{i\in I_r}, \{x_i^2\}_{i\in I_r})$, $X_{r,j_r} = (1, x_{j_r}, x_{j_r}^2)$.

And finally when using cubic prediction with intercept,

$X_r = (1_{|I_r|}, \{x_i\}_{i\in I_r}, \{x_i^2\}_{i\in I_r}, \{x_i^3\}_{i\in I_r})$, $X_{r,j_r} = (1, x_{j_r}, x_{j_r}^2, x_{j_r}^3)$.

Not using an intercept results in the above matrices losing their first column.

Using the least squares expression of $\underline{\hat{\beta}}_r$ obtained above, we can express $\hat{c}_{r,j_r} = \sum_{i\in I_r} a_i^r c_{r,i}$, with the prediction weights $\underline{a}^r = \{a_i^r\}_{i\in I_r}$ given by

$$\underline{a}^r = X_{r,j_r}(X_r^T X_r)^{-1} X_r^T. \tag{3.19}$$

Observe that regardless the used regression order, the prediction is linear in the values of the scaling coefficients.

Influences of adaptiveness on the prediction weights.

When regression with an intercept is used, the prediction weights within each step sum to one ($\sum_{i\in I_r} a_i^r = 1$), so the areas where the function is constant yield exactly zero detail coefficients. When an intercept is not used, the weights are not guaranteed to sum to one. This is intuitively obvious since we fit curves that will always have a slope.

For the usual lifting scheme with fixed prediction order (be it linear, quadratic or cubic), formula (3.19) reveals that the prediction weights $\underline{a}^r$ only depend on the grid structure and on the chosen type of prediction, but not on the initial

function.

However, when using an adaptive procedure, at each step of the algorithm the choice of regression order is dictated by the signal structure, as the algorithm seeks to obtain the smallest detail in absolute value. For a better picture, let us consider what happens at stage $n-1$, when the first point, with index j_n, is chosen for removal. In the prediction stage, the curve to be fitted is obtained once we minimize $|c_{n,j_n} - \sum_{i \in I_n} a_i^n c_{n,i}|$ subject to type of prediction, and eventually to neighbourhood selection (for AdaptPred, AdaptNeigh respectively). As a consequence, the prediction weights depend not only on the locations $\underline{x}$, but also on the signal.

The dependence of the prediction weights on the signal induces a dependence of the updated integrals on the initial function. Hence the choice of the point to be removed next and the update weights also depend on the signal.

In other words, adaptiveness comes at the cost of building filters which depend on the initial signal, as well as on the locations $\underline{x}$. Therefore the adaptive lifting algorithm is not a linear transform (unlike the lifting scheme using linear, quadratic or cubic prediction), and various properties, such as its stability are difficult to assess.

3.5.2 Stability of the transform

Classical wavelet constructions produce stable wavelet bases. In simple terms, the elements of a stable basis are close to orthogonal, rather than arbitrarily close to each other.

Formally, a stable basis is defined as a Riesz basis, i.e. $\{\psi_k\}_k$ is a Riesz basis of $L^2(\mathbb{R})$ if, $\forall f \in L^2(\mathbb{R})$, there exists $\{d_k\}_k$ such that $f(x) = \sum_k d_k \psi_k(x)$ and

$$m\|d\|_{l_2} \leq \|f\|_{L^2} \leq M\|d\|_{l_2}, \tag{3.20}$$

where m, M are finite, depend only on the wavelet basis and are of comparable magnitude. The ratio $k = M/m$ is called *the condition number of the basis*

$\{\psi_k\}_k$. In the ideal case of working with an orthonormal basis, it equals 1 (as $m = M = 1$), while large values of the condition number illustrate an ill-conditioning of the basis.

We denote here the basis by $\{\psi_k\}_k$ to suggest that we are interested in the stability of a wavelet basis and that $\{d_k\}_k$ refers to the decomposition of f on a wavelet basis; however, the above definition is general.

For second generation wavelet constructions using the lifting scheme, although we may start with an initial MRA which defines stable bases, there is no guarantee that the resulting wavelet basis is stable— preservation of biorthogonality achieved through lifting is a necessary, but by no means sufficient condition for stability. Second generation wavelet bases have been shown in the literature to exhibit serious stability problems, discussed for example in Simoens and Vandewalle (2003), Vanraes *et al.* (2002).

Simoens and Vandewalle (2003) investigate the stability of wavelet bases obtained through the lifting algorithm using the odd/even split as introduced by Sweldens (1996). They show that prediction and update weights that are uniformly bounded in norm are required for a stable transform. They note that, loosely put, if the update weights are large, then the wavelet functions constructed through the update step will be close to the scaling functions from the same stage. To stabilize the wavelet basis, a modification of the update step is proposed, which consists in using a local semi-orthogonalization— construct wavelet functions that are orthogonal on the space spanned by a subset of scaling functions from the same level. This brings the basis closer to orthogonal, but it destroys the number of vanishing moments for the primal wavelets, hence this method is combined with the standard update to restore it.

From their study, it appears that higher irregularity in the locations induces higher instability in the wavelet basis. Also, while the number of vanishing moments for primal wavelets does not seem to influence stability, the dual number of vanishing moments considerably does (the higher the order of the MRA, the higher the instability of the wavelet basis).

Vanraes *et al.* (2002) observe that at each level of the lifting scheme, the update weights, which are responsible for the degree of non-orthogonality/overlap between the primal scaling and wavelet functions at the same stage, are influenced by the scaling functions obtained through the prediction step, and therefore they propose an alteration of this step. Irregularities in the initial observations translate in the potential use at the prediction step of data corresponding to locations that may be far apart. This is referred to as 'mixing of scales' (note again that scale in this context is influenced by the irregularity in observations, and it does not refer to a dyadic dilation of a basis function). The new scaling functions built in such conditions appear to have unwanted features, such as negative integrals, when designing MRA's of higher order than 2. This induces instability in the resulting basis, hence they propose to change the split procedure to prevent from such mixing, and then to modify the prediction step accordingly.

Potential instability of a wavelet basis may seriously affect the thresholding process. We have already seen that the main ingredient of thresholding is the sparsity of the true wavelet coefficients, which allows for interpreting small observed wavelet coefficients as due to noise.

Since sparsity of a wavelet decomposition is determined by the number of vanishing moments in the dual wavelets, any attempt to 'stabilise' the wavelet basis which affects the prediction stage of the lifting algorithm can result in altering the final sparsity of the wavelet coefficients.

Lack of orthogonality in second generation wavelet bases means that the signal energy is not preserved in the wavelet domain anymore. Small wavelet coefficients may potentially carry a substantial amount of information, and if no attention is paid we could end up thresholding essential coefficients and therefore losing significant amounts of information on the true signal. Also, altering the value of a detail coefficient may result in an unexpectedly large effect in the reconstructed signal.

In what follows we will investigate stability issues in our transforms.

Computing the condition numbers.

In an extensive simulation study, we investigated the stability of our transforms, along with their sparsity and denoising performance. Sparsity and denoising capacity will be thoroughly demonstrated in the next chapter, and for now we are discussing stability issues.

A practical way to assess the stability of a basis is by computing its condition number. The definition given before establishes that $k = M/m$, where M, m are the Riesz bounds associated to the wavelet basis, which is not easy to implement in practice. However, there is an alternative way of computing k, which involves the matrix associated with the fast wavelet transform.

We have already seen that the discrete wavelet transform can be put under a matrix multiplication form; on the same principle, we can also associate a matrix to the fast wavelet transform in the second generation wavelet context. Since the direct transform uses dual filters, we will denote the matrix associated to it by $\tilde{W}$, as opposed to the notation W used for the matrix associated to the DWT.

Due to the orthonormality of the classical wavelet bases, the matrix W is orthogonal, and therefore the matrix associated to the inverse transform is just W^T. This property does not hold for second generation MRA's, where we will denote the matrix associated to the inverse transform by W (since it uses primal filters) and it is nothing else than $\tilde{W}^{-1}$. The condition number can then be computed as $k = \|\tilde{W}\|_2 \|W\|_2$, where norm $\|\cdot\|_2$ of a $n \times n$ square matrix is given by $\|\tilde{W}\|_2^2 = \sum_{i,j=1}^{n} \tilde{w}_{i,j}^2$. In what follows we will show how to obtain the matrix generated by any of our transforms, as this means that we can compute their corresponding condition numbers.

In section 3.3 we obtained the expressions for the filters used at each step $(r-1)$ of the lifting algorithm removing one coefficient at a time. When using an adaptive transform, the filters can be obtained by using the same formulas, but with the appropriate prediction and update weights generated through the adaptive selection process.

Write now the filters from each step (say r, after the point indexed j_{r+1} has been removed) under matrix format, $\tilde{H}_r = \{\tilde{h}_{r,i,l}\}_{i\in S_r, l\in S_{r+1}} \in \mathscr{M}_{r,r+1}(\mathbb{R})$, and similarly $\tilde{G}_r = \{\tilde{g}_{r,j_{r+1},l}\}_{l\in S_{r+1}} \in \mathscr{M}_{1,r+1}(\mathbb{R})$. Denote by

$$\tilde{A}_r = \begin{pmatrix} \tilde{H}_r \\ \tilde{G}_r \end{pmatrix}, \tag{3.21}$$

the $(r+1)\times(r+1)$ matrix that comprises all the filters used at stage r for decomposing the signal (hence at step $n-1$ when the first split is performed, $\tilde{A}_{n-1}$ is of size $n\times n$). Then $\tilde{W}$ can be expressed as a function of $\tilde{A}_r$ with $r\in\{n-1,\dots,r_0\}$, where r_0 is the coarsest level down to which the transform is performed (where the point indexed by j_{r_0+1} is removed)

$$\tilde{W} := \tilde{W}_n^{n-r_0}, \tag{3.22}$$

with the notation explained in what follows. We denote by $\tilde{W}_r^s$ the $r\times r$ matrix (for any s) generated by a wavelet decomposition that started at stage r (where r scaling coefficients are available) with s counting the number of stages used in the wavelet decomposition (i.e. $s = \text{card}\{r-1,\dots,r-s\}$). The matrix $\tilde{W}$ can be obtained for $r := n$ by using the following recursive construction as a function of s

$$\tilde{W}_r^s = \begin{pmatrix} \tilde{W}_{r-1}^{s-1} & \mathbb{0}^T \\ \mathbb{0} & 1 \end{pmatrix} \tilde{A}_{r-1}, \tag{3.23}$$

with $\tilde{W}_r^1 = \tilde{A}_{r-1}, \forall r$ and $\mathbb{0}\in\mathscr{M}_{1,r-1}(\mathbb{R})$.

We have already noted that our adaptive transforms come at the cost of building filters that depend not just on the data locations, but also on the initial function. The formulas above highlight the dependence of the matrix $\tilde{W}$ on the filters used at each step of the transform, hence for our adaptive constructions $\tilde{W}$ depends both on the irregularities of the data locations and on the features of the signal. Note that the matrix $\tilde{W}$ can also be computed by performing n forward lifting transforms on functions determined by the

Kronecker vectors of length n corresponding to locations $\underline{x}$, with the value of 1 sequentially inserted on positions $1, \ldots, n$. When 1 appears on position k, the corresponding lifted sequence gives the k^{th} column of $\tilde{W}$.

We have numerically computed the condition numbers for our adaptive lifting algorithms with neighbourhoods of sizes 2 to 4, using both symmetrical and closest neighbours. The interpretation of their values is of course influenced by the dependence on the characteristics of the actual test function.

Additionally, we have also computed the condition numbers of the lifting scheme as introduced by Jansen *et al.* (2001, 2004), with linear, quadratic and cubic predictions respectively, neighbourhood sizes ranging from the minimal one (2 for linear, 3 for quadratic and 4 for cubic) to 4, in both configurations (symmetrical and closest). Their corresponding condition numbers however, only reflect the irregularity in the grid.

All transforms were tested on functions that display a wide range of behaviours and smoothness levels, observed at locations with various degrees of irregularity (see section 4.1). We postpone the presentation of the test functions until the next chapter, where we give a detailed exposition of the simulation study we performed.

Analysing the results, some conclusions emerge: the usage of larger neighbourhoods induces instability in the resulting wavelet basis, and so does the usage of higher prediction order. These findings are consistent with those mentioned in the reviewed literature. We saw that Vanraes *et al.* (2002) suggested that scale mixing induces instability. In our construction, scale mixing is most likely to appear when using neighbourhoods of larger sizes, which automatically include distant points. Also, Simoens and Vandewalle (2003) noted that MRA's of higher order are associated with high condition numbers. In our algorithm, this is the equivalent of using a prediction step that fits curves of higher order to the neighbourhood data. However, our construction does not seem to be very sensitive when increasing the degree of irregularity in data locations.

3.5.3 Characteristics of the adaptive lifting construction

In the light of our analysis, we took the following strategy in order to ensure stable transforms:

(i) Choosing the neighbourhood.

In both adaptive algorithms the neighbourhood size has to be specified by the user, and a choice can be made based on the prior knowledge of the signal. We do however advise against using large neighbourhoods, as this increases the chances of using points that do not belong to the same scale.

In our algorithm, when at a particular stage the point to be removed happens to be on the boundary, rather than using the requested number of neighbours (which would then come only from one side), we only use its closest neighbour to prevent against using 'artificial' neighbours. Also, after reiterating the algorithm several times, when the number of scaling coefficients has significantly decreased, the requested number of neighbours might not be available. In this case, we decrease the number of used neighbours to the maximum available. This in turn results in decreasing the number of vanishing moments for the dual wavelets, which potentially diminishes the sparsity of the resulting wavelet coefficients.

Our investigations have shown that using asymmetrical neighbours does not seem to have an impact on our transforms, as long as we make use of the appropriate number of neighbours. If closest neighbours are chosen, the prediction weights can take negative values. When updating the corresponding integrals, some of them will in turn become smaller rather than larger, which contradicts the intuition that if we remove a point each of the remaining points should span a larger set to account for the removed point. In this situation the observation from Jansen *et al.* (2004) that the scales of the wavelet functions are a monotonic function of the order of removal, does not hold.

(ii) Regression order.

Unstable transforms are generated using curves of higher order than the available number of neighbours would allow (i.e. curves of the highest possible order, forced through the origin). Hence we use each type of prediction with its appropriate number of neighbours— the higher the order of prediction we use, the more neighbours we will have to request in order to get non-degenerated curves. In order to determine a line with a slope, we will need at least 2 neighbours, for a parabola at least 3, while for a cubic at least 4. At certain steps of the transform we will be in the situation of not having enough neighbours, hence the order of prediction needs to be decreased, while for boundary points we will always predict using step functions. In consequence, we need to be aware that if we go low enough in our decomposition when requesting a higher order polynomial prediction, in fact we will obtain a mixture of regression orders (although we required a fixed order). So the final order of our primal MRA will not be exactly the requested one, and this again may have an effect on the approximation capacity of the wavelet basis. However, the update weights are obtained at each step such that the primal wavelets have one vanishing moment, so the final dual MRA will always have one vanishing moment.

(iii) Update weights. The update weights are also responsible for higher condition numbers, and in our algorithm we used $b_i^r = I_{r,j_r} I_{r-1,i} / \sum_{k \in I_r} I_{r-1,k}^2$, as suggested by Jansen *et al.* (2001). This choice ensures that the obtained weights have minimum norm, which prevents the new wavelet and scaling functions at each stage (here r) from being too close to each other. For the future, it would be interesting to investigate a possible transformation (perhaps similar to the local semi-orthogonalization suggested by Simoens and Vandewalle (2003)) that would bring our bases closer to orthogonality.

3.5.4 Inverting an adaptive lifting scheme

Inverting an adaptive transform is straightforward and it follows the same basic steps as the inversion of the usual lifting scheme. However, we need to record the prediction operator used at every step (for instance at stage r, quadratic prediction with an intercept), and in the case of AdaptNeigh, we also need to record the type of neighbourhood that generated (at each step) the smallest detail in absolute value (say, at stage r one neighbour at the left and two neighbours at the right).

3.5.5 Handling multiple observations at a single grid-point

Although so far we have only considered the situation where each location x_i is associated to exactly one observation f_i, we encounter many real situations where we can have several values f_i at the same location x_i. The famous motorcycle data of Silverman (1985) is one such example (see figure 3.1, which shows the resulting estimated curves by using our adaptive transform AP1S and two linear smoothing methods).

For such situations, if we were to change nothing in our algorithm, then we would find ourselves in situations where some points are assigned zero integrals, as the distance from a point to itself is zero. Hence the multiple points would always be the first ones to be removed. To prevent against this situation, multiple points are considered to have only one location value.

Next, let us inspect what happens if (some of) the neighbours of the point chosen to be removed next have multiple f values. The prediction step uses a linear least squares approach to estimate the unknown parameters of the regression curve to be fit to the data, which can naturally cope with multiple observations. When an adaptive transform is used, the usual procedure can be applied with no modification, i.e. fit various regression curves and choose the one that provides the smallest detail in absolute value.

In the update step, all multiple neighbours get updated using the corresponding detail obtained in the prediction stage. So neighbour points that were multiple remain multiple after the update step.

If the point to be removed is itself multiple, then we inspected more possibilities of handling it. Multiple detail coefficients can be obtained, but this raises problems in the next step when we have to update the neighbours, some of which might also be multiple points. We have therefore chosen to produce one detail coefficient by taking the mean of the distinct individual coefficients, although other quantities, such as the minimum, can be interesting alternatives.

Finally, when the coarsest level for decomposition has been reached, if some of the scaling points are multiple, we replace them by their means, and then invert the lifting transform.

3.6 Sparsity and denoising performance of the adaptive algorithms

3.6.1 Sparsity results

Here we only present conclusions regarding the sparsity properties of our transforms. These claims are based on a thorough investigation of the behaviour of our transforms on various test functions and degrees of grid irregularity. The next chapter will provide full details on setting the simulation study and its results.

We use the following abbreviations to refer to the names of the methods investigated: the lifting algorithm with simple linear/quadratic/cubic prediction will be referred to as LP/QP/CP respectively. AdaptPred will be denoted by AP and AdaptNeigh by AN. After this two letter code we use a number to indicate the used number of neighbours, and a letter N/S to indicate if nearest/symmetrical neighbours have been used (hence when the code is S,

twice the number of neighbours is used). The letter N/S is not needed for AdaptNeigh, since this procedure searches through both types of configurations anyway.

As mentioned before, along with testing the behaviour of our transforms with neighbourhood sizes up to 4, we have also investigated the linear transforms (LP, QP and CP), also with neighbourhood sizes up to 4.

Out of all linear algorithms, LP with 2 neighbours gives the best compression results, and it is also associated to small condition numbers. However, in terms of sparsity, the adaptive procedures perform better than any of the linear ones, with AN producing the sparsest outcomes. Hence increasing the adaptiveness increases the sparsity of the results. Due to stability considerations, we recommend the use of AN1, followed by AP2N and AP1S.

When compared to classical wavelet methods (on regular grids), our AN1 produces competitive results, and it additionally has the advantage of not needing to specify a wavelet basis for decomposition (a choice that can seriously influence the sparsity of the classical wavelet coefficients).

Also, our simulations showed that increasing the irregularity of the data locations $\underline{x}$ does not affect the sparseness of the detail coefficients obtained by decomposing the signal through any of the linear or adaptive techniques.

3.6.2 Wavelet shrinkage using adaptive lifting

Assume we are in the familiar nonparametric regression setting, and the observations $\underline{f}$ are modelled as

$$f_i = g(x_i) + \varepsilon_i, \tag{3.24}$$

where the locations $\underline{x}$ are assumed irregular and fixed, $\underline{g} = \{g(x_i)\}_{i \in \overline{1,n}}$ are the true signal values, to be estimated, and ε_i is identically distributed, independent noise assumed to follow a $N(0, \sigma^2)$ distribution.

Remember that the wavelet shrinkage/thresholding approach consists in

three steps: (i) decompose $\underline{f}$ into a sequence of scaling and detail coefficients, (ii) shrink/threshold the wavelet coefficients and (iii) invert the wavelet transform, which gives an estimate $\hat{\underline{g}}$ of $\underline{g}$. A successful classical wavelet shrinkage approach relies on how efficient the wavelet transform is at sparsely representing functions. Since our transforms produce sparse sets of details, using a shrinkage technique is feasible with adaptive lifting. In our shrinkage approach we shall use an adapted version of the empirical Bayes thresholding technique of Johnstone and Silverman (2004a,b, 2005), modified to suit the characteristics of our transforms. Hence we start by presenting this method in the context of the DWT, as it was initially introduced, and then we will discuss and present the modifications needed to suit the adaptive lifting transform.

The empirical Bayesian approach to wavelet shrinkage.

The empirical Bayes method relies on the sparsity of the true DWT wavelet coefficients corresponding to many classes of functions. This allows for placing a prior distribution on the true wavelet coefficients, $d^*_{j,k}$, that expresses the initial sparsity 'belief', which will then be updated by using the information contained in the observed wavelet coefficients, $d_{j,k}$. For each $d^*_{j,k}$ its posterior distribution is obtained, and an estimate $\hat{d}^*_{j,k}$ can be obtained by taking the mean or median of its corresponding posterior distribution. Heuristically, this is equivalent to assuming that the function g is not fixed, finding the distribution function corresponding to the 'class' of functions to which g belongs (based on its noisy observation f), and then taking the mean or median of this distribution to obtain a representation of g.

Since (true) wavelet coefficients at different levels differ in sparsity, with the finest levels mostly consisting of zero details, while the coarsest levels mostly containing bigger coefficients that carry a lot of signal, in their work Johnstone and Silverman (2004a) carry out thresholding in a scale dependent fashion.

More exactly, the true wavelet coefficients within a level j are modelled by a prior distribution describing each of them as being zero with probability π_j (to be estimated) or to have come from a heavy-tailed distribution γ, with

probability $1 - \pi_j$. So for the true DWT wavelet coefficients at level j, we assume

$$d^*_{j,k} \sim \pi_j \delta_0 + (1 - \pi_j)\gamma, \quad k \in \{0, \ldots, 2^j - 1\}, \tag{3.25}$$

independent random variables.

As $d_{j,k} = d^*_{j,k} + e_{j,k}$ with $e_{j,k}$ independently distributed as $N(0, \sigma^2)$ we have $d_{j,k} \sim N(d^*_{j,k}, \sigma^2)$, independently conditional on the $d^*_{j,k}$.

Johnstone and Silverman (2004a) set out the details for obtaining the posterior density of $d^*_{j,k}$ given $d_{j,k}$ for two choices of γ, Laplace and quasi-Cauchy. In our work we will use a prior mixture involving the quasi-Cauchy density function. Each of the true wavelet coefficients $d^*_{j,k}$ will then be associated to a posterior distribution, and an estimate $\hat{d}^*_{j,k}$ can be obtained by taking the respective posterior mean or median. The DWT transform can then be inverted, and an estimate for $\underline{g}$ obtained.

To estimate the probability π_j within each level j, use a marginal maximum likelihood approach involving the tractable common (marginal) density of $\{d_{j,k}\}_k$. The probability π_j gives the proportion of zero (true) wavelet coefficients at level j, and is high for fine levels, which corresponds to a high threshold, and low for coarse levels, where a small threshold is used.

Since the noise is assumed Gaussian, it is appropriate to estimate its standard deviation σ by the 'mad' of the finest observed details, which are assumed to be mostly representing the noise.

The empirical Bayesian approach for adaptive lifting.

When the locations $\underline{x}$ are irregular, using the DWT to decompose $\underline{f}$ is no longer appropriate. In what follows we investigate the model obtained by using the linear or adaptive lifting transforms to decompose the signal. Hence $\tilde{W}\underline{f} = \tilde{W}\underline{g} + \tilde{W}\underline{\varepsilon}$, where $\tilde{W}$ is the matrix associated to the lifting transform. Equivalently, we have

$$d_j = d^*_j + e_j, \tag{3.26}$$

where $\underline{d} = \{d_j\}_j$ is the observed detail, $\underline{d}^* = \{d^*_j\}_j$ is the true detail and

$\underline{e} = \{e_j\}_j$ represents the noise in the wavelet domain.

If one of the linear lifting transforms is used, then $\tilde{W}$ depends only on $\underline{x}$ and $\underline{d}^*$, $\underline{e}$ are the details obtained by applying the same lifting transform on the sequences $\underline{g}$, respectively $\underline{\varepsilon}$.

However, if an adaptive transform is used, the matrix $\tilde{W}$ depends on the structure of $\underline{f}$, and $\tilde{W}\underline{g}$ no longer represents the details obtained by applying the same adaptive lifting transform to $\underline{g}$, but merely the details obtained by applying a pre-determined lifting scheme generated by the local structure of $\underline{f}$. The approximation we make here is to assume that the two functions f and g have similar structures, and therefore consider $\underline{d}^*$ to be the sequence of true details. The vector $\underline{e}$ is considered to represent the noise, although it is not obtained by applying the same adaptive lifting scheme to the initial noise $\underline{\varepsilon}$. Conditional on the local structure of the initial signal, $\underline{e}$ can nevertheless be expressed as a linear combination of the initial noise $\underline{\varepsilon}$, $\underline{e} = \tilde{W}\underline{\varepsilon}$. As the initial noise components ε_i are assumed to be independent, normally distributed random variables with zero mean, it follows that e_j are normally distributed with zero mean too.

However, $\mathrm{cov}(e_j, e_{j'}) = \mathrm{cov}(\sum_{k=1}^n \tilde{w}_{j,k}\varepsilon_k, \sum_{k'=1}^n \tilde{w}_{j',k'}\varepsilon_{k'})$, where $\tilde{w}_{j,k}$ is the (j,k)th entry of the matrix $\tilde{W}$. Hence

$$\mathrm{cov}(e_j, e_{j'}) = \sum_{k=1}^{n}\sum_{k'=1}^{n} \tilde{w}_{j,k}\tilde{w}_{j',k'}\,\mathrm{cov}(\varepsilon_k, \varepsilon_{k'}). \tag{3.27}$$

As the initial noise is uncorrelated with variance σ^2, we have

$$\mathrm{cov}(e_j, e_{j'}) = \sigma^2 \sum_{k=1}^{n} \tilde{w}_{j,k}\tilde{w}_{j',k}, \tag{3.28}$$

or in short $\mathrm{cov}(e_j, e_{j'}) = \sigma^2(\tilde{W}\tilde{W}^T)_{j,j'}$. This shows that the errors $\underline{e}$ are correlated, and different coefficients e_j have different variances. From model (3.26) it follows that the observed details $\underline{d}$ are also normally distributed and correlated, with the same variance-covariance matrix as the errors.

We now list the issues that appear when applying an empirical Bayesian procedure to the detail coefficients of (3.26), and our modifications

- In the empirical Bayes approach previously introduced, the observed detail coefficients within a level (here d_j) are assumed to be independently distributed conditional on the true details (d_j^*). This obviously does not hold in our construction, and as suggested in Jansen *et al.* (2004), we will ignore the existing correlations between the observed details. This is backed up by Johnstone and Silverman (2004a) who ensure that although ignoring correlations results in loss of information in the estimation procedure, as long as there is not too much dependence the procedure will still perform well.

- Note that while in the empirical Bayes procedure for DWT data the wavelet coefficients are thresholded/shrunk according to their scale, the detail coefficients in model (3.26) have an associated scale of a continuous nature (a scale measure was introduced in section 3.3). We will therefore construct *artificial levels* to which we assign the details as a function of their scale, as suggested by Jansen *et al.* (2004). More exactly, we create levels by dividing the sequence of integral lengths $\{I_{r,j_r}\}_r$ into smaller sequences— up to its median, beween its median and its upper quartile, between its upper quartile and the 87.5th quantile, and so on. The coefficients are assigned to their corresponding 'level', with the finest scale coefficients corresponding to the finest artificial level, etc. Making use of the division into artificial levels, we can estimate the probability π_j for each level by maximum likelihood, and the observed details at the finest level can be used for estimating σ.

- The last problem we need to deal with in order to be able to use an empirical Bayes technique for thresholding the details d_j is the fact that different details have different variances, as we have seen above— $\mathrm{cov}(d_j, d_{j'}) = \sigma^2(\tilde{W}\tilde{W}^T)_{j,j'}$, so $\mathrm{var}(d_j) = \sigma^2\mathrm{diag}(\tilde{W}\tilde{W}^T)_j$, where by

$\text{diag}(A)_j$ we denote the (j,j)th entry of the matrix A.

To overcome this we will take the natural approach of renormalizing the data, i.e. the likelihood of the normalized details $d_j\{\text{diag}(\tilde{W}\tilde{W}^T)_j\}^{-1/2}$ is given by $N(d_j^*\{\text{diag}(\tilde{W}\tilde{W}^T)_j\}^{-1/2}, \sigma^2)$ conditional on the true normalized details $d_j^*\{\text{diag}(\tilde{W}\tilde{W}^T)_j\}^{-1/2}$ (and assume conditional independence within each level). Since the true details are modified only through multiplication by a fixed quantity, the prior of sparsity can now be translated to the modified details, $d_j^*\{\text{diag}(\tilde{W}\tilde{W}^T)_j\}^{-1/2} \sim \pi_l\delta_0 + (1-\pi_l)\gamma$, independently for all details that belong to the same artificial level l. In this context, σ can be estimated by $\text{mad}(d_j\{\text{diag}(\tilde{W}\tilde{W}^T)_j\}^{-1/2}$, j such that $d_j \in$ finest artificial level), and the posterior distribution can be worked out as usual. The final shrunk/thresholded coefficients are given by the posterior mean/median of the normalized details, multiplied by $\{\text{diag}(\tilde{W}\tilde{W}^T)_j\}^{1/2}$. The modified detail coefficients are then inverted to obtain the denoised signal.

We note here that we have also developed procedures to take into account situations when the initial signal observations are subject to heteroscedastic noise. Assuming the initial observations variances are known up to proportionality, i.e. $\text{var}(f_i) = \sigma^2\eta_i^2$, where η_i is a known proportionality factor and σ^2 is unknown. After applying the lifting transform, the variances of the detail coefficients are given by $\text{var}(d_j) = \sigma^2 \sum_{i=1}^n \eta_i^2 \tilde{w}_{j,i}^2$. The procedure entirely parallels the one described above. To estimate σ we normalize the wavelet coefficients by dividing by $\{\sum_{i=1}^n \eta_i^2 \tilde{w}_{j,i}^2\}^{1/2}$ and then use the 'mad' of the normalized details belonging to the first artificial level. After normalizing, thresholding and un-normalizing, we can invert the transform to form an estimate.

If the variance is heteroscedastic and we do not have any other structural knowledge about it, then we take an approach similar to that suggested by Kovac and Silverman (2000) and estimate the variance σ_j associated to the detail coefficient d_j as follows: identify all detail coefficients within a window

centered on the x_j in the data domain, estimate σ_j by taking the 'mad' of those identified coefficients that belong to the finest artificial level and threshold. We can then invert the transform and obtain an estimate for $\underline{g}$.

Remark. The way the correlation structure builds in the transformed data is more obviously revealed by inspecting how the prediction and update steps build upon each other. Again, since the structure of the adaptive lifting transforms is dependent on the input function, f, the following results are obtained by conditioning on the local structure of the signal.

The first step of the lifting transform (3.2) gives

$$d_{j_n} = c_{n,j_n} - \sum_{i \in I_n} a_i^n c_{n,i},$$

where $c_{n,i} = f_i$ are independent random variables, distributed as $c_{n,i} \sim N(g(x_i), \sigma^2)$.

It follows that d_{j_n} is itself a normally distributed random variable, with variance given by

$$\operatorname{var}(d_{j_n}) = \sigma^2 \left\{ 1 + \sum_{i \in I_n} (a_i^n)^2 \right\} \tag{3.29}$$

and $\operatorname{cov}(c_{n,i}, d_{j_n}) = -a_i^n \sigma^2$, for all $i \in I_n$ $(i \neq j_n)$, $\operatorname{cov}(c_{n,i}, d_{j_n}) = 0$, for $i \notin I_n$ $(i \neq j_n)$ and $\operatorname{cov}(c_{n,i}, d_{j_n}) = \sigma^2$, for $i = j_n$.

From the update step (3.4), $c_{n-1,i} = c_{n,i} + b_i^n d_{j_n}$, for all $i \in I_n$ $(i \neq j_n)$, hence $\operatorname{var}(c_{n-1,i}) = \operatorname{var}(c_{n,i}) + 2b_i^n \operatorname{cov}(c_{n,i}, d_{j_n}) + (b_i^n)^2 \operatorname{var}(d_{j_n})$, so $\operatorname{var}(c_{n-1,i}) = \sigma^2 - 2a_i^n b_i^n \sigma^2 + (b_i^n)^2 \operatorname{var}(d_{j_n})$, for all $i \in I_n$ $(i \neq j_n)$. Also, from (3.4), $c_{n-1,i} = c_{n,i}$, for all $i \notin I_n$ $(i \neq j_n)$, hence $\operatorname{var}(c_{n-1,i}) = \sigma^2$.

From the update step it follows for $i, j \in I_n$ $(i, j \neq j_n)$

$$\operatorname{cov}(c_{n-1,i}, c_{n-1,j}) = \delta_{i,j}\sigma^2 + (-a_i^n b_j^n - a_j^n b_i^n)\sigma^2 + b_i^n b_j^n \operatorname{var}(d_{j_n}), \tag{3.30}$$

for $i \in I_n$, $j \notin I_n$ $(i, j \neq j_n)$ we have

$$\operatorname{cov}(c_{n-1,i}, c_{n-1,j}) = \operatorname{cov}(c_{n,i} + b_i^n d_{j_n}, c_{n,j}) = 0 \tag{3.31}$$

and for all $i, j \notin I_n$ $(i, j \neq j_n)$,

$$\operatorname{cov}(c_{n-1,i}, c_{n-1,j}) = \operatorname{cov}(c_{n,i}, c_{n,j}) = 0. \tag{3.32}$$

Hence the update step induces correlations between the scaling coefficients at the next coarser level $(n-1)$, and also between the newly generated detail and (some of) the scaling coefficients at the upper level. As the algorithm proceeds with formula (3.2) for the next coarser level $n-1$, we see that in order to obtain the variance of the next detail coefficient we need to use the correlations between the scaling coefficients obtained above, so the correlations propagate from level to level.

However, although such an approach unveils the way the correlations are built through the transform, at the same time it hides an obvious understanding of the normality of the coarse and detail coefficients. This is due to the recursiveness in computing the scaling and wavelet coefficients. By combining the update relations $c_{r-1,i} = c_{r,i}$ for $i \in S_{r-1} \setminus I_r$ and $c_{r-1,i} = c_{r,i} + b_i^r d_{j_r}$ for $i \in I_r$ with the definition of the detail $d_{j_r} = c_{r,j_r} - \sum_{i \in I_r} a_i^r c_{r,i}$, it follows that for any r, all $c_{r-1,i}$ and d_{j_r} can be written as linear combinations of $\{c_{r,k}\}_{k \in S_r}$. Recursively, this leads to the fact that $c_{r-1,i}$ and d_{j_r} can be expressed as linear combinations of the initial $c_{n,k} = f_k$ with $k \in S_n$, which are independent, normally distributed random variables, so the normality of the coefficients obtained by the lifting transform is now apparent. This is just another way of obtaining the matrix representation of the lifting transform.

3.6.3 Denoising performance of our adaptive algorithms

As for the sparsity results, here we will only synthesize the main results from our simulation study in which we tested the denoising capacity of our methods. The next chapter will provide full details on setting the conditions of the study and on the comparisons performed.

We investigated the denoising capacity of our adaptive lifting transforms,

AP and AN, as well as the linear lifting transforms LP, QP and CP, all of them with neighbourhoods of up to size 4. Various degrees of irregularity in the locations have been considered (see section 4.1), together with various degrees of noise level (see section 4.3 for details). Additionally, we compared our methods with the wavelet technique developed by Kovac and Silverman (2000) (we will refer to it as KS) which is able to work on irregular data, as well as with *Locfit* (Loader (1997, 1999)), the smoothing spline function in S-Plus (`smooth.spline()`) and the denoising algorithm of Comte and Rozenholc (2004).

We found that our adaptive methods perform very well. For signals that present a lot of discontinuities, AN1 works extremely well, and it outperforms all four competitors. For smoother signals, the best denoising performance is obtained by using AP with neighbourhoods of size 2, regardless their configuration. With the exception of one smoother tested function, our AP outperforms all other methods, and the competitor coming closest is generally KS.

The results for our methods display a high degree of homogeneity within the level of grid irregularity for each noise level.

3.7 Conclusions and further work

In this chapter we introduced two novel nonparametric regression methods based on introducing adaptivity into the 'one coefficient at a time' lifting algorithm of Jansen *et al.* (2001, 2004). The proposed methods have the flexibility of handling any type of location irregularity and signal length, as well as naturally working on multiple observations at the same location. Simulation results (detailed in the next chapter) show that our proposed methods are very competitive when compared to established wavelet and non-wavelet techniques designed to work on irregular data. By locally building wavelet functions adapted to the signal smoothness, our proposed adaptive methods have the benefit of not having to deal with the (possibly) subjective choice of

what wavelet to use, encountered in the classical wavelet approaches.

Several aspects can be studied in further work, such as extending the proposed adaptive methods to work on data of higher dimension, investigate possible ways of bringing our transforms closer to orthogonality and devise a way of shrinking the wavelet coefficients that could deal with the coefficient correlations. A challenge would be to further study the statistical properties of adaptive lifting.

Chapter 4

Adaptive Lifting: simulations and results

In this chapter we describe the details of the simulation study through which we investigated the behaviour of our adaptive lifting transforms, as well as that of the lifting transforms using linear, quadratic and cubic prediction. All these transforms are based upon the lifting scheme removing 'one coefficient at a time' and have been introduced in the previous chapter.

This chapter discusses and interprets the obtained results. The conclusions we reach will substantiate the various claims made in the previous chapter regarding the performance of our adaptive lifting algorithms.

The presentation is organized as follows: first we introduce the test functions that we will use, then move onto investigating the sparsity properties of our transforms and finally we establish the denoising performance of our algorithms.

For decomposing signals we will use our adaptive lifting transforms, AdaptPred (AP) and AdaptNeigh (AN), and also the linear lifting algorithms with linear, quadratic and respectively cubic prediction (LP, QP, respectively CP). We shall investigate their behaviour when using appropriate neighbourhoods of sizes up to 4, in both configurations, nearest (N) or symmetrical (S) neighbours. The methods QP, CP with their corresponding number of neighbours

needed to prevent from fitting degenerate curves, and AN with more than two neighbours did not do very well (see section 3.5.2), so we will not present them in our report.

In the sections that analyse the sparsity and denoising performance, we will compare our algorithms with competitive methods, wavelet and non-wavelet based, which are suitable for our type of observations. At that stage, we will also give a brief presentation of these 'competitors'.

We will end the chapter by presenting a real-life example.

4.1 Test functions: construction and sampling

Test functions. The behaviour of our adaptive transforms, as well as that of the 'competitor' methods, will be tested on the test functions *Blocks*, *Bumps*, *HeaviSine* and *Doppler* introduced by Donoho and Johnstone (1994), which attempt to model various signals that arise in practice, and on the *Ppoly* function introduced by Nason and Silverman (1994)— see figure 4.1. *Blocks* is a piecewise constant function with jumps of various heights, representative of functions that appear in imaging. *Bumps* is constructed as a sum of 'bumps' located at the same points as the jumps of *Blocks*, of varying heights and widths— such signals appear, for instance, in nuclear magnetic resonance (NMR) spectroscopy. *HeaviSine* is a sinusoid of period 1 with two jumps, *Doppler* is a signal with variable (decreasing) frequency and *Ppoly* is a piecewise polynomial with a jump.

Grid locations. Let us now construct the locations at which these functions will be sampled. Start with a regular grid of length $n = 256$ that spans the interval [0,1], and generate irregularities in the distribution of locations by gradually 'jittering' this initial (regular) grid. Throughout our study we shall use grids with three degrees of irregularity, constructed as follows.

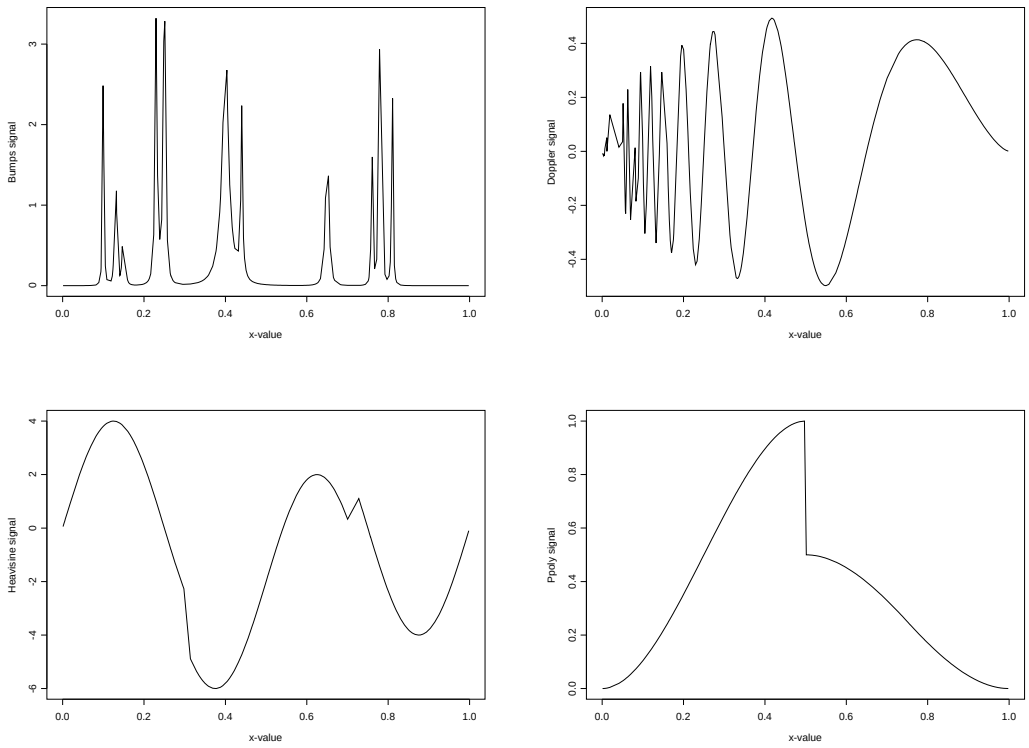

Figure 4.1: *Bumps*, *Doppler*, *HeaviSine* and *Ppoly* signals sampled at 256 irregular locations.

> Start with the initial regular locations, and shift each location with a random value generated from a uniform distribution on the interval $(-d/(n-1), d/(n-1))$. The degree of grid irregularity is dictated by the magnitude of d. We used three choices $d_1 = 0.01, d_2 = 0.1$ and $d_3 = 1$, each of them corresponding to higher location irregularity. This construction ensures a gradual departure from regularity, and therefore we will have a scale of comparability with techniques that only work on regular observations.

We now have the test signals (say g) constructed as described above, which we shall sample at $n = 256$ locations, $\underline{x}$. The sampled function values are the 'true' values and correspond to the notation $\underline{g}$ of the previous chapters.

Having constructed the (true) signals and established in the previous chapter how to decompose them, we are now ready to start investigating the sparsity

of the wavelet representations obtained through our transforms.

4.2 Sparsity investigation

For evaluating how successful our constructions are at sparsely representing functions, we will consider the test functions introduced in the previous section, sampled both at regular and irregular locations. Sampling on regular grids facilitates the comparison with classical wavelet methods. We shall use the three degrees of jitter previously introduced to check if our adaptive representations are affected by the increasing degree of irregularity in the grid. As a diagnostic tool, we will use a *sparsity plot*, constructed as follows.

4.2.1 Sparsity plot construction

First fully decompose (with the chosen lifting algorithm) the test function ($\underline{g}$) down to two scaling coefficients, all the other coefficients being transformed into details, then order the details in a decreasing sequence as a function of the absolute value of their magnitude.

Invert the wavelet transform first using only the information in the two scaling coefficients (i.e. consider all the details equal to zero), and then gradually introduce detail coefficients, one by one, starting from the largest one in absolute value, inverting the lifting algorithm each time.

Denote by i the number of wavelet coefficients that are included when inverting the transform. At first $i = 0$ (when only the two scaling coefficients participate in the reconstruction) and we obtain the reconstructed signal $\underline{\hat{g}}(0)$, followed by $i = 1$ (the two scaling coefficients and the largest detail in absolute value are used for inverting the transform), corresponding to $\underline{\hat{g}}(1)$, then $i = 2$ and so on up to $i = n - 2$, which provides perfect reconstruction of the initial signal, since all the scaling and detail coefficients are used in the inversion process, $\underline{\hat{g}}(n-2) = \underline{g}$.

Through this construction we are able to identify how many (large) wavelet

coefficients are needed to provide a reconstruction that virtually coincides with the test signal— the smaller their number, the sparser the wavelet decomposition.

To obtain an average behaviour of our methods, we perform this procedure over $K = 50$ data sets $(\underline{x}^j, \underline{g}^j)$, $j \in 1, \ldots, K$, within each jitter level and each test function. We shall measure the overall degree of closeness between the reconstructed function using i detail coefficients and the true function, by the integrated squared error

$$\mathrm{ISE}(i) = K^{-1} \sum_{j=1}^{K} \sum_{k=1}^{n} (\hat{g}_k^j(i) - g_k^j)^2. \tag{4.1}$$

For each jitter level and test function, the sparsity plot consists in the curve obtained by making the correspondence between the number of (large) details (i) included in the inverse transform and the overall amount of error in the reconstructed signals ($\mathrm{ISE}(i)$), see for instance figure 4.6.

As previously mentioned, we will also employ the same procedure, but using classical wavelets. More exactly, first take a regular grid and sample the test functions on it. Then fully decompose the signal using the DWT with a Daubechies wavelet with a specified number of vanishing moments, and arrange the wavelet coefficients in a decreasing sequence according to their absolute size. Sequentially invert the DWT as explained above, starting with the largest wavelet coefficient in absolute value, and then gradually introducing the detail of the next size down.

For our lifting algorithms, for each combination of test signal and level of jitter, we repeated this procedure several times. The source of variation in this case comes from the grid irregularity. In the classical wavelet approach however, the grid is fixed through its regularity, so there is no point in repeating the same procedure several times. Consequently, the curve obtained in the sparsity plot for classical wavelets is not an average curve.

At a first thought it might seem that the integrated squared error should

decrease with increasing i since we bring in more information. This is indeed the case when using classical orthogonal bases of wavelets— the orthonormality of the wavelet basis ensures the norm equivalence between the (reconstructed) signal and its wavelet coefficients. Hence the more coefficients are brought into the description, the smaller the error is. As we have seen in the previous chapter, second generation wavelet bases are not guaranteed to be stable bases, and consequently there is no norm equivalence between the signal and its wavelet coefficients. Since the reconstruction error in the signal cannot be tightly bound by the error in the wavelet coefficients, there is no guarantee that using say 15 largest coefficients is surely an improvement over using 12 largest coefficients. This explains the vague 'wiggliness' that appears in the sparsity curves associated to our transforms.

4.2.2 Sparsity results

Analysing the sparsity plots corresponding to each type of signal, jitter and lifting 'one coefficient at a time' algorithm, we conclude that the adaptive transforms produce sparser results than the algorithms using a fixed type of prediction (as an example, refer to figure 4.2). We recommend working with techniques that do not use large neighbourhoods, due to stability issues. Therefore our results and figures will mostly concentrate on presenting AP1S, AP2N and AN1.

Out of the linear lifting algorithms, LP1S, LP2N generally give the best compression.

Our adaptive techniques are also competitive with those using classical wavelet bases on regular grids (see figures 4.5 and 4.6), with the added benefit that they can work on any type of observations and do not require the specification of a certain wavelet, which can seriously affect the sparsity of the details.

The irregularity in the locations does not dramatically affect the sparsity

of the outcome, as it can be noticed for instance in figures 4.3, 4.4.

The type of signal however does influence the sparsity of its wavelet coefficients, and zero reconstruction error is achieved much faster for smoother signals than for signals that present many discontinuities— see for example the difference in the 'drop' rate in figure 4.2 between *HeaviSine* and *Blocks*.

We have also decomposed our test signals sampled on irregular grids using the method introduced by Kovac and Silverman (2000), suitable for any type of signals (see section 4.3 for its complete description). We will refer to this method as KS.

KS works by first performing an interpolation of the original data onto a regular grid, and we then use the same procedure as in the case of classical wavelets on regular grids for assessing sparsity. However, the signal is reconstructed at the new regular locations, and the quality of interpolation influences the way it compares to the initial (true) signal. If the interpolated function values are not a good approximation of the true values, then the sparsity curve will not tend to zero, even when a reconstruction using all details is performed. The lack of smoothness of the signal will accentuate this phenomenon— compare for example figure 4.5 to figure 4.6.

Our analysis leads us to conclude that

- The adaptive lifting algorithms produce sparse wavelet representations.
- Our proposed methods are competitive when compared to classical wavelet-based techniques, and have the benefit of not having to deal with a pre-choice of wavelet.
- The wavelet coefficients constructed adaptively are sparser than the ones obtained by a non-adaptive lifting algorithm.
- Sparsity of the wavelet coefficients is not adversely affected by an increase in grid irregularity.

Since sparsity of the wavelet coefficients is an essential ingredient in wavelet

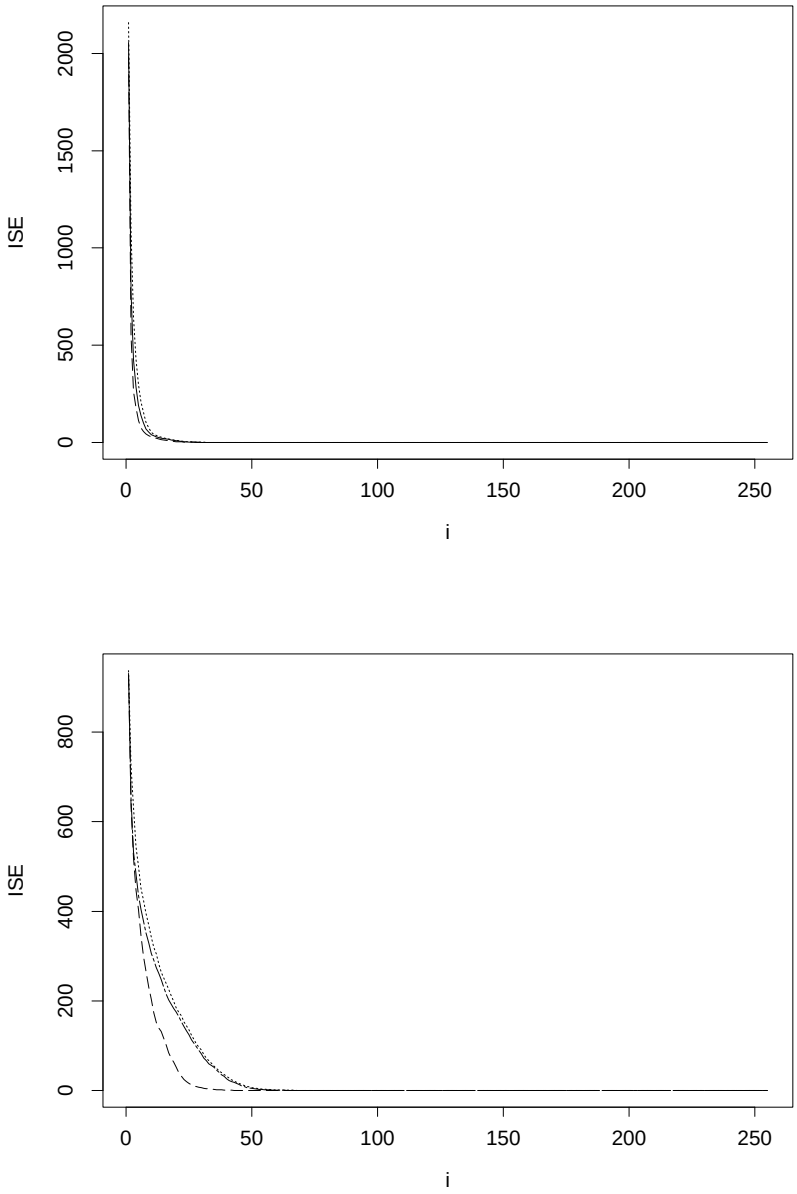

Figure 4.2: Top: Comparison between LP1S (dotted), AP1F (solid) and AN1 (dashed) on *HeaviSine* sampled on an irregular grid ($d = 0.01$). Bottom: Comparison between LP1S (solid), AP2N (dot-dashed) and AN1 (dashed) on *Blocks* sampled on an irregular grid ($d = 1$).

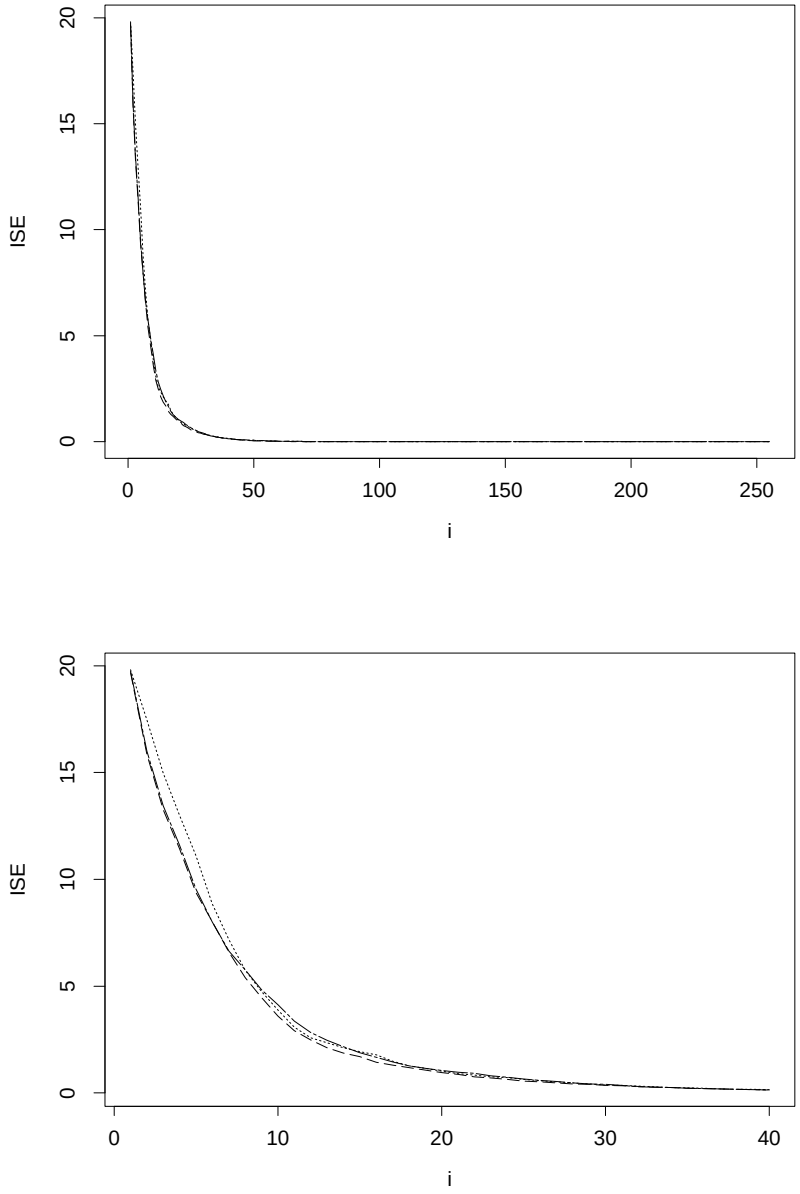

Figure 4.3: Top: Sparsity plot for *Doppler* signal using AP2N over the three different jitter values: d_1 (dotted); d_2 (dot-dashed); d_3 (dashed). Bottom: Blowup of the above figure.

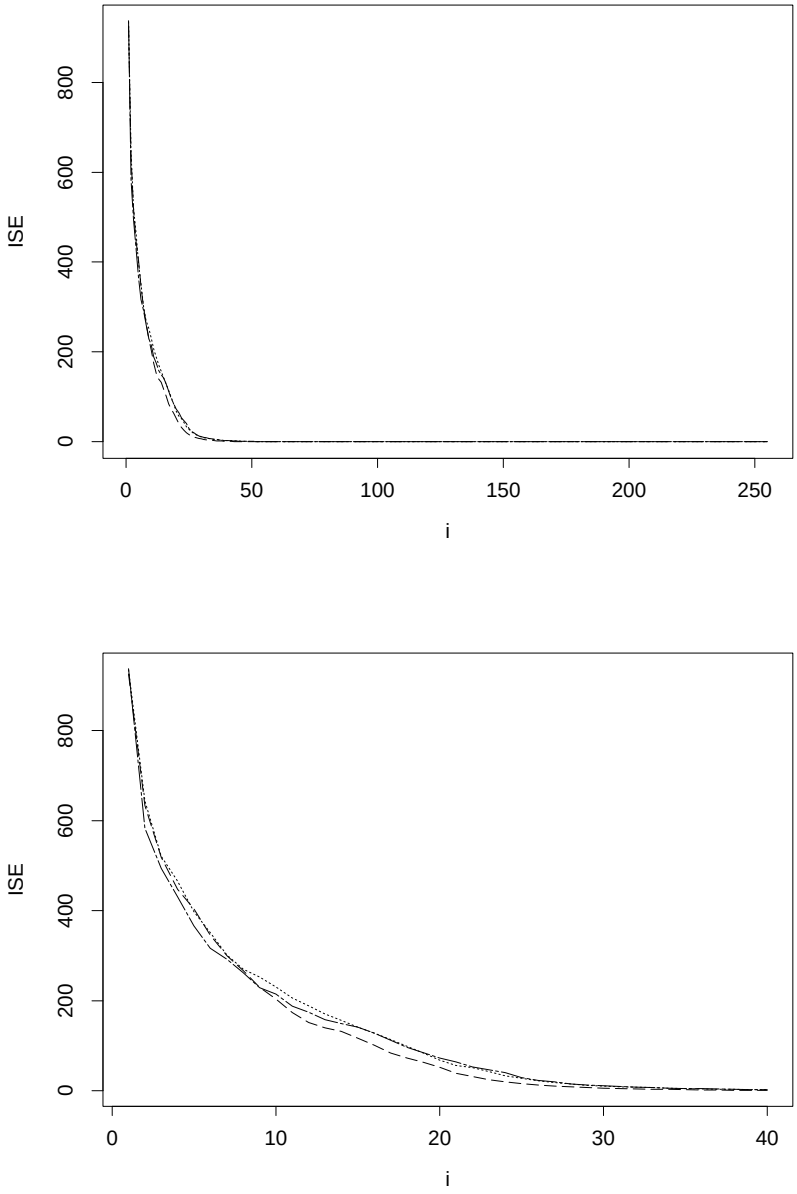

Figure 4.4: Top: Sparsity plot for *Blocks* signal using AN1 over the three different jitter values: d_1 (dotted); d_2 (dot-dashed); d_3 (dashed). Bottom: Blowup of the above figure.

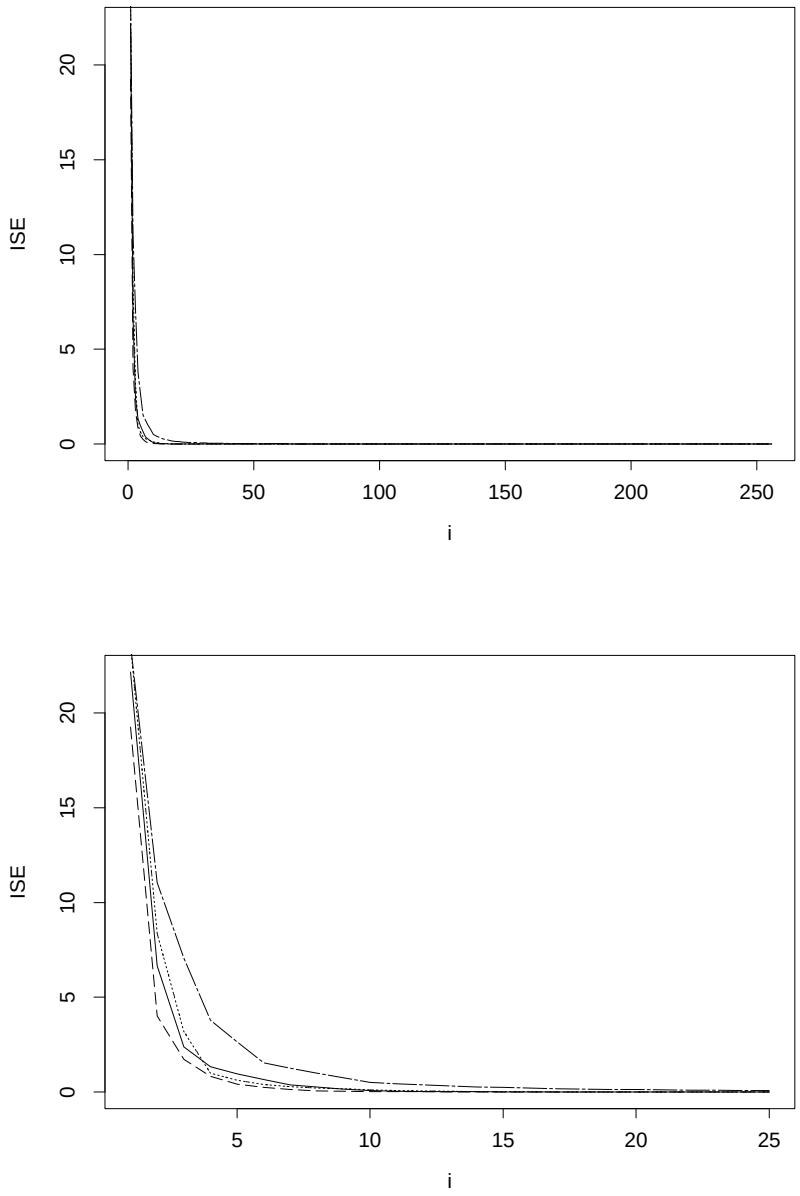

Figure 4.5: Top: Comparison between AP2N (solid), AN1 (dashed) and KS using D5 wavelets (sparse dotted) on *Ppoly* sampled on an irregular grid ($d = 0.1$). Regular grid for Daubechies' Extremal Phase wavelets: best sparsity attained with D5 wavelets (dotted), worst sparsity was Haar wavelets (dot dashed). Bottom: Blowup of the above figure.

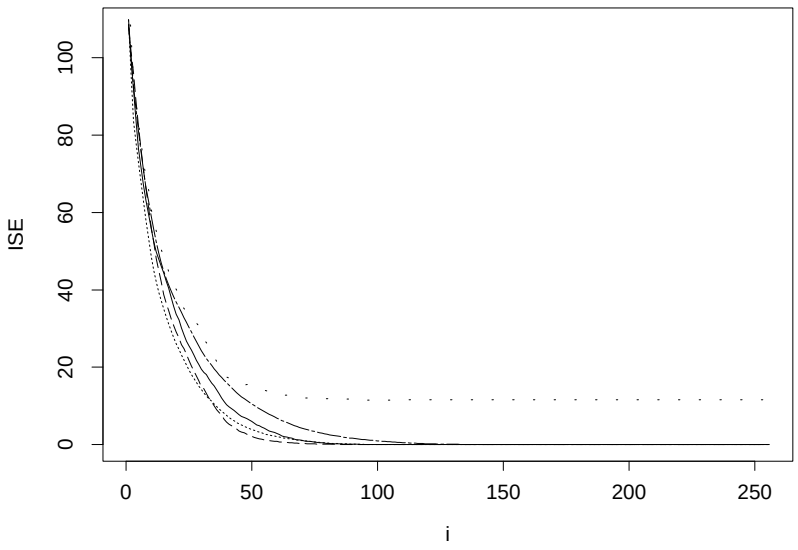

Figure 4.6: Comparison between AP1S (solid), AN1 (dashed) and KS using D2 wavelets (sparse dotted) on *Bumps* sampled on an irregular grid ($d = 0.01$). Regular grid for Daubechies' Extremal Phase wavelets: best sparsity attained with D2 wavelets (dotted), worst sparsity was D10 wavelets (dot dashed).

shrinkage, we can now proceed onto investigating the denoising performance of our adaptive transforms.

4.3 Denoising performance

While the 'true' signals are all we need for investigating the sparsity of our transforms, in order to assess the denoising performance, we shall first construct 'noisy' versions of the test signals.

Noisy versions of the test functions. In what follows we describe our construction for obtaining noisy versions of the signals introduced in section 4.1. To ensure comparability with other studies, such as the one in Barber and Nason (2004), the signal shall first be normalized as follows.

The new 'true' signal is given by $(x_i, g_i/\sqrt{\text{var}(\underline{g})})_{i\in\overline{1,n}}$. For ease of notation, we shall denote $\tilde{g}_i = g_i/\sqrt{\text{var}(\underline{g})}$ for all $i \in \overline{1,n}$.

Then we generate three levels of homogeneous noise, $\varepsilon_i \sim N(0, \sigma_\varepsilon^2)$, by

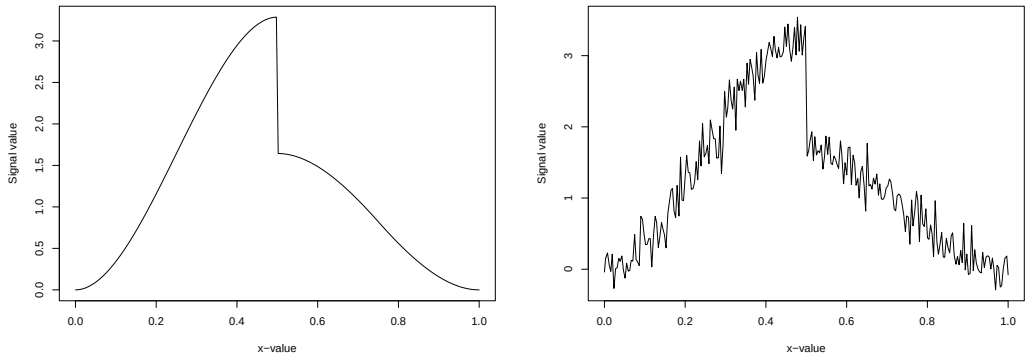

Figure 4.7: Plot showing on the left the normalized *Ppoly* test function, sampled at 256 irregular locations ($d = 0.1$), and its noisy version with SNR=5.

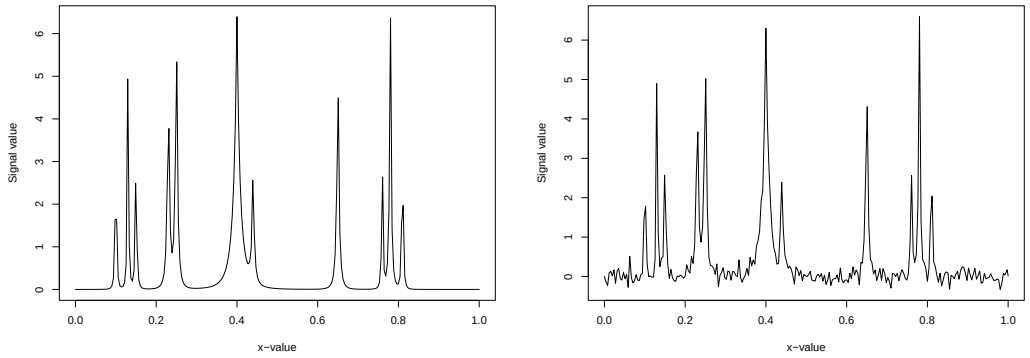

Figure 4.8: Plot showing on the left the normalized *Bumps* test function, sampled at 256 irregular locations ($d = 0.01$), and its noisy version with SNR=7.

setting the signal-to-noise ratio SNR $= \sqrt{\text{var}(\underline{\tilde{g}})}/\sigma_\varepsilon$ to take values 3, 5 and 7 (where by construction $\text{var}(\underline{\tilde{g}}) = 1$).

The noise obtained as such will be added onto the original signal, creating a noisy version of it $(x_i, f_i)_{i\in\overline{1,n}}$, where $f_i = \tilde{g}_i + \varepsilon_i$.

Figures 4.7 and 4.8 show noisy versions of the test functions, corresponding to SNR=5, respectively SNR=7. The lower the SNR, the higher the noise level is, and the more difficult the task of denoising will be.

4.3.1 Comparisons performed

Once a decision has been made upon the lifting transform to be used for decomposing the signal $(x_i, f_i)_{i\in\overline{1,n}}$, we shall employ the modified version of the empirical Bayesian thresholding procedure with posterior median to try and filter the noise out (see section 3.6.2 of previous chapter for details on the method and its adaptation to our transform). Simulations in Nunes and Nason (2005) have shown that an exact choice of resolution level in the adaptive lifting algorithms using the modified empirical Bayes thresholding procedure is not essential and a decomposition carried to a low enough level should provide good results. Therefore with the adaptive lifting algorithms a full decomposition of the signal is carried out, and the obtained $n-2$ wavelet coefficients are then subjected to thresholding.

In order to quantify the performance of the used lifting method, for each grid type and each level of noise we performed the denoising procedure $K = 100$ times, i.e. for K grids of size n pertaining to location jitter d_l, we sample a test function to obtain $\underline{\tilde{g}}^k$, and then add noise at a set level, to generate noisy observations $\underline{f}^k$. Then decompose the noisy signal $\underline{f}^k$, threshold the obtained detail coefficients and obtain an estimate of the true function, $\underline{\hat{\tilde{g}}}^k$. A measure of the overall accuracy of the estimates is given by the average mean square error,

$$\text{AMSE} = (nK)^{-1} \sum_{k=1}^{K} \sum_{i=1}^{n} (\hat{\tilde{g}}_i^k - \tilde{g}_i^k)^2. \tag{4.2}$$

Additionally, we tested and compared our methods against *Locfit* of Loader (1997, 1999), the smoothing spline function implemented in S-Plus, `smooth.spline()` (Green and Silverman (1994)), the Comte-Rozenholc method (Comte and Rozenholc (2004)) and the Kovac-Silverman wavelet procedure (Kovac and Silverman (2000)).

Let us now briefly discuss the methods we chose to compare with ours.

Locfit is a local regression technique capable of working on irregularly spaced

data with multiple observations at the same location. The unknown function is assumed to be smooth (Loader (1997), page 1), and we will see this assumption reflected in the results obtained when denoising curves with severe jumps using *Locfit*. A polynomial model is fitted within a sliding window along the signal, and a weighted least squares criterion is used for locally estimating the curve. The window bandwidth acts as a smoothing parameter— too large, and the estimate will be oversmooth, too narrow and the estimate will preserve too many of the original (noisy) features. In order to optimise *Locfit*'s behaviour, we will choose the window bandwidth by cross-validation. In tables 4.1–4.3 this procedure appears under the name of *Locfit*. The code can be found at `http://cm.bell-labs.com/stat/project/locfit` and can be run under S-Plus.

`smooth.spline()` function of S-Plus fits a cubic spline (i.e. a piecewise cubic polynomial with continuous second derivative, Green and Silverman (1994)) to the data, and is also able to work on irregular grids and multiple observations. The spline function is estimated by minimizing a criterion that involves both the variance and the bias of the estimate (hence it is similar to the mean square error), but in unequal proportions— minimize $\sum_i (f_i - \tilde{g}(x_i))^2 + \lambda \int \tilde{g}''(x)^2\, dx$. The parameter λ sets the balance between the fidelity to the data and the curve smoothness, so it is equivalent to the window bandwidth in the case of local regression. In our simulations we choose λ by cross-validation to ensure an optimal fit of the estimate to the data. This procedure is labelled SSCV in tables 4.1–4.3.

The Comte-Rozenholc method (CR) is a data-driven algorithm that aims to estimate the true (unknown) regression function $\tilde{g}$ by fitting to the data piecewise standard and trigonometric polynomials. Their procedure is capable to handle irregularly spaced locations and multiple observations,

and it (automatically) searches for the model (i.e. number and position of the knots, polynomial degrees, up to a maximum of 74, associated to each interval and type of polynomial) that produces an optimal estimator for $\tilde{g}$, in a sense to be specified next. For each model, an estimator of $\tilde{g}$ is computed by minimizing the usual least squares criterion, over all functions in the class of piecewise polynomials defined by Comte and Rozenholc (2004). Out of the obtained collection of estimators, one will be chosen based on a criterion that realizes a trade-off between the bias and variance terms (see Comte and Rozenholc (2004) for details). The Matlab code for this method has been kindly supplied by Yves Rozenholc, and in our simulations we allowed for searches up to the maximal polynomial degree, 74. In tables 4.1–4.3 the results obtained by using this method appear on the row labelled CR.

The Kovac-Silverman procedure (KS) is a wavelet technique able to work on irregular grids, so it is suitable for comparison with our adaptive lifting methods. However, for data with multiple observations at a single location, the user must decide how to handle them. A description of this method can be found in section 3.2 of the previous chapter. It is important to note at this point that KS produces function estimates at interpolated regularly spaced locations, rather than at the original irregular locations as our adaptive methods do. Hence the AMSE values for KS are calculated on the regular interpolated grid, and therefore they are only approximately comparable to ours, which are computed at the initial (irregular) data points. With KS, as with all classical wavelet based methods, several choices have to be made: wavelet family and wavelet smoothness, primary resolution level used in the wavelet decomposition and finally, the denoising technique. For a comprehensive study, we investigated the members of the Daubechies Extremal Phase wavelet family with vanishing moments from one to ten (D1,...,D10). Furthermore, since Kovac

and Silverman (2000) recommend using SureShrink for thresholding the wavelet coefficients, and since empirical Bayesian thresholding (implemented in EbayesThresh, Johnstone and Silverman (2004b)) is a more recent methodology that proved to have good theoretical and practical properties, we tested both techniques: EbayesThresh with posterior median thresholding and SureShrink. Johnstone and Silverman (2005) comment that EbayesThresh is largely insensitive to the choice of primary level, as long as this is low enough, so we fully decomposed the signal when using it. When using SureShrink, we decomposed the noisy function using all possible primary resolution levels. The software that implements the KS procedure is available in the R-package WaveThresh (Nason (1998)), available at `http://www.stats.bris.ac.uk/~wavethresh`.

Note that unlike KS, our adaptive algorithms automatically choose the wavelet to use, so this choice no longer rests with the user. The choices that bear similarity to the classical wavelet denoising methods are the primary resolution level, and of course that of thresholding method. Due to the characteristics of EbayesThresh and of our adaptive lifting, as mentioned before we will fully decompose the signals down to two scaling coefficients when using our method. Our algorithms are available in the R-package Adlift, at `http://www.stats.bris.ac.uk/~maman/computerstuff/Adlift.html`.

4.3.2 Simulation results on Kovac-Silverman method

Tables 4.1–4.3 report the AMSE values ($\times 10^3$) obtained for our linear and adaptive lifting methods that do not use large neighbourhoods, and also for the 'competitor' methods described in the previous section.

However, for the KS method, we only report the best results obtained for certain (wavelet, primary resolution level, thresholding technique) combinations. More exactly, the AMSE values are reported for the following choices *Blocks*: (D1, 2, SureShrink), *Bumps*: (D2, 0, EbayesThresh), *HeaviSine*: (D4,

4, SureShrink), *Doppler*: (D4, 5, SureShrink) and *Ppoly*: (D5, 4, SureShrink). It is important to realise though that in practice we would not know which combination is best, and the choice can seriously influence the final performance of the estimator (so, in practice the KS results would be somewhat worse).

In order to assess the combination (wavelet, primary resolution level, thresholding technique) that produces the best estimates using KS, we carried out a separate simulation study: for each test function, each wavelet (D1,...,D10) and each primary resolution level for SureShrink, or maximum resolution level for EbayesThresh, we simulated data sets of length $n = 256$ on the three types of irregularly spaced locations (corresponding to jitters d_1–d_3) with three noise levels (corresponding to SNR 3, 5 and 7). For full details on the simulation study for KS, the reader can refer to the technical report accompanying Nunes *et al.* (2004).

Simulations showed a high degree of variation in estimation accuracy across the wavelet smoothness and primary resolution when using SureShrink. With EbayesThresh the results are much more homogeneous across wavelets, but the performance is poorer than SureShrink on all test signals except for *Bumps*. We conclude that

- For *Blocks* signal, SureShrink with Haar wavelet (D1) and primary levels 2, 3 and 4 give the best results. The wavelet smoothness appears to influence more the estimation accuracy than the primary level.

- For *Bumps*, EbayesThresh with any wavelet D1 up to D4 give best results, and the AMSE values are generally quite homogeneous across wavelet smoothness.

- For *HeaviSine* test function, more combinations become available with decreasing the noise level. However, for all SNR's, SureShrink using any wavelet D3 up to D10 with primary resolution level 4 are good choices.

Table 4.1: AMSE ($\times 10^3$) simulation results for test signals with SNR=3 with three levels of jitter, d_ℓ, for various denoising methods described in the text.

	Blocks			*Bumps*			*HeaviSine*			*Doppler*			*Ppoly*		
Method	d_1	d_2	d_3	d_1	d_2	d_3	d_1	d_2	d_3	d_1	d_2	d_3	d_1	d_2	d_3
LP1S	72	71	68	81	80	73	20	20	21	54	53	52	16	16	18
LP2N	70	73	67	84	83	73	20	20	22	55	56	51	16	16	17
AP1S	72	68	59	77	77	62	20	20	23	52	50	48	16	17	18
AP2N	69	70	59	78	75	64	21	21	22	53	52	48	15	16	17
AP3N	69	68	68	76	74	73	46	44	41	64	65	61	42	39	36
AN1	55	54	52	66	67	61	36	39	37	61	61	59	38	33	32
Locfit	73	72	64	110	108	101	11	11	11	58	58	54	21	20	19
SSCV	74	74	67	307	315	250	12	11	12	61	60	53	20	20	19
KS	79	78	87	179	181	259	13	12	15	51	52	57	18	17	18
CR	119	119	133	332	313	284	25	25	25	155	155	148	13	13	13

Table 4.2: AMSE ($\times 10^3$) simulation results for test signals with SNR=5 with three levels of jitter, d_ℓ, for various denoising methods described in the text.

	Blocks			*Bumps*			*HeaviSine*			*Doppler*			*Ppoly*		
Method	d_1	d_2	d_3	d_1	d_2	d_3	d_1	d_2	d_3	d_1	d_2	d_3	d_1	d_2	d_3
LP1S	24	25	22	31	28	27	10	10	10	23	23	23	6	6	7
LP2N	23	23	22	30	30	27	10	10	11	23	23	22	6	6	6
AP1S	22	23	20	30	29	23	10	10	10	22	22	21	6	6	7
AP2N	23	23	20	30	29	23	10	10	11	22	21	21	6	6	7
AP3N	27	27	26	30	30	29	18	18	16	26	26	26	16	15	14
AN1	19	20	18	26	26	24	15	16	16	25	24	24	13	13	12
Locfit	35	35	34	40	40	39	7	7	7	25	26	25	12	12	11
SSCV	51	51	46	277	285	227	7	7	7	37	37	30	11	12	11
KS	52	52	59	130	134	213	8	7	8	29	28	33	9	9	10
CR	85	85	101	288	272	247	23	23	23	136	137	133	11	11	12

- On *Doppler*, SureShrink with wavelets D4 and D5, both with primary resolution levels 4 and 5 provide good results across all SNR's. On SNR=7 more choices are available.

- On *Ppoly*, the best combination (wavelet, primary level) changes within SureShrink with the noise level, and a compromise is D5 with primary resolution 4.

Table 4.3: AMSE ($\times 10^3$) simulation results for test signals with SNR=7 with three levels of jitter, d_ℓ, for various denoising methods described in the text.

	Blocks			*Bumps*			*HeaviSine*			*Doppler*			*Ppoly*		
Method	d_1	d_2	d_3	d_1	d_2	d_3	d_1	d_2	d_3	d_1	d_2	d_3	d_1	d_2	d_3
LP1S	11	11	11	15	15	14	6	6	6	12	12	13	3	3	3
LP2N	11	11	10	16	15	13	6	6	6	13	12	12	3	3	3
AP1S	10	10	10	15	14	12	6	6	6	12	12	11	3	3	4
AP2N	11	10	10	15	14	12	6	6	6	12	12	12	3	3	4
AP3N	14	14	14	16	16	16	10	10	10	14	14	14	8	8	7
AN1	10	10	9	14	14	13	9	9	8	14	14	13	7	7	6
Locfit	20	20	19	21	20	20	5	5	5	13	13	16	9	9	8
SSCV	44	44	39	269	273	220	5	5	5	30	30	23	8	8	8
KS	45	45	52	119	122	195	6	6	5	22	22	25	5	5	6
CR	78	79	92	280	269	230	22	22	22	132	130	128	11	11	11

4.3.3 Simulation results and comparisons for adaptive lifting

Let us now carefully analyse tables 4.1–4.3. The simulation results indicate that

- Our adaptive methods perform very well, with AN1 being the best choice for signals presenting sharper discontinuities, such as *Blocks* and *Bumps*, while AP methods with 2 neighbours are suitable for smoother signals, such as the test functions *HeaviSine*, *Doppler* and *Ppoly*.

- The irregularity in locations does not influence the denoising performance of our methods. With decreasing the noise level though, of course we obtain better estimates of the true function.

- On the more discontinuous test functions, *Blocks* and *Bumps*, AN1 outperforms all four competitors across all noise and irregularity levels. *Locfit* comes closest to AN1 and its behaviour gets better with decreasing the noise level, but it still underperforms when compared to AN1.

- On the smoother *HeaviSine*, our adaptive method (AP1S/AP2N) is outperformed by all three competitors except for CR for SNR 3 and 5, while

on SNR=7 all methods have similar denoising performances, except for CR which underperforms all of them.

- For *Doppler*, AP1S, AP2N display better denoising capability than their competitors, with KS being the only method that approaches our results at SNR=3, while for SNR's 5 and 7, *Locfit* comes closest to it.

- Finally, on *Ppoly*, our adaptive method (AP with neighbourhood of size 2) gives better results than all its competitors, and KS comes closest to it. Only for SNR=3, CR outperforms our adaptive method.

It would be now useful to visualise the differences in the denoised signals from figures 4.7 and 4.8, when employing the above methods. Refer now to figures 4.9 and 4.10. For *Ppoly*, the estimated curve obtained by using our adaptive method AP1S, corresponds to a mean square error (MSE) of 3, considerably lower than than the MSE for the other estimated curves (12-13). Note that our estimate picks well the location and magnitude of the jump, as well as the true function's behaviour immediately after it. Some small artifacts however, are displayed along the curve. Visually, amongst the competitors, the closest to the true curve is the one estimated using KS, but none of them accurately picks the steepness of the discontinuity. In figure 4.10, we notice that *Bumps* is a much more difficult signal to denoise due to its discontinuities. In terms of the MSE, AN1 provides best results (12), followed by *Locfit* (17). The estimate obtained by `smooth.spline` is worst— note the inaccuracies in the estimated height of peaks and number. KS also gets some of the peak heights wrong, and it seems completely unable to smooth the areas between them.

4.3.4 Comparison on a modified version of *HeaviSine*

Delouille *et al.* (2004) perform a comparison between the lifting methods introduced in their paper, and the method of Antoniadis and Fan (2001) (to which

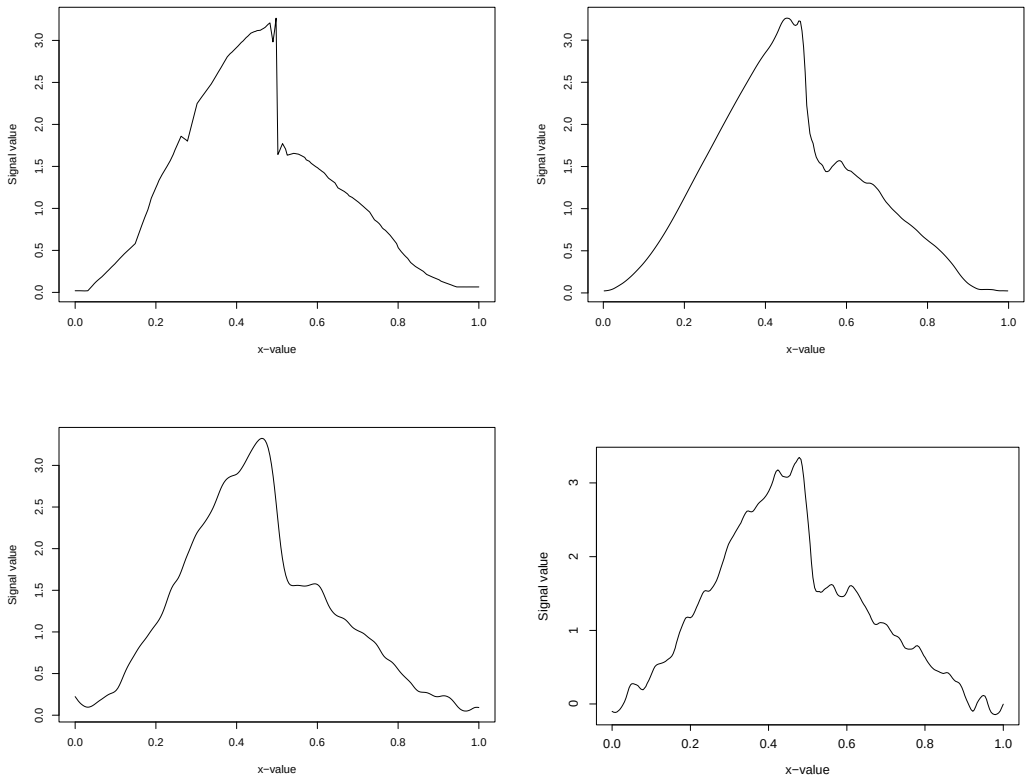

Figure 4.9: Denoised versions of the noisy *Ppoly* signal shown in figure 4.7. Top left: estimated curve obtained using AP1S and EbayesThresh with posterior median thresholding, top right: KS estimate using Daubechies' D5 wavelet, 4 primary resolution levels and SureShrink. Bottom left: estimated curve obtained with smoothing spline with cross-validated smoothing parameter, bottom right: denoised curve using *Locfit* with cross-validated window bandwidth.

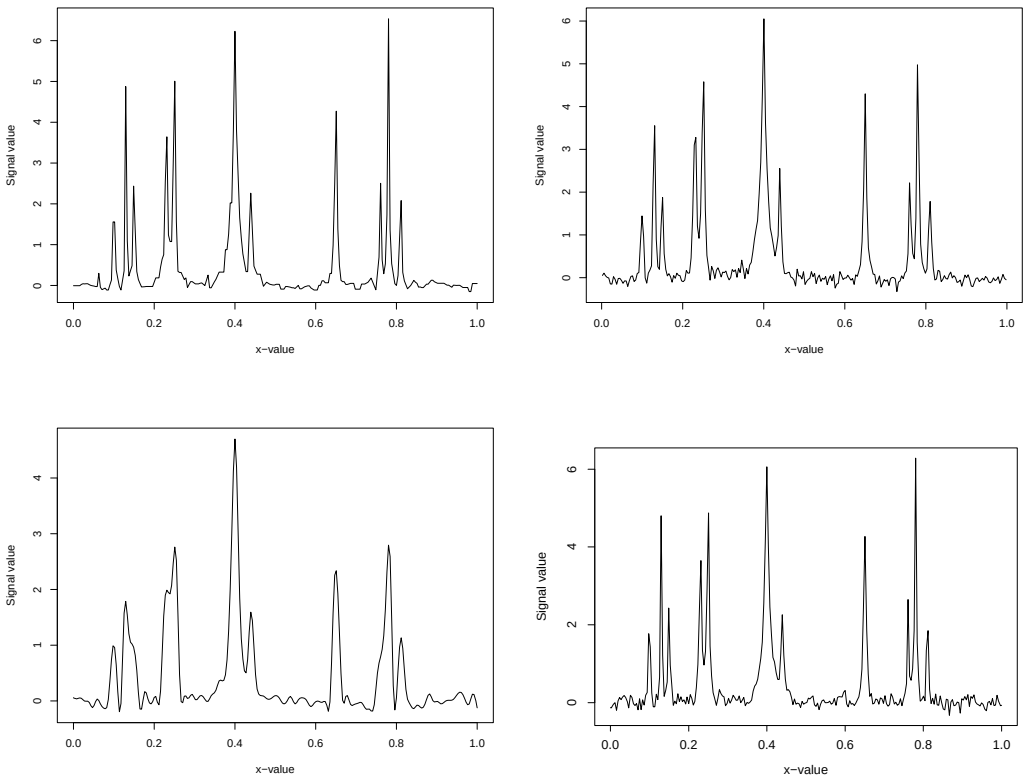

Figure 4.10: Denoised versions of the noisy *Bumps* signal shown in figure 4.8. Top left: estimated curve obtained using AN1 and EbayesThresh with posterior median thresholding, top right: KS estimate using Daubechies' D2 wavelet, 0 primary resolution levels and EbayesThresh. Bottom left: estimated curve obtained with smoothing spline with cross-validated smoothing parameter, bottom right: denoised curve using *Locfit* with cross-validated window bandwidth.

Table 4.4: Results of the Simulation Study, $n = 100$, SNR=4. AN1 result computed here, all other results as computed by Delouille *et al.* (2004). First row: square root of median MSE value; Second row: interval shows square root of 1st and 3rd quartiles of the MSE results over 500 simulations. All results $\times 10^3$.

	Delouille *et al.*				
AN1	With Update	No update	ANTO/FAN	KS	SUPSMO
588	610	792	819	775	706
[517, 654]	[526, 675]	[661, 989]	[759, 875]	[688, 856]	[629, 807]

they refer as ANTO/FAN), KS and the super smoother of Friedman (1984) (SUPSMO). The test function used was a modified version of *HeaviSine* with steeper jumps. The x_i locations follow a normal distribution with mean 0.5 and standard deviation 0.2, and the noise level was set with SNR=4.

We performed 100 simulations on signals constructed as above, using our denoising technique based on decomposing the signal with AN1, and we report the resulting AMSE values in table 4.4. Our method provides the best result, also slightly better than the lifting procedure proposed by Delouille *et al.* (2004).

It should not come as a surprise that while AN1 underperforms KS for the usual *HeaviSine* test signal, it performs better than KS for this modified version of *HeaviSine*, since the discontinuities in this modified signal version are much steeper, and therefore a behaviour similar to that on *Blocks* or *Bumps* was to be expected.

4.3.5 Real data example: the motorcycle experiment

We have already mentioned the data from Silverman (1985), the outcome of an experiment for assessing efficacy of crash helmets. It consists of readings taken through time of an accelerometer that monitored the head of a motorcyclist in a simulated motorcycle crash. The time points are irregularly spaced, there are multiple observations corresponding to some time locations and the observations are subject to error. Figure 4.11 shows the observations; note that the

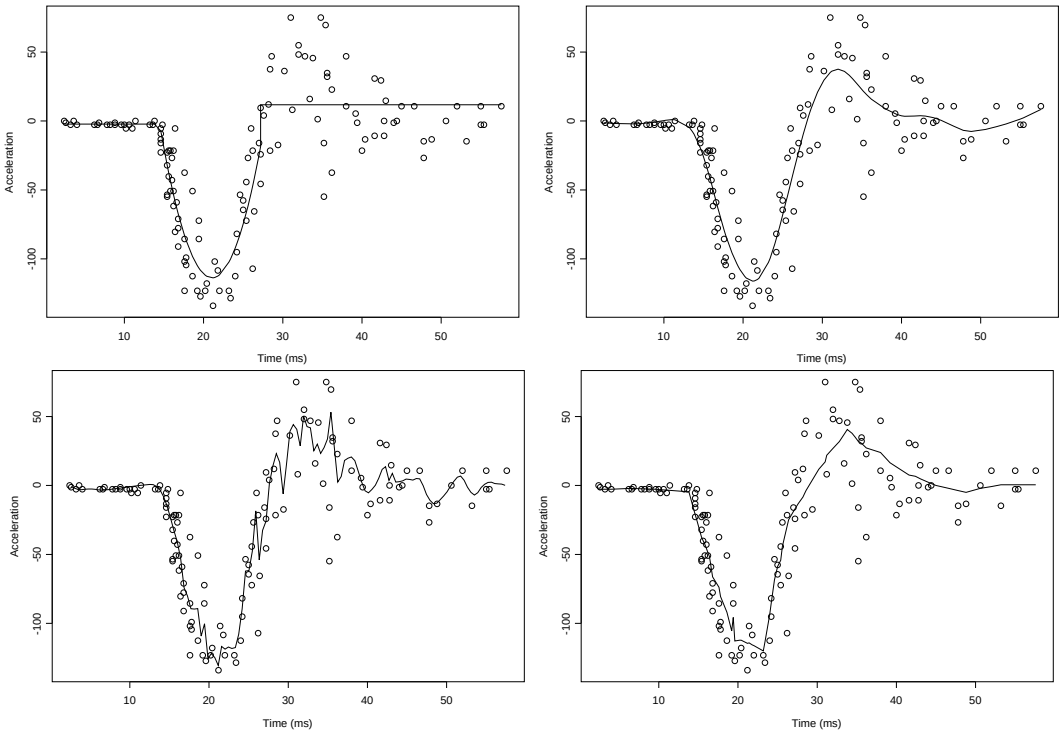

Figure 4.11: Motorcycle crash data and denoised estimates. In each case the small circles show the data and the solid line denotes the estimate. Top left: CR estimate with `rmax`=74, `lmax`=data length, `lmin`=1, `sigma2` and `Dmax` automatically chosen; Top right: smoothing spline with cross-validated smoothing parameter; Bottom left: KS estimate, multiple y-values are averaged resulting in 94 points input to KS, D6 Daubechies' wavelet, primary resolution of 3 and SureShrink thresholding; Bottom right: adaptive lifting using AP1S with heteroscedastic variance computation (see section 3.6.2), EbayesThresh posterior median thresholding.

variance of the data appears to be heterogeneous, hence it will be treated as such.

Our estimate shows most similarities to the ones produced using `smooth.spline` and *Locfit* with cross validated smoothing parameters (we do not show here the curve obtained by using *Locfit*, as it is virtually identical to the one generated by the smoothing spline approach— see figure 3.1). However, our estimate does not have the same degree of smoothness (as the basis functions are linear) and the peak is estimated to occur at a slightly later time (some estimates produced in Kovac (1998) display the same feature). Also, our curve has a 'glitch' near the bottom, which we do not believe to be a true feature— the same

kind of behaviour can be noticed in denoising *Ppoly*, see figure 4.9. The KS estimate shown here is very noisy, as we used the basic algorithm with no allowance for the changing variance. Kovac (1998) first analyses the data for presence of outliers, which next get removed, and consequently gets a much better looking estimate. Completely contrasting, the CR method estimates a constant behaviour after reaching the peak, estimated to happen at an earlier time than the other methods indicate.

Since the curve we try to estimate describes acceleration in a physical process, we believe that most likely it is a smooth curve, so it might be that *Locfit* and the smoothing spline approach produce the best results here. Our adaptive lifting estimate does a reasonable job in this situation, but its main potential lies in denoising signals with sharp jumps.

4.4 Conclusions

In this chapter we detailed the simulation study through which we investigated the performance of the adaptive lifting algorithms we proposed in the previous chapter. We evaluated the sparsity of their corresponding wavelet decompositions and how successful our algorithms are at denoising signals with various degrees of smoothness. In this process, we identified which methods are appropriate for analysing functions with various smoothness degrees, and then compared their performances against wavelet and non-wavelet methods able to work on irregular data. The simulation results showed that our proposed techniques display a competitive behaviour and many times they outperform their competitors.

Chapter 5

Transmembrane segment prediction

In this chapter we shall examine the problem of predicting hydrophobic segments along the sequence of a transmembrane protein, when no information on the protein is available, other than its primary structure. The work we shall present is based on Knight and Nason (2004).

5.1 Motivation

In the speciality literature, proposed methods for transmembrane segment prediction range from the use of simple regression techniques (Kyte and Doolittle (1982)) to neural networks (Rost *et al.* (1995, 1996)). Established prediction methods are often based upon hydrophobicity analysis, and a section of the current literature is devoted to the use of classical wavelet methods in this context (Lio and Vannucci (2000); Fisher *et al.* (2003)). Such methodology proved successful, but as with all classical wavelet constructions, certain assumptions are needed. In the 'real-life' context of proteins, a serious assumption is that the protein residues are modelled as equally spaced.

Our work is motivated by the intuition that we might improve transmembrane segment prediction if we were somehow able to take into account the

resolved three dimensional (3D) information available in proteins that were similar to the protein of interest. Therefore, in our approach we shall challenge the assumption of regularly spaced residues, and construct family-based dissimilarity matrices for estimating the distance between residues. The matrices will be derived from resolved 3D structures of similar aligned proteins. Furthermore, this new construction will call for wavelets able to work on irregular settings. For predicting the highly hydrophobic segments, we will propose methods that involve adaptive second generation wavelets, constructed following the approach in chapter 3. We shall show that incorporating 3D resolved structure by introducing irregular distances improves prediction both in terms of the existence of predicted segments compared to experimentally determined ones, and also the proportion of correctly predicted segments.

Outline of the chapter. Briefly, this chapter is structured as follows: we shall start by introducing a few basic notions on proteins, tailored to the needs of this chapter. Then we shall formulate in detail the problem we address, review existing approaches for solving it and propose our methodology. We provide a study on proteins from Rost *et al.* (1995), where we shall analyse, compare and discuss our proposed methods. Comparisons with methods involving classical wavelets proposed in the current literature will be provided, and we shall demonstrate the superiority of our methods.

5.2 An introduction to proteins

Amino acids. The building blocks of proteins are amino acids. They are organic molecules that contain a carbon atom (known as the alpha carbon, C_α) bonded to a hydrogen atom (H), an amino group (NH_2), a carboxyl group (COOH) and a side chain (R). The structure of the R-group is the one that determines the chemical identity and special properties of each amino acid. Naturally, different amino acids have different properties, such as hydrophobic or hydrophilic, basic or acidic, sulfur containing etc. Many amino acids fall into

more than just one group, because their side chain can have several properties.

There are 20 different amino acids used in synthesizing proteins— Alanine (ALA), Arginine (ARG), Asparagine (ASN), Aspartic acid (ASP), Cysteine (CYS), Glutamine (GLN), Glutamic acid (GLU), Glycine (GLY), Histidine (HIS), Isoleucine (ILE), Leucine (LEU), Lysine (LYS), Methionine (MET), Phenylalanine (PHE), Proline (PRO), Serine (SER), Threonine (THR), Tryptophan (TRP), Tyrosine (TYR) and Valine (VAL).

Peptide bonds. Amino acids have the property of reacting inter-molecularly through the two groups with opposite chemical characters, the amino and carboxyl groups— with elimination of water, a covalent bond is formed between these groups. This bond is known as a peptide bond, and what is left after the elimination of water is called the (amino acid) *residues*.

This way, more amino acids can be joined together, resulting in a peptide chain. Longer peptides (usually containing more than fifty amino acids) are known as polypeptide chains. A protein is made up of one or more polypeptide chains, and it contains tens and hundreds of residues. All proteins have a rigid spatial structure, to which we will usually refer as its 3D structure.

Proteins. A protein is therefore a long chain of amino acids covalently linked through peptide bonds, although many proteins contain more than just one chain.

Polypeptide chains are directional— one end contains the amino group and the other end contains the carboxyl group, hence they can be represented as an ordered sequence of amino acids. The ordered and directional sequence of residues is known as its *primary structure*. Note that the reversed sequence does not correspond to the same molecule, and the same happens for a sequence where two amino acids have been interchanged. As a protein can contain more polypeptides, each chain comes with its own primary structure.

One of the goals of bioinformatics is to predict protein's 3D shape from its primary structure. The 3D structure of some proteins can be experimentally obtained, mostly by using X-ray crystallography or nuclear magnetic resonance

(NMR) spectroscopy. However, for the vast majority of proteins sequence information is all that is available.

So far we have only mentioned the primary structure of a protein. However, proteins are fully characterised by also providing information on their secondary, tertiary and quaternary structures.

The secondary structure of a protein refers to the 3D shape taken by residues which are close to one another. The most common structures are known as alpha helices and beta sheets. In an alpha helix, the residues are arranged such that they resemble a spiral twisting clockwise. Although there is a standard model in which each turn contains 3.6 residues, most helices are somewhat distorted in comparison— they can be tighter (only 3 residues per turn), or longer (4 residues per turn). When the protein chain is extended into an almost linear geometry, the residues are said to form a beta strand. Beta strands run alongside each other (in a parallel or anti parallel fashion), and they are said to form beta sheets. The protein can also change direction, and such regions are known as turns. Some regions simply do not have a regular secondary structure, and they are known as random coil.

Covalent bonds between atoms are not the only forces that act towards obtaining the final protein fold. Hydrogen bonds between residues of the same or different chains, nature (for example aqueous or lipidic) of the medium in which the protein exists, possible formations of disulfide bridges (SH-SH) that arrange two residues close in space even though they may not be close in sequence, are only a few examples of forces that contribute to protein folding. The folding of the total chain, the way in which the elements of the secondary structure are arranged to form the overall 3D structure of that chain is described in the tertiary structure of the protein.

Finally, the quaternary structure refers to the combination of tertiary structures of two or more chains, in forming the complete unit. So the interest at this stage are the inter-, rather than the intra-chain interactions.

Here we have only provided a brief overview of some elementary notions on

proteins. However, for a detailed presentation, the interested reader can refer to Stryer (1988).

5.3 The problem

Membrane proteins (named like this because of their location at the cell membrane) are an important class of protein structures, but experimental determination of their 3D structure can often be very difficult, both for X-ray crystallography and NMR spectroscopy. Therefore, for this type of protein, prediction of various structural aspects using only the information contained in the residue sequence is a problem of interest, see for example Lio and Vannucci (2000).

Membrane proteins can be either intrinsic or integral (they have one or more segments embedded in the plasma membrane) or extrinsic (they are only bound to the membrane indirectly through various interactions). Most integral membrane proteins span the entire lipid bilayer, hence their name of *transmembrane proteins*. Transmembrane proteins, whose 3D structures have been experimentally resolved, have revealed that the transmembranar segments consist of either alpha helices or, less commonly, beta strands.

So far, methods for predicting the locations of transmembrane segments have been primarily directed towards helical transmembrane proteins. Helices embedded in the lipid bilayer primarily consist of residues with hydrophobic R-groups, pointing outwards in the lipid bilayer, and this feature can be used in order to identify them.

Typical methodology for predicting transmembrane helices includes *hydrophobicity analysis*, see for example Kyte and Doolittle (1982), Engelman *et al.* (1986), Lio and Vannucci (2000). However, hydrophobicity analysis as a tool for prediction is not limited to transmembrane segments only, and it has also been used for hydrophobic cores of globular proteins (see Hirakawa *et al.* (1999) for instance).

Our approach is also based on hydrophobicity analysis.

5.4 The hydropathy plot

Hydrophobicity analysis is centred on analysing the hydropathy profile associated to each protein, obtained by using a measure of hydrophobicity associated with each amino acid.

Various measures for the hydrophobicity of amino acids have been constructed (for example the scale of Kyte and Doolittle (1982), or the scale of Eisenberg *et al.* (1984)), and also combined measures of hydrophobicity and helicity to be used in the context of helical transmembrane proteins (see for instance the Lio and Vannucci (2000) scale).

By means of these scales, the primary structure of the protein can be converted into a hydropathy profile: we obtain a signal which on the horizontal axis has the residues in their order of appearance in the protein primary structure, and on the vertical axis their corresponding values from the hydrophobicity index.

After investigating the compatibility of our method with the previously mentioned scales, we decided to use in our study the Kyte and Doolittle measure of hydrophobicity.

As already mentioned, in previous studies the residues were processed assuming that they were equally spaced. However, if each residue is thought of as a 3D structure determined by its atoms, then plausibly one should not automatically consider the distances between any two residues to be equal.

If additionally one was presented with supplementary secondary and tertiary structure information, then precise local information (which typically we do not have) would be gained on the residue positions, and a 3D parametric function could be fitted in order to accurately obtain the inter-residue distances.

We will use the resolved 3D information contained in proteins that are

similar to our protein of interest in order to estimate the residue locations. Our intuition is that making use of this additional information would help improve upon the estimation of the function that 'models' the hydrophobicity level along the protein.

Hence, as explained in the introduction, we will challenge the assumption of equally spaced residues, and estimate the distances between consecutive residues.

5.4.1 The distance matrix

We shall first determine which protein sequences with resolved 3D structure are similar to the sequence of a query protein. This can be established by using local alignment methods that identify regions where the sequences are similar and then score the aligned residues in those regions by means of a scoring matrix. Scoring (or substitution) matrices estimate the probability of a change of each possible residue in a sequence into any another residue. Henikoff and Henikoff (1992) introduced the BLOSUM(BLOcks SUbstitution Matrix)x scoring matrices, computed using blocks of clustered related sequences that share at least x% sequence identity (inside each block). Henikoff and Henikoff (1993) showed that fast alignment methods, such as FASTA (Lipman and Pearson (1985)) or BLAST (Altschul *et al.* (1990)), using scoring matrices derived from distantly related proteins display high accuracy in scoring alignments, with the best results obtained by using BLOSUM62. Therefore, in our approach we used FASTA with BLOSUM62.

Our aim is to use the known 3D structure of aligned protein sequences in order to estimate the distance between each pair of consecutive residues in the primary structure of the protein of interest. We will work with the 3D structures as given in the Protein Data Bank (PDB), available at `http://www.rcsb.org/pdb/`.

First we identify all the appearances of each specific residue pair in the primary structures of the aligned chains, and then compute all the corresponding

Euclidean distances. Their average will provide an estimate for the distance between the two (consecutive) residues. In computing the Euclidean distance between two residues, the x, y, z coordinates given by their corresponding PDB file of all their atoms are used. Typically, distances between atoms are measured in Ångström units, where an Ångström (Å) equals 10^{-10} meters. The result is a 20×20 matrix D, where D_{ij} is measured in Å and it contains the average of the Euclidean distances computed between the residues i and j, from all the aligned chains where they appear in this order.

Let us emphasize here that D is not a symmetric matrix, hence the distance from ARG to LYS, say, is different to that from LYS to ARG.

At this point one might like to refer to Figure 5.1, which gives an indication of the range of estimated distances between different pairs of amino acids, as well as their variation. This distance matrix has been computed using 402 matching proteins, each with various sequence lengths.

The amino acids are clustered according to their R–group nature, and separated in Figure 5.1 by white lines. So, GLY–ILE form the first group of amino acids with aliphatic R–groups, SER and THR are non-aromatic amino acids with hydroxyl R–groups, CYS and MET have sulphur containing R–groups, ASP–GLN are acidic amino acids and their amides, ARG–HIS are basic amino acids, PHE–TRP are amino acids with aromatic rings and PRO is the sole imino acid. It is notable, for example, that pairs consisting of amino acids with aromatic rings typically have low standard deviations, but middling mean distances.

Note that some residue pairs will only appear in the primary structure of the protein being investigated, and not also in the primary structures of the chains aligned to it. In such situations, we use the distances supplied by a family-oriented distance matrix. In the calculation of a family matrix, the chains with determined 3D structure that have been aligned to protein sequences belonging to that family are used. Hence the missing distances for a protein will be imputed from its corresponding overall family distance matrix.

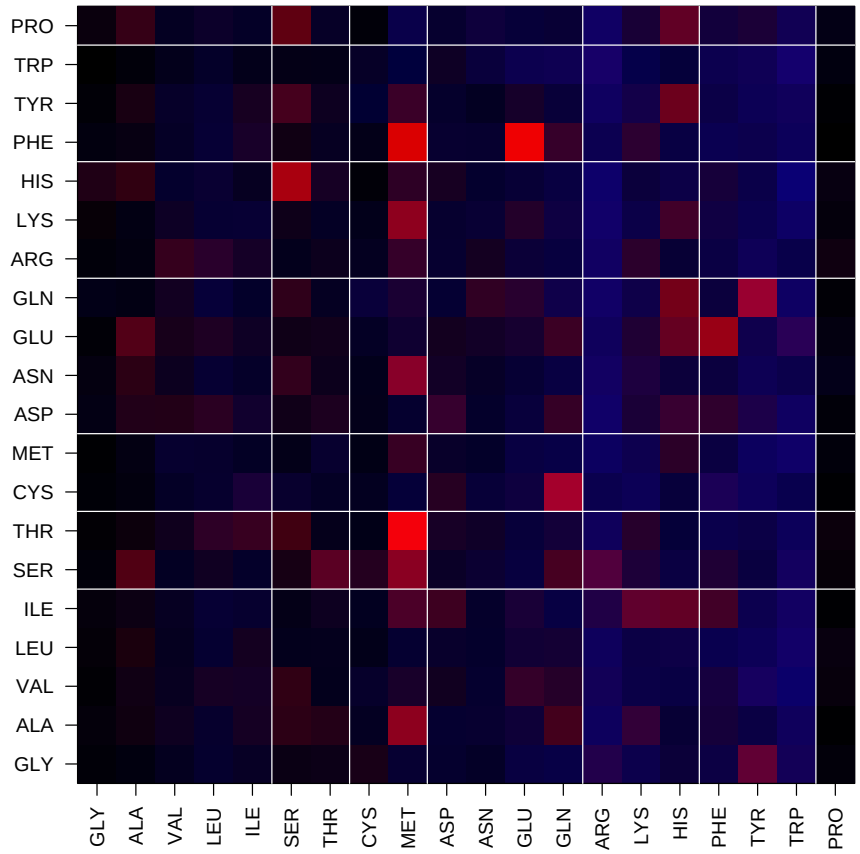

Figure 5.1: Overall 'average' distance matrix (in Ångström units). The intensity of a pixel (dark/light) corresponds to the mean distance for that residue pair. The colour of a pixel (blue/red) corresponds to the standard deviation for that amino acid pair. The brightest pixel in the figure occurs at MET–THR, with a distance of 10.2. It is also one of the most variable, with a standard deviation of 16.7– MET–PHE is the most variable (most red) with a standard deviation of 18.7. The darkest combination is GLY–TRP, with a distance of 4.5, while the least variable is MET–TRP, having a standard deviation of 0.28. The lower and upper quartiles of the mean distance (standard deviation) are 5.2 and 6.1 (0.74 and 2.52).

The distance matrix in Figure 5.1 is less 'specific' and hence less variable than family-based distance matrices. See for instance the family-based distance matrix in figure 5.2, computed using chains with known 3D structure coming from 128 proteins, similar to 22 investigated proteins from the ligand-gated ionic family.

If the sequence of interest has no aligned sequences with resolved 3D structure, then the overall family matrix can be used for estimating inter-residue distances.

5.4.2 Using the distance matrix for constructing the hydropathy plot

Having estimated the distances between each pair of consecutive residues in the primary structure of the protein of interest, we compute a coordinate value for each residue in the chain, based on its distance to the previous residue. The coordinate associated to each residue will indicate its estimated distance to the previous and next residues.

The residues will then be plotted on the horizontal axis according to these coordinates. Therefore, rather than considering them to be equally spaced, they will have an uneven distribution. Let us denote the location (coordinate) corresponding to residue i in the protein chain by x_i, for use in the next section.

A shortcoming of this matrix approach is that it only takes into consideration the distances between consecutive residues, and hence models the protein as being a straight chain, rather than modelling its 3D shape. Overcoming this restriction and trying to estimate the 3D function behind the protein shape is an interesting point for future research, and we suspect that it would bring us even closer to correctly estimating the hydrophobicity level as a function of the protein's amino acid composition and shape.

Figure 5.4 shows an example of hydrophobicity signal constructed as described above. Note that an exact visual assessment of the segments consisting

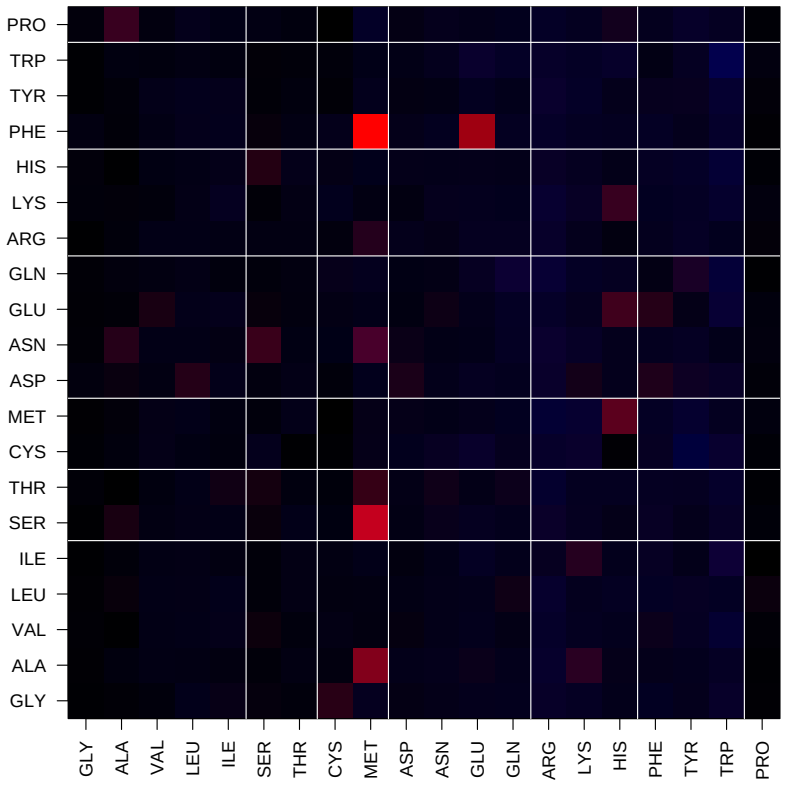

Figure 5.2: Distance matrix corresponding to the ligand-gated ionic family (in Ångström units). The intensity of a pixel (dark/light) corresponds to the mean distance for that residue pair. The colour of a pixel (blue/red) corresponds to the standard deviation for that amino acid pair. The brightest pixel in the figure occurs at MET–PHE, with a distance of 28.2. It is also the most variable, with a standard deviation of 43.56. The darkest combination is CYS–MET, with a distance of 4.4, while the least variable is CYS–CYS, having a standard deviation of 0.12. The lower and upper quartiles of the mean distance (standard deviation) are 5.1 and 6.0 (0.58 and 0.88).

of highly hydrophobic residues is virtually impossible, hence proper statistical tools are needed.

5.5 Wavelets and the hydropathy profile

Since transmembrane helices are sequences of predominantly hydrophobic residues, our purpose is to detect the points at which sharp changes occur in the hydrophobicity signal. This amounts to modelling the profile as noise-contaminated, and estimating the underlying signal.

We model each of our hydrophobicity observations as $f_i = g(x_i) + \varepsilon_i$, where n is the length of the protein chain of interest, f_i denotes the observed (according to the chosen scale) hydrophobicity level of residue i (located at x_i), $g(x_i)$ is the true hydropathy value of the same residue, to be estimated, and $\{\varepsilon_i\}_i$ models identically distributed, independent noise, assumed to follow a $N(0, \sigma^2)$ distribution. In other words, based on just one observation, f_i, at each location x_i in the protein chain, we want to estimate the true signal value, $g_i = g(x_i)$. The assumption of independent observations is a necessary mathematical requirement, which we are aware that is likely to hold only approximately in the biological context.

We shall address the statistical problem of denoising the above hydropathy signal ($\{f_i\}_{i \in \overline{1,n}}$) by using wavelet methodology. So far, wavelet based smoothing methods using classical wavelets (Lio and Vannucci (2000), Fisher *et al.* (2003)) have been used and shown to perform well in the task of transmembrane segment prediction.

However, as the construction of our hydropathy profile uses an uneven distribution for residue locations, we shall use the adaptive lifting scheme introduced in chapter 3 in order to construct second generation wavelets that adjust to the protein hydropathy features. As explained in section 3.6.2 of chapter 3, by means of these wavelet functions, the noise corrupted signal will be converted into $d_j = d_j^* + e_j$, with $\{d_j\}_j$ being the observed wavelet co-

efficients, $\{d_j^*\}_j$ the 'true' wavelet coefficients and $\{e_j\}_j$ the transform of the noise $\{\varepsilon_j\}_j$. Estimates $\{\hat{d}_j^*\}_j$ of the true wavelet coefficients will be obtained by shrinking the observed details using the modified empirical Bayesian technique described in 3.6.2. The adaptive lifting transform can then be inverted, which yields an estimate $\hat{g}$ of the true hydropathy profile at the residue locations $\{x_i\}_i$.

In the previous chapter, we have seen that in the task of denoising smooth signals, or signals with a small number of discontinuities, AdaptPred with neighbourhoods of size 2 performs best, while for denoising non-smooth signals, AdaptNeigh using up to 2 neighbours at each stage, gave the best results. Hence for denoising our version of the hydropathy profile (obtained as explained in section 5.4.2), we shall use these methods combined with either the posterior mean or median of the corresponding posterior distributions.

5.6 Prediction of transmembrane segments

For obtaining an estimate of the true hydropathy level g_i corresponding to each residue, i, in the sequence of a protein of interest, we propose the following procedure

1. Estimate the distance between any two consecutive residues in the protein sequence, and use the Kyte and Doolittle hydrophobicity scale to generate the hydropathy profile associated to the protein of interest (see section 5.4).

2. Use adaptively constructed wavelets and statistical shrinkage to estimate the 'true' hydropathy signal. We propose 4 alternatives

    - AdaptPred with 2 neighbours, combined with the modified empirical Bayes shrinkage using posterior mean and median of the corresponding posterior distributions (see sections 3.4 and 3.6.2).

- AdaptNeigh using up to 2 neighbours, combined with the modified empirical Bayes shrinkage procedure, using both posterior mean and median of the corresponding posterior distributions (see sections 3.4 and 3.6.2).

3. Classify all residues corresponding to smoothed hydrophobicities larger than the estimated (smoothed) average as transmembranar, provided that they are part of hydrophobic stretches longer than 11 residues, as recommended in Rost *et al.* (1995).

We will refer to the proposed methods as *AP2 mean*, *AP2 median*, *AN1 mean* and *AN1 median*, respectively.

In a study that we will present in what follows, we will compare the performances of our proposed methods against the methodology involving classical wavelets. Since simulations in the previous chapter showed that the AdaptPred methods with neighbourhood of size 2 have similar denoising performances regardless the type of neighbourhood configuration, (see tables in section 4.3.3), we investigated AdaptPred with a prediction step that uses the 2 closest neighbours.

For clarity of exposure, we also describe the methodology involving classical wavelets, which essentially consists of the same steps as above, dealt with in a different manner

1. Use the Kyte and Doolittle hydrophobicity scale to generate the hydropathy profile associated to the protein of interest, assuming its residues are equally spaced (see section 5.4).

2. Use classical wavelets and empirical Bayesian shrinkage with posterior mean, median policies to estimate the 'true' hydropathy signal (see section 3.6.2).

3. Classify all residues corresponding to smoothed hydrophobicities larger than the estimated (smoothed) average as transmembranar, provided

that they are part of hydrophobic stretches longer than 11 residues.

When using classical wavelet methods, a choice of wavelet and primary resolution must be made, hence we tested Daubechies wavelets from the least asymmetric family with vanishing moments from 2 to 10. We chose to work with the wavelet family generated by the least asymmetric wavelet with 8 vanishing moments, 's8', and use 4 resolution levels in the hydropathy signal decomposition. The same choice has been previously reported in the literature by Lio and Vannucci (2000).

We will refer to these methods using the above choice of wavelet and resolution level as *Daub mean*, respectively *Daub median*. We note here that in the decomposition using adaptive wavelets, we kept the same number of scaling coefficients as in the decomposition using Daubechies 's8'.

5.7 Case study

We tested our proposed methods using 46 of the 48 proteins from Rost *et al.* (1995) (the double–blind set, available from

`http://www.embl-heidelberg.de/~rost/Papers/1996_phdtop/Blind.html`).

The search on the AD1 antigen retrieves the entry 'cd63-rat', which subsequently appears in the database, and the glutamate receptor A precursor contains a much longer sequence than the rest, causing computer memory difficulties with our algorithms (on our machine with 2Gb memory size, S-Plus cannot allocate a vector of size 4.1Gb). All proteins in this study are integral membrane proteins of helical nature, for which reliable experimental data assessing locations of transmembrane helices is available.

In the above dataset, 15 proteins belong to the tetraspanin family (TM4SF), 22 belong to the ligand-gated ionic channel family (TC 1.A.9), and the rest belong to different families. The last group consists of only 9 proteins, hence we added another 10 proteins randomly selected from the set of 83 cross-validation proteins used in the same study by Rost *et al.* (1995). This way the size of

this group was boosted to 19 proteins.

The natural division of the dataset into families has led us to construct average distance matrices corresponding to each of the two main families. In the calculation of each matrix, the chains aligned to the sequences belonging to each family have been used. Hence for each of the proteins in one of these families, the missing distances will be imputed from its corresponding overall distance matrix.

Along with these two matrices, we have also computed another two family-specific distance matrices, based this time on the structure of the entire proteins (rather than only on the chains) that were aligned to sequences belonging to each family. If the family-specific matrix computed based on the aligned chains contains no information on a particular residue combination, the missing value is taken from the family-specific matrix which uses the entire protein structure.

When a protein that does not belong to one of these two families is being analysed, we use the distances supplied by an overall 'average' distance matrix (see figure 5.1), computed from a database comprising 402 proteins with determined 3D structure— the ones aligned to all the proteins investigated.

On this protein dataset, we shall show that transmembrane segment prediction improves by incorporating the inter-residue distances. As mentioned above, a set of 10 proteins has been added to the 46 proteins dataset, in order to boost the number of proteins that do not belong to either of the two families. As a consequence, we will report the overall results obtained on all 56 proteins.

We compared our proposed methods versus the classical ones (all described in 5.6). Since the results obtained by using *AP2 median* and *Daub median* are systematically underperforming the other methods, we shall not report them, and concentrate instead on *AP2 mean*, *AN1 mean*, *AN1 median* and *Daub mean*.

In order to assess the behaviour of any method, we will first introduce the

measures for performance we used.

5.7.1 Measures of prediction accuracy

All methods produce their corresponding predicted transmembranar helices which we have to compare against the experimental data available, and assess which is the better prediction.

In the context of segment prediction, we believe that there is no obvious measure that would give a concise answer as to which of the predictions is better, hence we used several measures for the accuracy of prediction.

Measures referring to the residue accuracy. (see for example Rost and Sander (1993))

- the percentage of residues predicted correctly in either of the two states (transmembranar or not), denoted Q_2,
- the percentage of residues which are correctly predicted as transmembranar, relative to the number of transmembranar residues observed, denoted Q_{obs},
- the percentage of residues which are correctly predicted as transmembranar, relative to the predicted number of transmembranar residues, denoted Q_{pred}.

Measures referring to the segment accuracy. (see for example Rost et al. (1996))

- the number of observed (true) transmembrane segments, denoted N_{obs},
- the number of predicted transmembrane segments, denoted N_{pred},
- the number of correctly predicted transmembrane segments, denoted N_{corr}, where a segment is considered to be correctly predicted if there is an overlap of at least 5 residues with a true one,

- sensitivity, i.e. the percentage of observed transmembrane segments that were correctly predicted, $Sens{=}N_{corr}/N_{obs}$,
- specificity, i.e. the percentage of predicted transmembrane segments that are correct, $Spec{=}N_{corr}/N_{pred}$,
- average observed segment length, $< L >_{obs}$,
- average predicted segment length, $< L >_{pred}$,
- segment overlap Sov_{obs}, Sov_{pred}, which are more sophisticated measures for evaluating (on a scale from 0% to 100%) respectively the correctness of segment prediction versus the true segments and the fraction of the predicted segments that is correct; these measures take into account the *degree* of overlap between predicted and observed segments, rather than considering a segment to be correctly predicted if there is an overlap of at least 5 residues (for more details see Zemla *et al.* (1999)).

5.7.2 Discussion of results

Using the above measures, we evaluated the performances of our methods versus the performance of the method employing Daubechies 's8' on equally spaced grids.

Out of our methods, we concluded that *AN1 mean* gives the best results throughout the study, hence this is the method we recommend, followed by *AN1 median*. Occasionally (even though very rare), it happens for these methods to produce predicted segments that are too short (average length under 14 residues, see Lio and Vannucci (2000)) or too long (average length over 34 residues, see Rost *et al.* (1995)), situation when *AP2 mean* should be chosen.

Results on the TM4SF family.

We found out that on the proteins belonging to the tetraspanin TM4SF family, the classical method mostly gives very good prediction, with only a few exceptions. On the leukocyte antigen CD37 (UniProt entry 'cd37-human')

Daubechies 's8' fails almost completely to recognize the true segments, giving Sov_{obs} and Sov_{pred} values of 0.5 and 0.34 respectively.

The results show that all of our methods have similar performances. Overall, our segment prediction accuracy is very similar to the one obtained through the classical method, as shown by the results in table 5.1.

We obtain higher sensitivity (i.e. the percentage of observed transmembranar segments that were correctly predicted), and very similar specificity, as well as very similar Sov values. These values indicate an accurate segment prediction, judged not only by the simple criterion of considering a segment correctly predicted if there is an overlap of at least 5 residues with a true one, but also by the better measure provided by Sov, which takes into account the change points as well.

The per-residue measures indicate a better behaviour for the classical method, but we should keep in mind that these measures should be considered with care, since we are primarily interested in sequences of residues and their positions within the chain.

At a close examination of the results based on which we obtained table 5.1, we notice that our method provides more homogenous estimation, and there is no failure of prediction for any of the proteins, unlike the classical method.

Example. We now illustrate the above discussion with an example: we will investigate the tumor associated antigen L6 (UniProt entry 't4s1-human') protein. It has a 202 residues long sequence, to which 4 proteins with known 3D structure have been aligned and used for estimating the inter-residue distances. The values for the missing pairs have been taken from the overall distance matrix associated with this family.

The transmembrane segments, determined experimentally, are thought to be: 10-30, 50-70, 94-114, 162-182.

We chose this example because it is one of the very few on which our methods *AN1 mean, median* prove to split the segments too much (producing $< L >_{pred} = 12.25$), and hence the second best method, *AP2 mean* will be used.

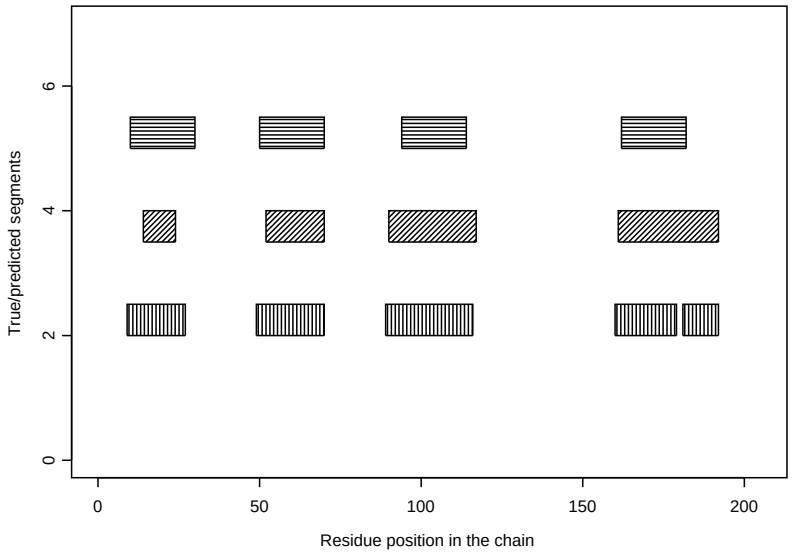

Figure 5.3: True and predicted segments for 't4s1-human': horizontally filled rectangles=True, diagonally filled rectangles=*AP2 mean*, vertically filled rectangles=*Daub mean*.

Our method predicts the following transmembrane helices 14-24, 52-70, 90-117, 161-192, while the classical wavelet method gives 9-27, 49-70, 89-116, 160-179, 181-192. It is easier if we simultaneously visualise the segments, as in figure 5.3.

The hydropathy profile corresponding to this protein appears in figure 5.4.

Figure 5.5 shows the centred denoised hydrophobicity profiles corresponding to both our method (top) and the classical one (bottom).

The accuracy measures for both methods are

1. Our method:

   $Q_2 = 0.85$, $Q_{obs} = 0.86$, $Q_{pred} = 0.80$, $< L >_{obs} = 21$, $< L >_{pred} = 22.5$, $N_{obs} = 4$, $N_{pred} = 4$, $N_{corr} = 4$, $Sens = 1$, $Spec = 1$, $Sov_{obs} = 0.93$, $Sov_{pred} = 0.96$.

2. Classical method:

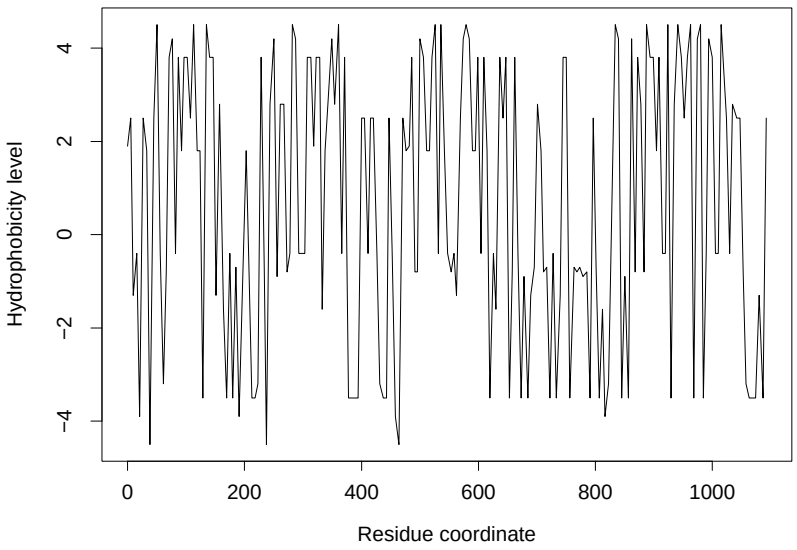

Figure 5.4: Hydropathy profile of 't4s1-human'.

$Q_2 = 0.88$, $Q_{obs} = 0.95$, $Q_{pred} = 0.79$, $< L >_{obs} = 21$, $< L >_{pred} = 20.2$, $N_{obs} = 4$, $N_{pred} = 5$, $N_{corr} = 4$, $Sens = 1$, $Spec = 0.8$, $Sov_{obs} = 0.83$, $Sov_{pred} = 0.90$.

In this example we see that *Daub mean* predicts an extra segment, of length 12, while *AP2 mean* does not identify correctly the upper bound of the last transmembrane helix, overextending it.

Results on the TC1A9 family.

On the ligand-gated ionic channel (TC 1.A.9) family, classical based wavelet methods give quite poor predictions most of the time, with (Sov_{obs},Sov_{pred}) values ranging from (0.51,0.3) to at most (1,0.51). For 4 proteins, values around (0.5,0.3) are obtained, hence the classical method fails to make a good prediction for them. Most of the *Sov* values are concentrated around (0.8,0.45), indicating that there is a tendency of overpredicting segments (predicting segments that are not truly transmembranar), and also of not being able to correctly detect the boundaries of the true segments. This generally translates in predicting a segment as being the merging of 2 or, in a few cases, even 3 true

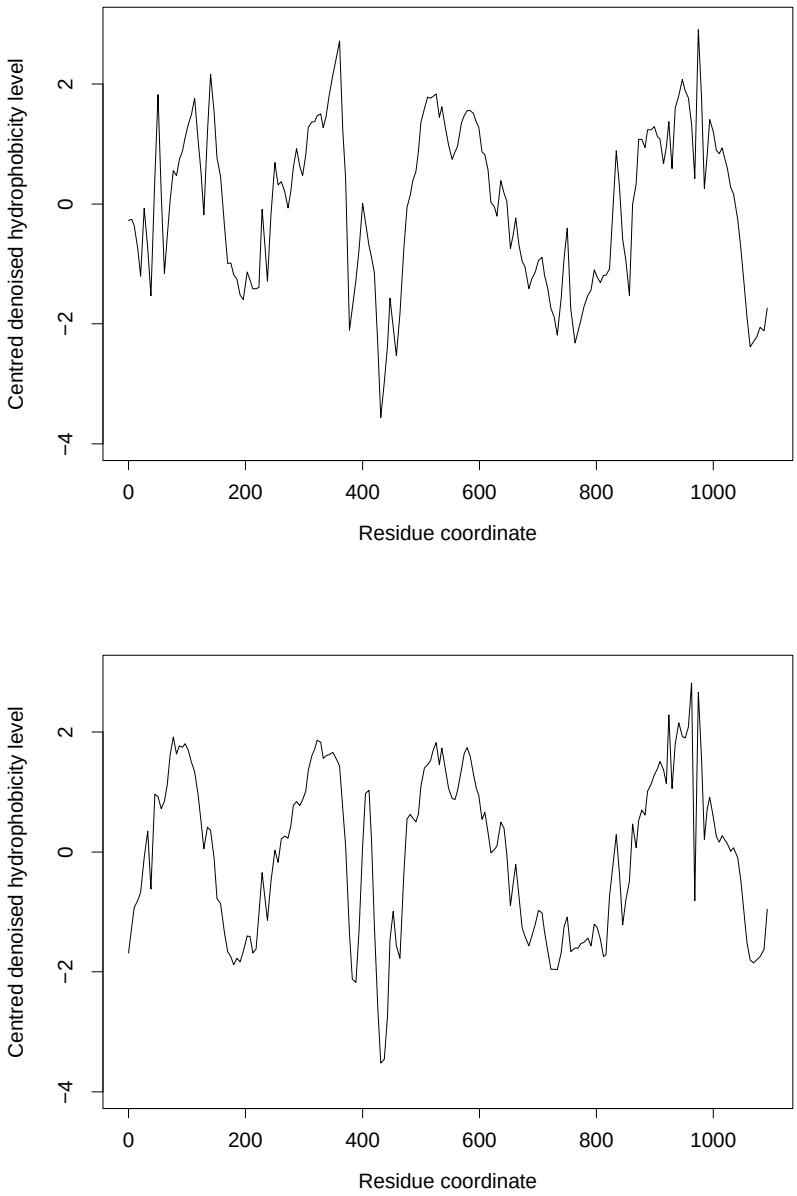

Figure 5.5: Centred denoised hydrophobicity profiles of 't4s1-human': top, using *AP2 mean*; bottom, using *Daub mean*.

segments.

Our methods give an improved prediction for most of the proteins. This time *AN1 mean* is superior to *AN1 median*, which is similar to *AP2 mean*; all of them provide better performances than *Daub mean*. Most of the *Sov* values for *AN1 mean* are within the range of (0.8-1,0.5-0.8), considerably higher than the results obtained using the classical wavelets.

By examining table 5.2, we notice that while improving the sensitivity (the boundaries of the true segments are correctly identified, and segments are seldom merged), we do not seem to be able to significantly improve upon the specificity of our prediction (some segments are falsely predicted as transmembrane).

Also for this family, the prediction performance given by our method is more homogenous than in the classical case.

Example. We now examine a protein belonging to this family: we chose the gamma-aminobutyric-acid receptor gamma-3 subunit precursor (UniProt entry 'gac3-mouse'), which displays the typical behaviour of both methods. It has a sequence of length 467 residues, to which chains coming from 8 proteins with determined 3D structure have been aligned. The inter-residue distances were computed based on these proteins, and the values of the missing pairs were imputed from the overall distance matrix corresponding to this family.

The experimentally determined transmembrane segments are believed to be 255-277, 281-303, 315-337, 444-467.

Our method predicts the following segments 5-15, 77-88, 116-133, 232-249, 253-277, 288-303, 317-332, 447-467, while by the usage of Daubechies wavelets, we obtain 1-15, 72-92, 118-134, 157-173, 230-296, 306-337, 443-467.

Figure 5.6 helps us 'see' the predicted segments versus the true ones.

The hydropathy profile of 'gac3-mouse' appears in figure 5.7. Note how due to its longer sequence, it is even more difficult to visually assess which could possibly be the hydrophobic segments.

Figure 5.8 shows the corresponding coarse versions of the hydropathy profile

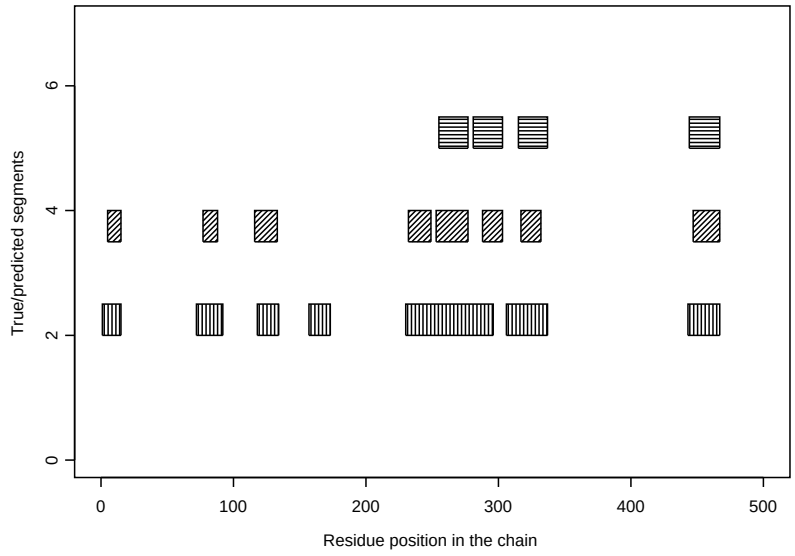

Figure 5.6: True and predicted segments for 'gac3-mouse': horizontally filled rectangles=True, diagonally filled rectangles=*AN1 mean*, vertically filled rectangles=*Daub mean*.

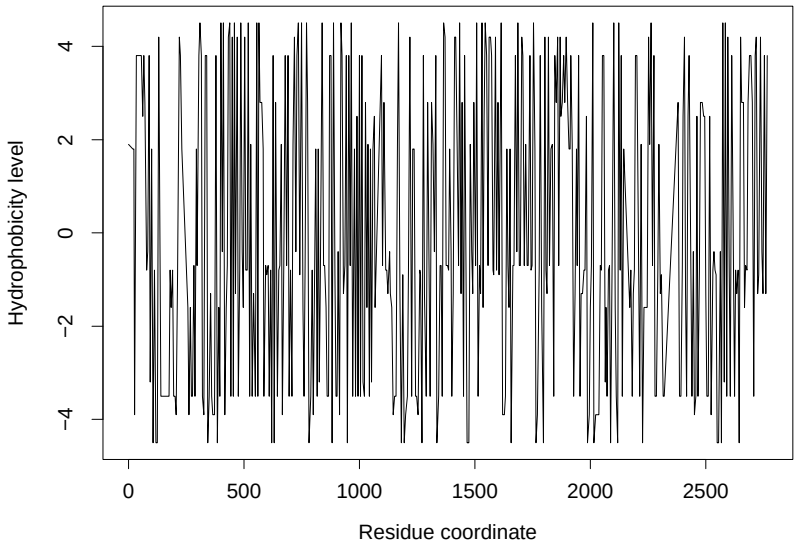

Figure 5.7: Hydropathy profile of 'gac3-mouse'.

of 'gac3-mouse'.

When measuring the performance of the predicted transmembrane segments for 'gac3-mouse', we obtain the following results

1. Our method:

   $Q_2 = 0.83$, $Q_{obs} = 0.82$, $Q_{pred} = 0.55$, $< L >_{obs} = 23.25$, $< L >_{pred} = 17.12$, $N_{obs} = 4$, $N_{pred} = 8$, $N_{corr} = 4$, $Sens = 1$, $Spec = 0.5$, $Sov_{obs} = 1$, $Sov_{pred} = 0.57$.

2. Classical method:

   $Q_2 = 0.75$, $Q_{obs} = 0.92$, $Q_{pred} = 0.44$, $< L >_{obs} = 23.25$, $< L >_{pred} = 27.71$, $N_{obs} = 4$, $N_{pred} = 7$, $N_{corr} = 3$, $Sens = 0.75$, $Spec = 0.43$, $Sov_{obs} = 0.72$, $Sov_{pred} = 0.44$.

We notice in this example the behaviour described earlier, in that both methods overpredict the transmembrane segments, and the classical wavelets are also merging some of the true segments.

Remark. All of the methods mentioned in the beginning of this paper (see for example Rost *et al.* (1995, 1996) or Fisher *et al.* (2003)), besides the mathematical filtering, employ a bio-chemical filtering as well, which we keep to a minimum (we only cut predicted helices containing at most 10 residues). Such further filtering and inspection of the already predicted segments will considerably improve the prediction specificity and sensitivity (and will also improve Sov_{pred}, Sov_{obs}), by eliminating some of the unlikely segments, or splitting the segments considered to be too large into two or more segments.

In our study, a closer examination of the obtained predicted segments in the ligand-gated ionic channel (TC 1.A.9) family shows that a lot of the segments wrongly predicted as transmembranar are very short (11-15 residues), and hence unlikely to 'survive' a bio-chemical filtering procedure. It may also be that some of them are too long, and splitting them into more segments might be a solution.

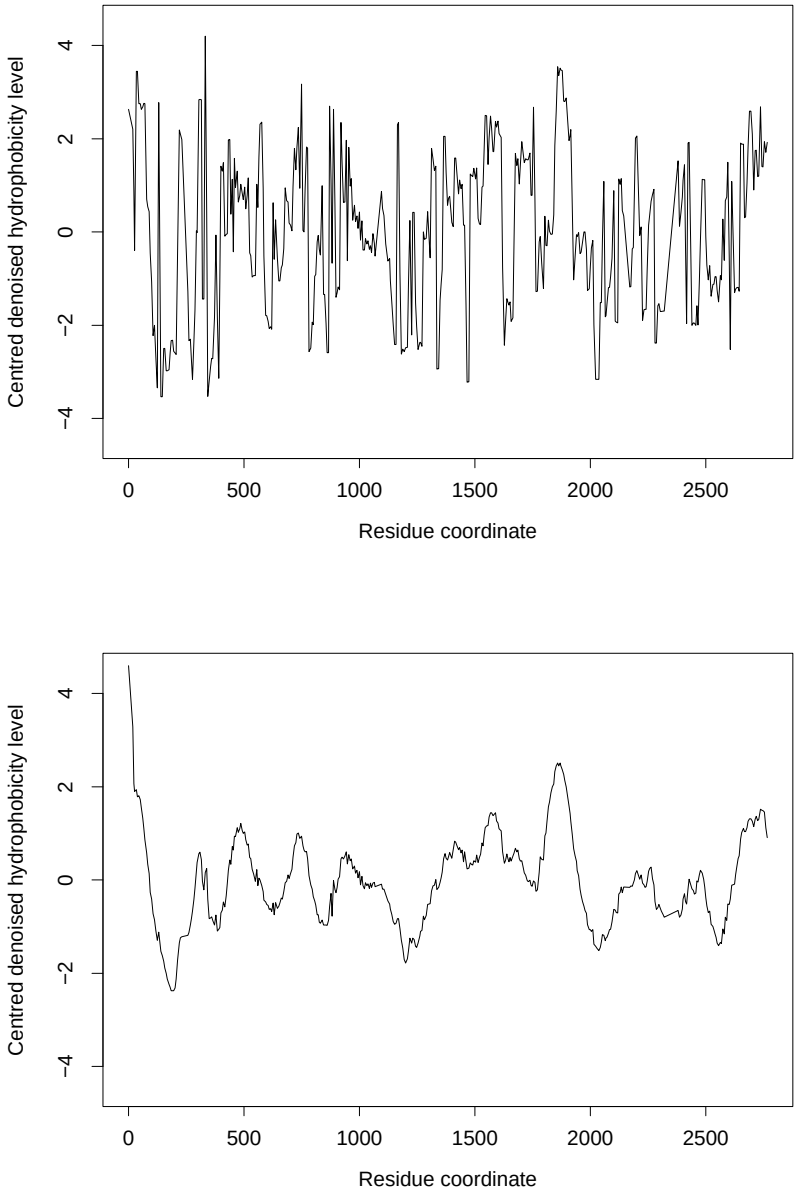

Figure 5.8: Centred denoised hydrophobicity profiles of 'gac3-mouse': top, using *AN1 mean*; bottom, using *Daub mean*.

In our approach, we kept exclusively a mathematical filtering procedure and investigated its behaviour with and without the information given by multiple aligned sequences with known 3D structure. Having improved upon the basic mathematical prediction, various other procedures (such as the bio-chemical filtering discussed above) could then be added, and contribute to an improved final prediction.

Results for the rest of the (19) proteins. Finally, for the rest of proteins, the ones belonging to different families, the predictions of our methods and of the one employing classical wavelets are quite good, with the exception of 3 proteins which have only one (true) transmembranar segment. For these proteins, our methods and the classical one have very similar performances, in that the *Sov* values are around (0.9-1,0.2-0.6), indicating that the methods correctly identify the true segment, but additionally predict false ones. For the remaining 16 proteins, none of the methods fails and the range of (Sov_{obs}, Sov_{pred}) values is (0.64-1,0.67-1) for *AN1 mean*, and (0.67-1,0.51-1) for *Daub mean*.

Analysing the results obtained on the whole set of 19 proteins, we see that our method either outperforms the results obtained through the Daubechies wavelets or gives similar results, and in only 3 cases we obtain worse results than by using the Daubechies 's8'. For this group of proteins, the best results are obtained by predicting through *AN1 mean* too.

Examining table 5.3 we notice that we obtain improved specificity values for *AN1 mean* as compared to the results obtained through the usage of classical wavelets, and a similar sensitivity value. This is reflected also by examining the *Sov* values.

Example. We will now examine the results for the undecaprenyl-phosphate galactosephosphotransferase (UniProt entry 'rfbp-salty'), which has no resolved 3D structure, and belongs to the bacterial sugar transferase family.

Starting from its primary structure of 476 residues, one protein with known 3D structure has been aligned to it and used in the computation of inter-residue

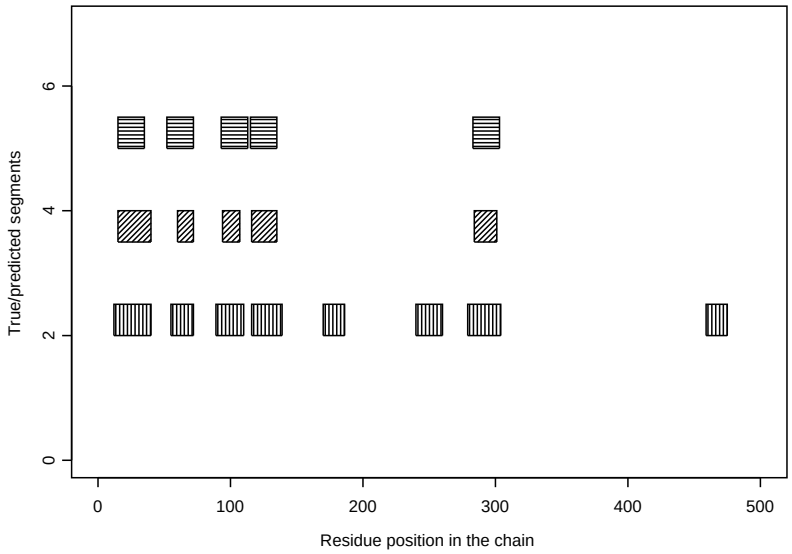

Figure 5.9: True and predicted segments for 'rfbp-salty': horizontally filled rectangles=True, diagonally filled rectangles=*AN1 mean*, vertically filled rectangles=*Daub mean*.

distances. The values for the missing pairs have been taken from the overall 'average' distance matrix.

Experimentally determined data is available for the transmembrane segments, which are thought to be 15-35, 52-72, 93-113, 115-135, 283-303.

After employing our method (*AN1 mean*), we obtain the following predicted segments 15-40, 60-72, 94-107, 116-135, 284-301, while through the usage of *Daub mean* we obtain 12-40, 55-72, 89-110, 116-139, 170-186, 240-260, 279-304, 459-475, corresponding to figure 5.9.

The estimated versions of the true hydropathy signal obtained through *AN1 mean*, and through *Daub mean* respectively, appear in figure 5.10.

The prediction performance is characterised by the following values of the indices previously introduced

1. Our method:

   $Q_2 = 0.95$, $Q_{obs} = 0.82$, $Q_{pred} = 0.95$, $< L >_{obs} = 21$, $< L >_{pred} = 18.2$, $N_{obs} = 5$, $N_{pred} = 5$, $N_{corr} = 5$, $Sens = 1$, $Spec = 1$, $Sov_{obs} = 0.98$,

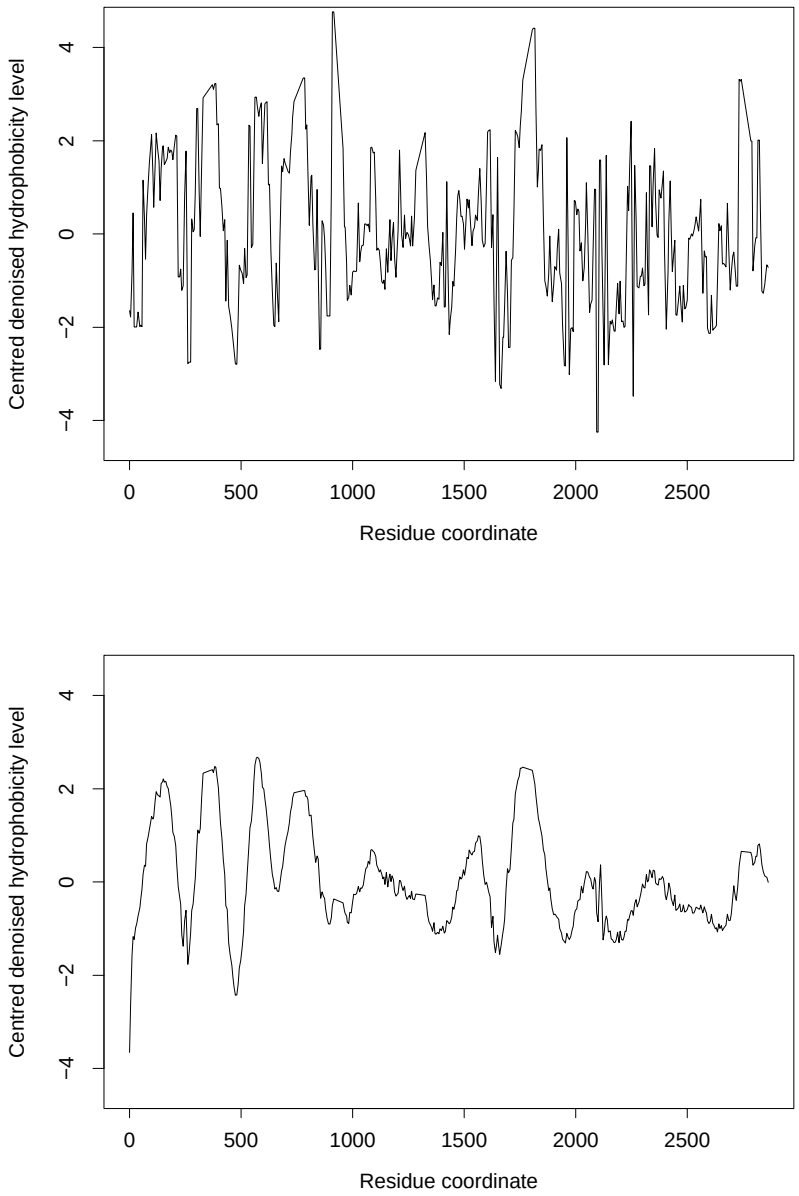

Figure 5.10: Centred denoised hydrophobicity profiles of 'rfbp-salty': top, using *AN1 mean*; bottom, using *Daub mean*.

$Sov_{pred} = 0.99$.

2. Classical wavelets:

   $Q_2 = 0.83$, $Q_{obs} = 0.93$, $Q_{pred} = 0.56$, $< L >_{obs} = 21$, $< L >_{pred} = 21.75$, $N_{obs} = 5$, $N_{pred} = 8$, $N_{corr} = 5$, $Sens = 1$, $Spec = 0.63$, $Sov_{obs} = 1$, $Sov_{pred} = 0.68$.

We see that our method does not predict extra segments, and it correctly identifies the true segments. The classical method falsely predicts as transmembranar three segments of lengths 17,21,17, which would be more difficult to eliminate through a further filtering step.

Overall conclusions. To conclude, examine table 5.4, which combines all previous data to show the overall tendency.

We compared the *Sov* values (since these are the most complete measures for the segment prediction accuracy) obtained through our methods versus the ones obtained by using classical wavelets. For performing the comparisons we used paired t-tests, since the sample size is large enough so that the tests should be robust against non-normality. For each of our methods, and both for Sov_{obs} and Sov_{pred}, we tested the null hypothesis of no difference between the mean *Sov* value of our method and the mean *Sov* value of the classical method, versus the alternative that our method provides a higher *Sov* value than the one obtained through the classical wavelets. We indicated the highly significant differences in table 5.4.

Based on a careful examination of the data and on the results of the significance tests, we conclude that we improve the quality of prediction by using resolved 3D structure of proteins that are similar to the proteins to be analysed— both in terms of the correctness of the segments with respect to the true segments and the proportion of predicted segments that are correct.

Wavelet methods using a second filtering step based on the chemical properties of the residues, report final sensitivity and specificity values of 0.93 and over. With no such further filtering, we obtain a value of 0.90, indicating that

if we additionally use such a procedure, we should obtain even higher sensitivity values. We refer to the sensitivity and specificity values since they are the measures usually reported in the literature, but we stress again that a much better measure, indicating more accurately the behaviour of the method, is *Sov*. Its observed value also confirms an improvement of the prediction accuracy. Due to the ligand-gated ionic channel family (TC 1.A.9), the specificity value in our study drops to 0.70, a higher value than the one corresponding to the classical method — 0.62, but yet a smaller value than the ones reported by the previous studies (Lio and Vannucci (2000), Fisher *et al.* (2003)), in which further to the mathematical filtering, a step of biochemical filtering is employed. The value of Sov_{pred} (0.76) is higher than the one provided by the specificity index, and it also points towards the existence of an improvement with respect to the classical method (which has Sov_{pred} of 0.67).

Remark. For the initial dataset consisting of 46 proteins, we have also tested our methodology using two different types of matrices for imputing the missing values when computing the coordinate of each residue. We remind the reader that so far we primarily used two matrices, one for each family. In the calculation of these matrices we used the chains with determined 3D structure that were aligned to the sequences belonging to each family, respectively. Solely for estimating the missing values in these matrices, we used another two matrices computed based on the structure of the entire proteins aligned to sequences belonging to each family. Now, we have also tested our methods using imputed values straight from the distances provided by these matrices. And finally, regardless the family to which the protein belongs, we have used the overall matrix computed based on 376 proteins, all the proteins aligned to the 46 proteins being analysed. Our intuition was that prediction should slightly decrease in accuracy by using less specific information. The tests proved that the specificity has slightly decreased, from the overall 0.76 to 0.74 (this difference being mainly due to the specificity decrease in the ligand-gated ionic channel family, from 0.59 to 0.55), while the sensitivity was not influenced.

5.8 Conclusions and further work

In this chapter we developed a new technique for transmembrane protein segment prediction. This improves on earlier wavelet methods by utilising resolved 3D structure information from similar proteins to provide irregularly spaced residues. The irregular spacing is generated by order-20 distance matrices which calculate inter-residue distances over families of similar proteins (and also a generic 'all-protein' matrix for use when a family matrix cannot produce a distance for a particular combination). This construction aims at obtaining a better estimate of the true function that models the level of hydrophobicity along the protein. We tested our method on helical transmembrane proteins, and consequently we generated distance matrices that reflect the helicity property. An interesting direction would be to further extend the study to beta-barrel transmembrane proteins. For the future, the 'paradigm' provides a way of generalising multiscale algorithms for irregularly spaced objects (such as proteins) and hence lifting shows great promise for directly utilising 3D resolved information in a mathematical multiscale manner which is informed by the biochemical reality.

	N_{pred}	N_{corr}	*Sens*	*Spec*	$< L >_{obs}$	$< L >_{pred}$	Sov_{obs}	Sov_{pred}	Q_2	Q_{obs}	Q_{pred}
AP2 mean	62	60	1.00	0.97	22.08	25.45	0.96	0.95	0.88	0.94	0.78
AN1 median	63	60	1.00	0.95	22.08	22.05	0.95	0.94	0.88	0.87	0.81
AN1 mean	62	60	1.00	0.97	22.08	20.36	0.93	0.93	0.88	0.82	0.84
Daub mean	58	57	0.95	0.98	22.08	28.90	0.93	0.92	0.90	0.96	0.80

Table 5.1: Results obtained on the TM4SF family (15 proteins, 60 experimentally determined transmembrane segments). N_{pred}, N_{corr} give the number of predicted, respectively correctly predicted transmembrane segments; *Sens*, *Spec* give the sensitivity, specificity of prediction; $< L >_{obs}$, $< L >_{pred}$ are the average length of the observed, predicted segments; Sov_{obs}, Sov_{pred} evaluate the correctness of prediction versus the true segments, and the fraction of the predicted segments that is correct; Q_2 is the percentage of correctly predicted residues, Q_{obs}, Q_{pred} measure the percentage of correctly predicted residues relative to the number of observed, respectively predicted transmembranar residues.

	N_{pred}	N_{corr}	*Sens*	*Spec*	$< L >_{obs}$	$< L >_{pred}$	Sov_{obs}	Sov_{pred}	Q_2	Q_{obs}	Q_{pred}
AP2 mean	168	72	0.82	0.43	22.34	25.21	0.85	0.49	0.77	0.95	0.45
AN1 median	173	73	0.83	0.42	22.34	21.71	0.84	0.50	0.76	0.84	0.44
AN1 mean	148	73	0.83	0.49	22.34	19.77	0.84	0.59	0.82	0.79	0.52
Daub mean	165	63	0.72	0.38	22.34	27.07	0.75	0.44	0.75	0.96	0.43

Table 5.2: Results obtained on the TC 1.A.9 family (22 proteins, 88 experimentally determined transmembrane segments). N_{pred}, N_{corr} give the number of predicted, respectively correctly predicted transmembrane segments; *Sens*, *Spec* give the sensitivity, specificity of prediction; $< L >_{obs}$, $< L >_{pred}$ are the average length of the observed, predicted segments; Sov_{obs}, Sov_{pred} evaluate the correctness of prediction versus the true segments, and the fraction of the predicted segments that is correct; Q_2 is the percentage of correctly predicted residues, Q_{obs}, Q_{pred} measure the percentage of correctly predicted residues relative to the number of observed, respectively predicted transmembranar residues.

	N_{pred}	N_{corr}	*Sens*	*Spec*	$< L >_{obs}$	$< L >_{pred}$	Sov_{obs}	Sov_{pred}	Q_2	Q_{obs}	Q_{pred}
AP2 mean	98	82	0.90	0.84	23.89	23.59	0.90	0.77	0.82	0.79	0.75
AN1 median	106	83	0.91	0.78	23.89	19.13	0.91	0.78	0.80	0.71	0.76
AN1 mean	95	82	0.90	0.86	23.89	17.99	0.89	0.82	0.83	0.66	0.84
Daub mean	98	80	0.88	0.82	23.89	24.02	0.92	0.75	0.81	0.80	0.73

Table 5.3: Results obtained on the rest of the proteins (19 proteins, 91 experimentally determined transmembrane segments). N_{pred}, N_{corr} give the number of predicted, respectively correctly predicted transmembrane segments; *Sens*, *Spec* give the sensitivity, specificity of prediction; $< L >_{obs}$, $< L >_{pred}$ are the average length of the observed, predicted segments; Sov_{obs}, Sov_{pred} evaluate the correctness of prediction versus the true segments, and the fraction of the predicted segments that is correct; Q_2 is the percentage of correctly predicted residues, Q_{obs}, Q_{pred} measure the percentage of correctly predicted residues relative to the number of observed, respectively predicted transmembranar residues.

	N_{pred}	N_{corr}	*Sens*	*Spec*	$< L >_{obs}$	$< L >_{pred}$	Sov_{obs}	Sov_{pred}	Q_2	Q_{obs}	Q_{pred}
AP2 mean	328	214	0.90	0.65	22.80	24.73	0.90^{β}	0.71^{β}	0.80	0.88	0.60
AN1 median	342	216	0.90	0.63	22.80	20.93	0.89^{γ}	0.71^{β}	0.80	0.79	0.60
AN1 mean	305	215	0.90	0.70	22.80	19.32	0.88	0.76^{α}	0.83	0.75	0.68
Daub mean	321	200	0.84	0.62	22.80	26.52	0.86	0.67	0.80	0.89	0.59

Table 5.4: Overall results (56 proteins, 239 experimentally determined transmembrane segments). N_{pred}, N_{corr} give the number of predicted, respectively correctly predicted transmembrane segments; *Sens*, *Spec* give the sensitivity, specificity of prediction; $< L >_{obs}$, $< L >_{pred}$ are the average length of the observed, predicted segments; Sov_{obs}, Sov_{pred} evaluate the correctness of prediction versus the true segments, and the fraction of the predicted segments that is correct; Q_2 is the percentage of correctly predicted residues, Q_{obs}, Q_{pred} measure the percentage of correctly predicted residues relative to the number of observed, respectively predicted transmembranar residues; α indicates a significantly higher *Sov* value for the corresponding method than for *Daub mean* at 99% confidence level, while β corresponds to a significantly higher result for our (corresponding) method at 95% confidence level and γ indicates a significantly higher result for our (corresponding) method at 90% confidence level.

Chapter 6

Spectral estimation for locally stationary time series with missing observations

6.1 Motivation

Time series with missing data frequently appear in practice. The 'patterns' of the missing observations are various: for instance a whole sequence of data might be missing due to a malfunction of the machine recording the observations, the data may be censored, observations may be missing at random or following a systematic pattern.

In this context, data analysis cannot take place within the well-specified framework devoted to discrete time series measured at equal time intervals.

Quite commonly, when missing observations are present in the data, they are imputed following various recommendations— 'common sense' is one of them, or some computations may be performed on the 'gappy' data (see Chatfield (2004)). If the missing observations occur with a periodic structure, Jones (1962) provides a development for spectral estimation of a stationary time series (i.e. time series for which the first and second order structures do

not exhibit time dependency), while Clinger and Van Ness (1976) discuss the problem of sampling stationary time series at fixed, unequally spaced time points such that spectrum estimation is still possible.

The existence of missing observations induces irregularities in the time locations, while certain types of data have by nature irregularly spaced observations: finance data sampled at high frequencies (such as records of traded stocks at a certain stock exchange) are such an example (Engle (2000)). Methods for autocovariance and spectral estimation for stochastic processes sampled at irregular locations have been developed (Hall *et al.* (1994); Bos *et al.* (2002)). These constructions are valid for stationary time series. However, in various fields, such as finance (Mikosch and Starica (2004)) and medicine (Nason *et al.* (2001)), modelling the observed data as stationary is not appropriate.

In this chapter we will investigate the problem of spectral estimation for a non-stationary process with missing observations. In our approach, non-stationarity is to be understood in the sense introduced by Nason *et al.* (2000). Our construction will make use of second generation wavelets, built following the 'one coefficient at a time' paradigm introduced in chapter 3.

This chapter is organized as follows: we will first briefly introduce (stationary) time series, and then we shall move towards ways of modelling time series data without imposing the strong assumption of stationarity. In this context, we will mention the concept of rescaled time introduced by Dahlhaus (1997), and then present the main results (needed for our development) in the construction of LSW processes from Nason *et al.* (2000). In order to estimate the wavelet spectrum of a LSW process with missing observations, we first construct a 'nondecimated' lifting scheme, and based on it we propose a periodogram and investigate some of its properties.

6.2 Brief introduction to time series

So far, we have been concerned with the estimation of a deterministic function, g, at some locations $\underline{x}$, from a sequence of noise contaminated observations of g at $\underline{x}$ ($\underline{f}$). We modelled the noise, and hence the observations, as independent random variables, and this assumption assisted us in estimating g.

When the observed data is modelled as a time series, a completely different framework is used. Each observation is again modelled as a random variable, but unlike before it is not assumed to be the noisy observation of a true value, but to be merely one value out of a possible range. In other words, at each location (usually denoting time), the observed value is a random variable coming from an (unknown) probability distribution. Successive observations are not assumed independent anymore, and the function that describes the values of the observations as a function of their location, and as such describes the whole process, is also assumed to be random. This is referred to in the literature under the name of a stochastic process, or simply a random function or process. Here we refer to it as a time series to emphasize that the observations are a function of time. If the experiment that produced the initial data could be repeated under identical conditions, for each repeat we would obtain a completely different 'realization' of the process (observations versus time). The data we obtain in practice is just a 'portion' of a single realization, based on which we want to describe the process behaviour over all time points.This is equivalent to considering an infinite dimensional probability distribution. Fortunately, it has been proved (see Priestley (1981) for references) that it suffices to analyse the joint distribution of the values of a process at an arbitrarily large, but finite number of time points. Although this substantially reduces the problem, imposing some assumptions on the process behaviour is still necessary.

In what follows we will first formally introduce time series.

Definition 6.2.1. *A time series is a stochastic process indexed over time, i.e.*

an application $X(\cdot\,,\cdot\,) : D \times \Omega \to \mathbb{R}$*, where* D *is the time space and* Ω *is the space of events, such that:*

1. *for each fixed time point* $t \in D$*,* $X(t,\cdot\,) : \Omega \to \mathbb{R}$ *is a random variable,*

2. *for each fixed event* ω*,* $X(\cdot\,,\omega) : D \to \mathbb{R}$ *is a function of time, called the trajectory of the process associated to* ω*.*

We used the notation ω in the above definition since this is the usual notation for events; however, it should not be confused to the notation that appears everywhere else in this thesis, in which ω is used for the frequency of a sinusoidal type function.

If $D \subseteq \mathbb{R}$, then the process is denoted by $X(t)$ and is said to be a *continuous time process.*

If D consists of a discrete set of values then we have a *discrete time process*, usually denoted by $(X_t)_{t\in D}$. In this case D is often taken to be $\mathbb{Z}$ or $\{0, 1, ..., N\}$ due to mathematical necessity, rather than as a true model for the data at hand (which might feature missing observations or irregularly spaced time points).

We shall denote the observed data by $X_0, ..., X_{T-1}$. This is often referred to under the name of time series, although we should bear in mind that there is a fundamental difference between the ensemble of all possible realizations and just (a 'portion' of) one realization.

In practice, time series data routinely crop up in practice in various fields, from marketing (for example the records of monthly sales of say plane tickets on a certain route, available from say, January 2000 to December 2004) to medicine (the heart rate of an infant, sampled at every 1/16Hz during a night, Nason *et al.* (2001)).

6.2.1 Stationary time series

As we have already mentioned, in order to be able to make inferences on the characteristics of a time series (such as its variance), certain assumptions must

be imposed on its evolution. Most often, the process is assumed to be such that if we divide any of its realizations into smaller sections, then each section looks very much like any other section of that realization, i.e. the statistical properties of the time series do not change with time. Such processes are called (strictly) stationary time series, and some excellent monographs entirely devoted to studying them exist— see for instance Priestley (1981), Chatfield (2004) or Brockwell and Davies (1991).

Definition 6.2.2. *We say that $(X_t)_{t\in\mathbb{Z}}$ is a (strictly) stationary time series if the joint distribution of $(X_{t_1},X_{t_2},...,X_{t_n})$ is the same as the joint distribution of $(X_{t_1+\tau},X_{t_2+\tau},...,X_{t_n+\tau})$, $\forall n, t_1, ..., t_n, \tau \in \mathbb{Z}$.*

This definition has several important implications:

- X_t are identically distributed for each t, hence $E(X_t)$ is the same for any t, so the series has no trend, and $\text{var}(X_t)$ is constant with respect to t,
- $\text{cov}(X_t, X_{t+\tau})$ only depends on the time shift τ (often referred to as 'lag'): $\text{cov}(X_t, X_{t+\tau}) = r_X(\tau)$, where $r_X(\cdot)$ is called the *autocovariance function* associated with the process $(X_t)_t$.

Most often, the definition above is relaxed and it is assumed that the process is *wide sense stationary*: the first and second order structures of the process do not exhibit time dependency (i.e. the series has no trend, it has the same variation regardless of the time point and its autocovariance function only depends on the time shift). Wide sense stationarity does not imply that the probability distribution of X_t is the same for each t, but only that their main features are the same— the moments up to order 2. In the case of Gaussian processes, wide sense stationarity implies strict sense stationarity.

The autocovariance function associated with a process $(X_t)_{t\in\mathbb{Z}}$ is a valuable tool that helps describe the behaviour of the process over time. Hence often it is of interest to model and estimate it. The process properties can also be analysed in the frequency domain, by analogy with the Fourier analysis of

deterministic functions. Of course, certain modifications are required, since we are now dealing with a stochastic process.

In the frequency domain, the goal is to understand how the process variance is accounted for by various frequencies of sinusoidal components. This can be achieved thanks to one of the most important results in the theory of stationary processes, the *Cramér representation of a stationary process* (Priestley (1981)), which states that any stationary time series can be represented in the limit as a 'sum' of sine and cosine waves with uncorrelated random coefficients. More exactly, a zero-mean, stationary discrete time series $(X_t)_{t\in\mathbb{Z}}$ can be written as

$$X_t = \int_{-\pi}^{\pi} A(\omega) e^{it\omega} d\xi(\omega),\ t \in \mathbb{Z}, \tag{6.1}$$

where ω is the frequency of the building blocks, $A(\omega)$ is their corresponding amplitude and $\{\xi(\omega)\}_\omega$ is a random process with orthonormal increments, i.e. $E\{d\xi(\omega)\} = 0$ and $\mathrm{cov}\{d\xi(\omega), d\xi(\omega')\} = d\omega\, \delta_{\{\omega=\omega'\}}(\omega)$. Note that for each realization of the process, we make use of a different $\{\xi(\omega)\}_\omega$, but all realizations are linked through the same deterministic function $A(\cdot)$. The 'energy' distribution of the time series $(X_t)_{t\in\mathbb{Z}}$ can then be characterised by the squared amplitudes, $|A(\cdot)|^2$.

Remark. The above formula can be viewed as an 'equivalent' of the representation (2.3) for deterministic functions. We should however note that there are some fundamental differences between (2.3) and its stochastic counterpart, (6.1). In the above formula for each frequency ω, $d\xi(\omega)$ is a random variable and the integral is in fact a stochastic integral. The equality (6.1) has therefore to be understood in the mean square sense. Also, since we are analyzing discrete time series with a unit time sampling, the range of frequencies is restricted to the interval $[-\pi, \pi]$ due to a phenomenon called aliasing, by which frequencies $\{\omega \pm 2k\pi\}_k$ are indistinguishable for ω (see for details Priestley (1981)).

For a process $(X_t)_{t\in\mathbb{Z}}$ having the representation (6.1), it can be shown

(Priestley (1981)) that its autocovariance function can be written as

$$r_X(\tau) = \int_{-\pi}^{\pi} |A(\omega)|^2 e^{i\omega\tau} d\omega,\ \tau \in \mathbb{Z}. \tag{6.2}$$

The following notation is usually used

$$f_X(\omega) = |A(\omega)|^2,\ \omega \in [-\pi, \pi], \tag{6.3}$$

and $f_X(\cdot)$ is called the *spectral density function of* $(X_t)_{t\in\mathbb{Z}}$. Formula (6.2) shows that the autocovariance function and the spectral density function corresponding to a process are a Fourier pair.

Priestley nicely summarizes the interpretation of the spectral density function: $f_X(\omega)d\omega$ represents 'the average (over all realizations) of the contribution to the total power (variance of the process) from components in X_t with frequencies between ω and $\omega + d\omega$'.

6.2.2 Locally stationary time series

We have already noted that in practice it is not always reasonable to assume that certain time series have a stationary behaviour. Once the stationarity assumption is dropped, other assumptions on the process, although less restrictive, still have to be imposed in order to be able to make inferences on the process characteristics, such as estimation of its variance.

Throughout this chapter we shall concentrate on trend-free processes with a second order structure that varies slowly with time. Such time series are called locally stationary (Nason and von Sachs (1999)), since they appear to have a stationary behaviour over short stretches of time. This ensures that their statistical characteristics (such as the autocovariance function) can be (locally) estimated by pooling the observed data over the regions of local stationarity.

One of the first attempts to formulate a spectral theory for nonstationary processes is due to Priestley (1965). In his paper, Priestley (1965) introduced,

by means of time-dependent amplitudes $A_t(\cdot)$, a class of nonstationary processes: the oscillatory processes. The variation of their amplitudes as a function of time is assumed to have a degree of regularity, which ensures the local stationary character of the process. A time-dependent evolutionary spectrum is also defined, which describes the frequency content of the process over time neighbourhoods.

Motivated by the need to construct a framework that would allow for asymptotic inference (such as establishing the consistency of various estimators), Dahlhaus (1997) introduced a new concept of rescaled time by controlling the evolution of the individual amplitudes $A_t(\omega)$ through a function dependent on rescaled time, $A(t/T, \omega)$, where $t \in \overline{0, T-1}$. In his construction, with larger T more information is collected on the local behaviour of the function $A(\cdot\,, \omega)$, defined on the fixed interval $[0, 1]$. Consequently, the model structure may be asymptotically obtained by addressing the familiar problem of curve estimation.

The formal definition of the locally stationary processes introduced by Dahlhaus (1997) is given in what follows.

Definition 6.2.3. *The sequence of zero-mean stochastic processes* $\{(X_{t,T})_{t\in\overline{0,T-1}}$, $T = 1, 2, ...\}$ *is said to be locally stationary with asymptotic transfer function* $A(\cdot\,, \cdot) : [0, 1] \times \mathbb{R} \to \mathbb{C}$, *continuous in the first argument and* 2π*-periodic in the second one, if there exists a representation*

$$X_{t,T} = \int_{-\pi}^{\pi} A^0_{t,T}(\omega) e^{it\omega} d\xi(\omega), \tag{6.4}$$

where $\exists\, K$ *such that*

$$sup_{t,\omega} |A^0_{t,T}(\omega) - A(\frac{t}{T}, \omega)| \leq \frac{K}{T}, \forall T,$$

and $\{\xi(\omega)\}_\omega$ *is a random process satisfying some specific properties (see Dahlhaus (1997) for a detailed description).*

Notice that in the above model a different process $(X_{t,T})_{t\in\overline{0,T-1}}$ corresponds to each T. Therefore, an increase in T must not be understood as extending a fixed time process further into the future, nor as discretizing it on a finer grid. The common factor of all these processes is the asymptotic transfer function $A(\cdot\,,\cdot\,)$ which regulates the behaviour of the time varying individual amplitudes. The smoothness of $A(\cdot\,,\cdot\,)$ in the first argument tunes the degree of local stationarity of each process.

An associated evolutionary spectral density function is defined in terms of rescaled time ($z = t/T$), $f_X(z,\omega) = |A(z,\omega)|^2$. This has the advantage of being uniquely defined, unlike the spectrum that arises through Priestley (1965)'s approach.

6.2.3 Locally stationary wavelet (LSW) processes

Wavelets have been so far used for a wide variety of problems that arises in time series analysis, as Nason and von Sachs (1999) show in their review paper and Percival and Walden (2000) in their comprehensive monograph.

Due to their nature, wavelets deliver a time–scale representation, complementary to the time–frequency interpretation that arises from a Fourier analysis. The classical Fourier spectral analysis is replaced by a wavelet spectral analysis. For stationary processes, Chiann and Morettin (1999) used the decomposition on an orthonormal wavelet basis to develop an alternative to the classical (Fourier) periodogram, the wavelet periodogram, used to estimate the wavelet spectrum.

This chapter builds upon the work of Nason *et al.* (2000), who proposed a new way to model time series with a time-dependent second order structure, based on a family of discrete nondecimated wavelets $\{\psi_{j,k}(t)\}_{j,k}$ which replaces the set of sine and cosine waves ($\{e^{it\omega}/\,\omega \in [-\pi,\pi]\}$), and on the concept of rescaled time of Dahlhaus (1997). Their process is assumed to have a local stationary behaviour, ensured by constraining the model coefficients to change

slowly within each scale.

The authors refer to processes built as above under the name of locally stationary wavelet (LSW) processes.

In what follows, rather than making the usual assumption that the observed data lives on scale $J(T) = \log_2(T)$, the observed time series is assumed to live on scale 0, so the finest level is -1 while the coarsest level is $-J(T)$ (in a multiresolution analysis frame $... \subseteq V_{-J(T)} \subseteq ... \subseteq V_{-1} \subseteq V_0$).

Discrete wavelets.

We shall first introduce the building blocks of a LSW process, the discrete nondecimated wavelets, as introduced by Nason *et al.* (2000).

Their construction is based upon a low and a high-pass filter ($\underline{h} = \{h_k\}_k$, $\underline{g} = \{g_k\}_k$, respectively) which satisfy the quadrature mirror filter relation ($g_k = (-1)^k h_{1-k}$) and have a finite number of non-zero coefficients, which we denote by $N_h = \text{card}\{k/h_k \neq 0\}$.

Definition 6.2.4. *The family $\{\psi_{j,k}(t)\}_{j \leq -1, k \in \mathbb{Z}}$ is a nondecimated collection of discrete wavelets associated to the filters $\{\underline{h}, \underline{g}\}$ if*

1. *at each scale $j \leq -1$ we have the discrete wavelet vector $\psi_j = (\psi_{j,0}, ..., \psi_{j,L_j-1})$, where L_j is the wavelet length, given by $L_j = (2^{-j} - 1)(N_h - 1) + 1$,*
2. *we define $\psi_{j,k}(t) = \psi_{j,k-t}, k \in \mathbb{Z}$.*

The vector sequence $\{\psi_j\}_{j \leq -1}$ is recursively given by the following equations:

$$\begin{aligned} \psi_{-1,k} &= \sum_n g_{k-2n} \delta_{0,n} = g_k, \; k \in \overline{0, L_{-1} - 1}, \\ \psi_{j-1,k} &= \sum_n h_{k-2n} \psi_{j,n}, \; j \leq -1, \; k \in \overline{0, L_{j-1} - 1}. \end{aligned}$$

Since $\psi_{j,\tau}(t) = \psi_{j,0}(t - \tau)$, it follows that within each scale j, the wavelet functions are translated versions of each other, each with compact support of length L_j, which increases with the coarser scale (i.e. with decreasing j). By assuming that the time series lives at scale 0, the support of the wavelets on the finest scale is fixed, $L_{-1} = N_h$, and independent of T.

We shall now also introduce the discrete autocorrelation wavelets of Nason *et al.* (2000).

Definition 6.2.5. *The discrete autocorrelation wavelet at scale $j \leq -1$ is defined as*

$$\Psi_j(\tau) = \sum_{k=max\{0,\tau\}}^{L_j-1+min\{0,\tau\}} \psi_{j,k}(0)\psi_{j,k}(\tau),\ \tau \in \mathbb{Z}. \tag{6.5}$$

Each $\Psi_j(\cdot)$ is symmetric about 0, has compact support $[1-L_j, L_j-1]$ and the family $\{\Psi_j(\cdot)\}_{j\leq -1}$ is linearly independent (Nason *et al.* (2000)).

The inner product of any two discrete autocorrelation wavelets, $\Psi_j(\cdot)$ and $\Psi_l(\cdot)$, is given by

$$A_{j,l} = \sum_{\tau=1-\min\{L_j,L_l\}}^{\min\{L_j,L_l\}-1} \Psi_j(\tau)\Psi_l(\tau),\ j,l \leq -1,$$

and we will denote $A_J = (A_{j,l})_{j,l\in\overline{-J(T),-1}}$.

LSW processes.

In what follows we give the main points of the formal definition of a LSW process, and the interested reader can refer to Nason *et al.* (2000) for the complete definition.

Definition 6.2.6. *A sequence of stochastic processes $\{(X_{t,T})_{t\in\overline{0,T-1}},\ T = 2^{J(T)} = 1, 2, ...\}$ is a zero-mean LSW process if it admits the following representation*

$$X_{t,T} = \sum_{j=-J(T)}^{-1} \sum_{k\in\mathbb{Z}} w_{j,k;T}\psi_{j,k}(t)\xi_{j,k}, \tag{6.6}$$

where $\psi_{j,k}(t)$ is a nondecimated discrete wavelet at scale j and location k, $w_{j,k;T}$ is its corresponding amplitude and $\{\xi_{j,k}\}_{j,k}$ is a sequence of zero-mean, orthonormal random variables.

Within each scale j, the evolution of the amplitudes $\{w_{j,k;T}\}_{k\in\overline{0,T-1}}$ is regulated by the Lipschitz continuous function $W_j(\cdot)$, defined for rescaled time $z = \frac{k}{T}$.

Note that as before, we (somewhat abusively) refer to the non-random part of the building blocks coefficients under the name of amplitudes. The functions $\{W_j(\cdot)\}_j$ are the equivalent of the asymptotic transfer function in definition 6.2.3, and control the degree of local stationarity of the process by forcing the amplitudes $\{w_{j,k;T}\}_k$ to vary slowly within each level.

By analogy with the evolutionary spectral density (see previous subsection), for the LSW process defined above, an *evolutionary wavelet spectrum* $\{S_j(\cdot)\}_{j\in\overline{-J(T),-1}}$ can be defined by

$$S_j(z) = |W_j(z)|^2 = \lim_{T\to\infty}|w_{j,\lfloor zT\rfloor;T}|^2, \text{ where } z \in (0,1) \tag{6.7}$$

and $\lfloor zT\rfloor$ denotes the largest integer not exceeding zT. Note that asymptotically, when $T \to \infty$ it follows that $J(T) \to \infty$, which consequently expands the 'span' of the scale j to $j \leq -1$. The spectrum defined above quantifies the contribution to the process variance made at location z and scale j. Nason *et al.* (2000) proved that any stationary process with absolutely summable autocovariance is also a LSW process, and in this case $\{W_j(\cdot)\}_j$ is a constant function within each scale, hence $\{S_j(\cdot)\}_j$ also depends only on the scale j.

For fixed T, the autocovariance of the process $(X_{t,T})_{t\in\overline{0,T-1}}$ depends both on the lag, τ and on the rescaled time location, z, and it is denoted by $c_T(z,\tau) = \mathrm{cov}(X_{\lfloor zT\rfloor}, X_{\lfloor zT\rfloor+\tau})$.

Nason *et al.* (2000) show that the autocovariance function $c_T(\cdot\,,\cdot)$ tends to an (asymptotic) local autocovariance $c(\cdot\,,\cdot)$: $|c_T(z,\tau) - c(z,\tau)| = O(T^{-1})$, where $c(z,\tau)$ is defined in the following.

Definition 6.2.7. *The local autocovariance function of a LSW process defined in 6.2.6 is given by*

$$c(z,\tau) = \sum_{j=-\infty}^{-1} S_j(z)\Psi_j(\tau). \tag{6.8}$$

Although representation (6.6) of a LSW process is not unique, the evolutionary wavelet spectrum is unique in terms of the local autocovariance, and

vice versa (Nason *et al.* (2000)).

The linear independence of the family $\{\Psi_j(\cdot)\}_{j\leq -1}$ ensures the invertibility of the covariance–spectrum representation (see Nason *et al.* (2000) for proof)

$$S_j(z) = \sum_{l=-\infty}^{-1} A_{j,l}^{-1} \left(\sum_{\tau} c(z,\tau)\Psi_l(\tau) \right), \tag{6.9}$$

where $A_J^{-1} = (A_{j,l}^{-1})_{j,l\in\overline{-J(T),-1}}$ is the inverse of the matrix A_J previously introduced.

Hence while for stationary time series the associated covariance function and spectral density function are Fourier pairs, for LSW processes they are 'wavelet conjugated'.

Relations (6.8) and (6.9) highlight that if we can estimate the spectrum $\{S_j(\cdot)\}_j$, then we can produce an estimate for the local autocovariance $c(\cdot,\cdot)$, and vice versa. Let $\hat{S}_j(z)$ be a spectrum estimator, then taking $\hat{c}(z,\tau) = \sum_{j=-J(T)}^{-1} \hat{S}_j(z)\Psi_j(\tau)$ we obtain an estimator for $c(z,\tau)$. For certain choices of $\hat{S}_j(z)$, the estimator $\hat{c}(z,\tau)$ enjoys good properties, such as consistency (see proposition 5 of Nason *et al.* (2000)).

So the focus is to obtain a well-behaved estimator for the spectrum. To achieve this, Nason *et al.* (2000) introduced the *wavelet periodogram of a LSW process* $(X_{t,T})_{t\in\overline{0,T-1}}$, this time constructed with respect to the nondecimated discrete wavelet family $\{\psi_{j,k}(t)\}_{j,k}$,

$$I_{k,T}^j = d_{j,k;T}^2, \tag{6.10}$$

where $d_{j,k;T} = \sum_{t=0}^{T-1} X_{t,T}\psi_{j,k}(t)$ is the empirical wavelet coefficient at scale j and location k.

For $z \in (0,1)$, denote by $\underline{I}_T(z) = (I_{\lfloor zT\rfloor,T}^j)_{j\in\overline{-J(T),-1}}$ the (vector) wavelet periodogram and by $\underline{S}(z) = (S_j(z))_{j\in\overline{-J(T),-1}}$ the (vector) evolutionary wavelet spectrum.

Nason *et al.* (2000) show that

$$E(\underline{I}_T(z)) = A_J \underline{S}(z) + O(T^{-1}),\ z \in (0,1), \tag{6.11}$$

which implies for $z = \frac{k}{T}$

$$E(I^j_{k,T}) = \sum_{l=-J(T)}^{-1} A_{j,l} S_l(\frac{k}{T}) + O(T^{-1}). \tag{6.12}$$

So the expected value of the wavelet periodogram is (asymptotically) a linear combination of wavelet spectrums, and a *corrected vector of periodograms*, $\underline{L}(z) = (L^j_{\lfloor zT \rfloor,T})_{j \in \overline{-J(T),-1}}$ will be used instead for estimating $\underline{S}(z)$:

$$\underline{L}(z) = A_J^{-1} \underline{I}_T(z).$$

Relation (6.11) shows that $\underline{L}(z)$ is asymptotically an unbiased estimator for the evolutionary wavelet spectrum, $\underline{S}(z)$ for all $z \in (0,1)$.

However, Nason *et al.* (2000) show that $\underline{I}_T(z)$ has an asymptotically non-vanishing variance, so it is not a consistent estimator for the wavelet spectrum. To obtain consistency, $I^j_{\lfloor zT \rfloor,T}$ will be first smoothed as a function of z within each scale j. Then correction with A_J^{-1} of the smoothed $\underline{I}_T(z)$ will provide a wavelet spectrum estimator, $(\hat{S}_j(z))_{j \in \overline{-J(T),-1}}$. For properties of this estimator, the reader is referred to Nason *et al.* (2000).

6.3 LSW processes with missing observations: heuristics of the problem

As we have already pointed out, time series with missing observations frequently arise in practice. In what follows, we shall assume that we observe a nonstationary time series that belongs to the class of LSW processes (introduced in section 6.2.3) which features missing observations. In this context,

we will address the problem of estimating the evolutionary wavelet spectrum $\{S_j(\cdot)\}_j$ associated with the process, analyze the problems that arise due to the missing observations and propose a second generation wavelet-based construction for estimating the spectrum.

In this section we will give a heuristic explanation of the proposed approach, and illustrate the discussion with an example. The next section will be concerned with formalising our proposal and investigating some of its properties.

We shall assume that for some T we observe $(X_{t,T})_{t\in\overline{0,T-1}}$, where $X_{t,T}$ admits the representation from definition 6.2.6,

$$X_{t,T} = \sum_{j=-J(T)}^{-1} \sum_{k\in\mathbb{Z}} w_{j,k;T}\psi_{j,k}(t)\xi_{j,k},$$

but unlike before, we do not have an observed value $X_{t,T}$ for each $t \in \overline{0, T-1}$. For a LSW process, defined as a sequence of stochastic processes (see definition 6.2.6), there are two ways in which the locations of the missing values can arise for different values of T— we can either assume that the locations change with T, or that the missing time locations corresponding to the smaller T are fixed. These issues need to be further considered for an asymptotic development.

As we have seen in the previous section, for the estimation of the process' characteristics of interest, such as the evolutionary wavelet spectrum or the local autocovariance function, we first need to obtain the wavelet periodogram, $I^j_{k,T} = d^2_{j,k;T}$ for each $j \in \{-J(T), \ldots, -1\}$ and $k \in \{0, \ldots, T-1\}$. In the computation of the empirical wavelet coefficients $d_{j,k;T}$, all process values $(X_{t,T})_{t\in\overline{0,T-1}}$ are involved since $d_{j,k;T} = \sum_{t=0}^{T-1} X_{t,T}\psi_{j,k}(t)$.

We now confront the fact that the previous sums would involve missing observations, due to our model formulation. Possible ways for circumventing this problem are: (i) estimate 'somehow' the missing values, or (ii) use a different wavelet decomposition to produce wavelet coefficients at the non-missing locations within each scale, by making use only of the observed process

values.

In the absence of information that would lead to informed estimation of the missing values, in this chapter we shall concentrate on providing a way for estimating the evolutionary wavelet spectrum when the process presents missing observations, by addressing the possible solution in (ii). Since the missing observations induce irregular spacing of the time points, we will propose an approach based on second generation wavelets, constructed using the lifting scheme based on the 'remove one coefficient at a time' paradigm of Jansen *et al.* (2004), presented in chapter 3.

However, once we decide to decompose the observed process using a different wavelet family to $\{\psi_{j,k}(\cdot)\}_{j,k}$, another problem arises. We have already seen that the use of discrete nondecimated wavelets ensures that the empirical wavelet coefficients $d_{j,k;T}$ are generated within *each scale* $j \in \{-J(T), \ldots, -1\}$, *at each location* $k \in \{0, \ldots, T-1\}$. The lifting scheme however, produces exactly one detail coefficients at each design point, which is therefore associated to only one (artificial) scale. This poses the first challenge in our approach.

6.3.1 'Nondecimated' second generation wavelet approach

We shall propose a second generation wavelet method to generate (at least) a wavelet coefficient corresponding to each time location within each scale, where the notion of scale is to be understood in the sense introduced by section 3.3 of chapter 3. We propose a bootstrap type approach for this problem.

We shall modify the lifting scheme removing one coefficient a time introduced by Jansen *et al.* (2001). In the original approach, at each step the point to be removed is chosen according to the size of the integral of its associated scaling function. We propose to modify this approach and *consider all possible permutations that define the removal order of the time points.* Each ordering consequently generates its corresponding empirical wavelet coefficients. If n observations of the process are available, the number of such permutations is

$n!$, which is too large to be feasible for implementation. Hence we shall take a 'large enough' sample of these permutations, and generate the empirical wavelet coefficients correspondingly.

Such a construction ensures that within each artificial scale (defined in section 3.6.2), a distribution of details associated to each location is obtained.

We now illustrate the above discussion with an example.

Example.

Let us take the evolutionary wavelet spectrum $\{S_j(\cdot)\}_{j\leq -1}$, described in formula (6.13), which at the finest level (-1) exhibits a burst of activity, and at the coarser level -4 exhibits a squared sinusoidal behaviour (see figure 6.1).

$$S_j(z) = \begin{cases} 1, & \text{for } j = -1,\ z \in (\frac{180}{256}, \frac{209}{256}), \\ \sin^2(4\pi z), & \text{for } j = -4, \\ 0, & \text{otherwise.} \end{cases} \tag{6.13}$$

Using the 'LSWsim' function implemented in the R-package WaveThresh, available at `http://www.stats.bris.ac.uk/~wavethresh`, we shall first simulate a LSW process of length $T = 256$ corresponding to the above spectrum. We take a random sample of $n = 200$ time points out of the 256, and then record their corresponding 'observed' process values in order to obtain an example of a LSW process with missing observations. An example of such a process (with missing data) appears in figure 6.2, and is represented on an irregular time grid. Note that the sinusoidal character might be guessed in the first half of the realization, but in the second half the burst is masking it. The distribution of the 56 missing time points is represented in figure 6.3, and their locations can be also seen in figure 6.2. We note that the region which features the activity burst has slightly more missing observations, which will probably influence the accuracy of the final spectrum estimator. Also, the estimation will be influenced by the overall proportion of missing observations.

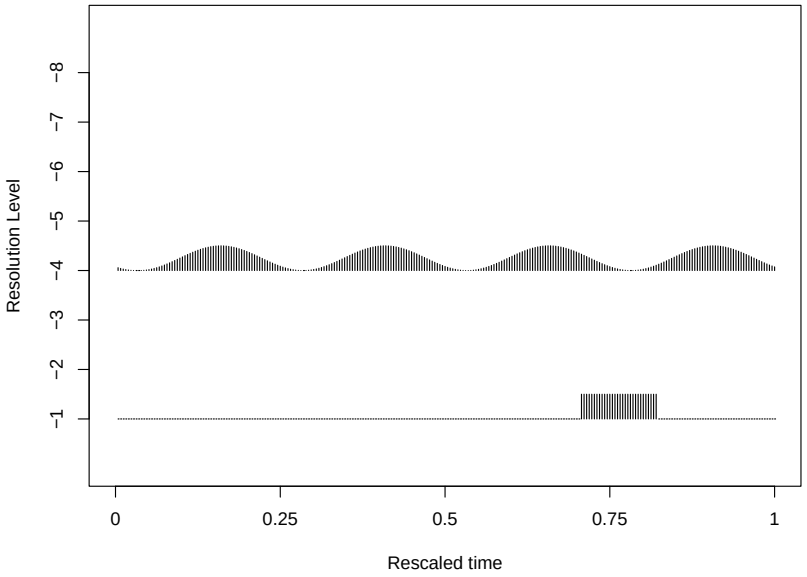

Figure 6.1: Evolutionary wavelet spectrum. The scale runs from finest (bottom) to coarsest (top).

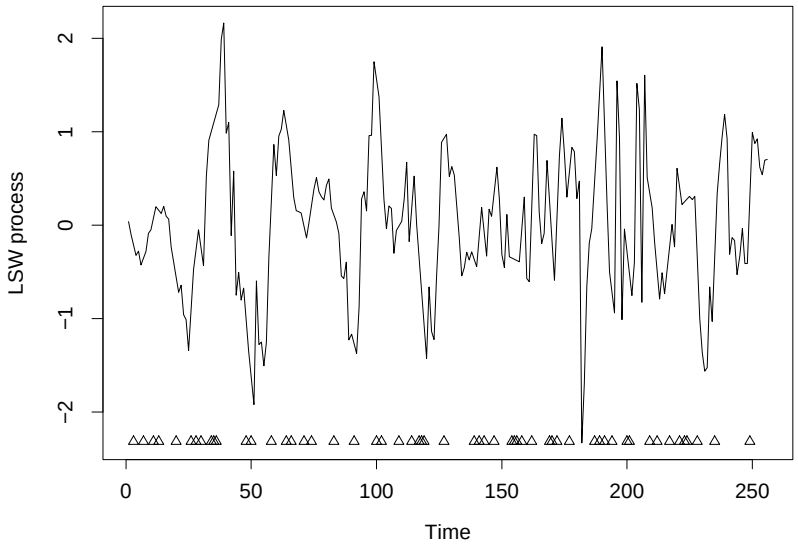

Figure 6.2: Simulated LSW process corresponding to the spectrum in figure 6.1 and featuring missing observations. Triangles indicate locations of missing time points.

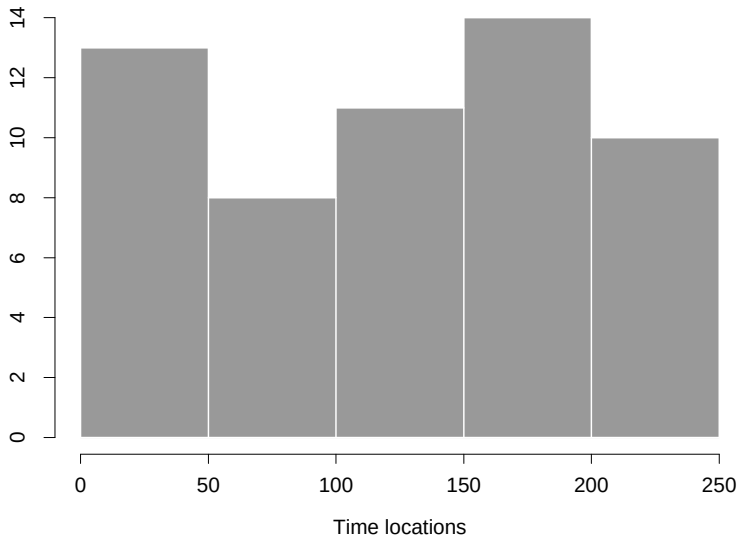

Figure 6.3: Distribution of the missing time points.

For the lifting procedure there are 256! possible removal orderings of the time points that can be used in order to generate the detail coefficients. In what follows we shall take a simple random sample of $m = 1000$ trajectories out of the total 256!. Each trajectory gives the order in which the empirical wavelet coefficients will be produced. For each case, we will modify the lifting scheme such that it follows the corresponding random path. We leave the prediction and update steps unchanged, and use a prediction step that employs linear regression with an intercept and with 2 neighbours in a symmetrical configuration in order to generate the details at each step.

The lifting algorithm generates 5 artificial levels in which the details are arranged, rather than 8 scales as for the initial signal. Except for the observation at time 157 which never appears at the finest artificial level, all the other points appear at all levels. Note that we could increase the number of random sampled trajectories, m, until we ensure that all points appear at all artificial levels. Following this procedure, each time point is associated with a distribution of empirical wavelet coefficients within each artificial level.

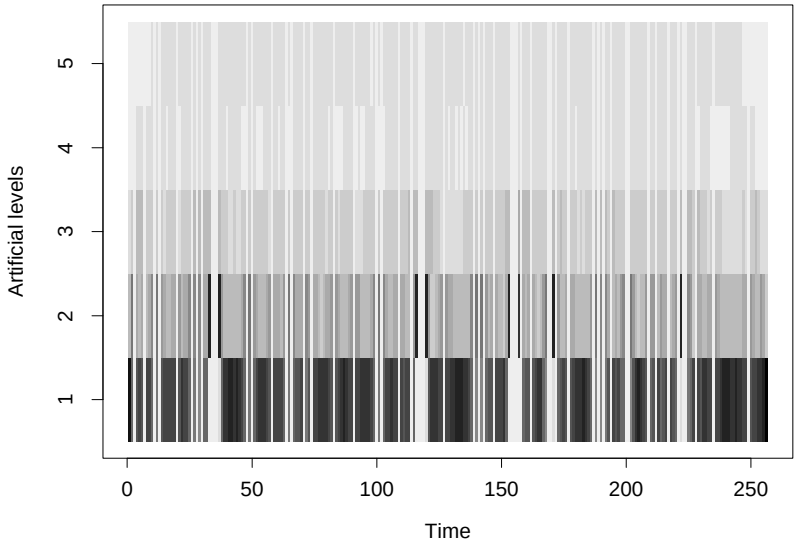

Figure 6.4: Number of times each time point is represented with details within each artificial level. The artificial levels are from finest (1, bottom) to coarsest (5, top). The intensity (darkness) of a pixel corresponds to a higher number of appearances. White lines extending over all artificial levels correspond to the missing time points.

Let us take as an example the observation at time 15. This time point has associated wavelet coefficients as follows: in the first artificial level (the finest level) it appeared 542 times, in the second level 206 times, in the third level 97 times, and 58 and 86 times in the last two levels respectively. Figure 6.4 represents the frequency of appearance of each time point with detail within each artificial level. Since by its construction the finest artificial scale incorporates half of the initial time points, it comes as no surprise that most time points are well represented with detail at this level.

Continuing to investigate the behaviour of the same time point, figure 6.5 gives the histograms of its corresponding detail values within artificial levels 1, 2 and 5. Note how the values of the empirical wavelet coefficients change with the artificial level: the coarser the level, the larger the span of their values, and the less 'concentrated' they are. A direction for future work would be to investigate a possible way of modelling the detail coefficients within each

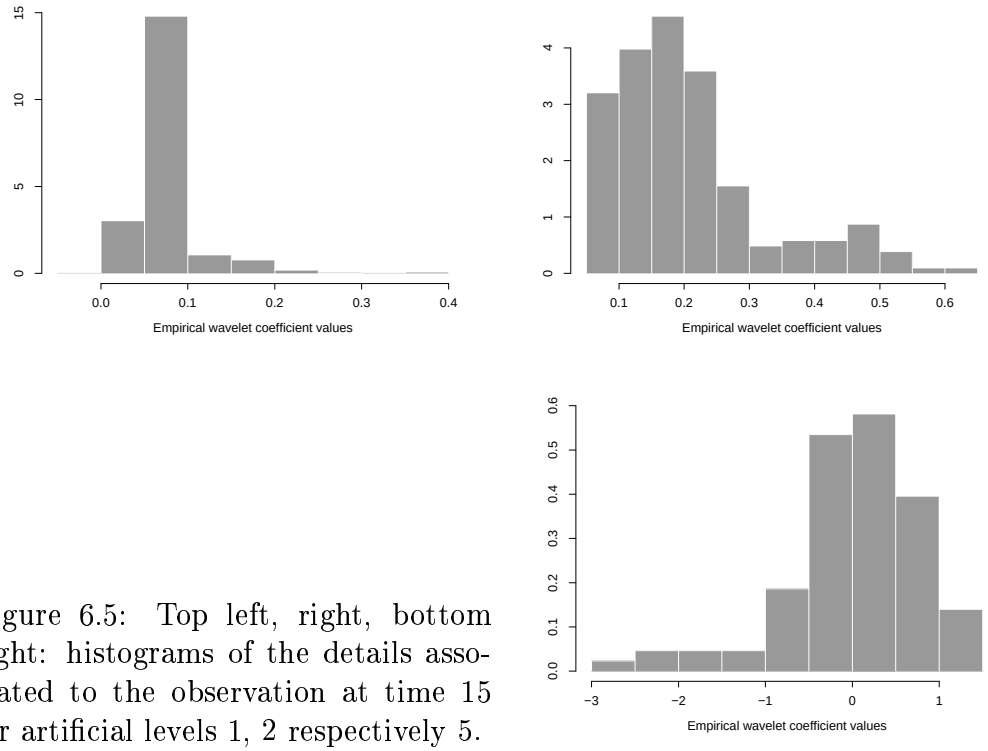

Figure 6.5: Top left, right, bottom right: histograms of the details associated to the observation at time 15 for artificial levels 1, 2 respectively 5.

artificial level, perhaps by using a mixture of normal distributions.

Therefore, in the light of the issues that arose in this example, we are confronted with the next problems. So far, we have obtained a distribution of details within each level, while the approach of Nason *et al.* (2000) generates an unique detail for each time location and scale. Also the scale in the wavelet spectrum is understood in the usual sense for classical wavelets, while in the second generation wavelet approach we construct a different, 'artificial' scale (section 3.6.2).

6.3.2 Proposed second generation wavelet periodogram

As a first approach to constructing a periodogram associated with our model, at each observed time location and at each (artificial) scale we used the squared average of the corresponding details, although further refinements are of course needed.

Figure 6.6 shows that even in such a naive approach, the power burst (and

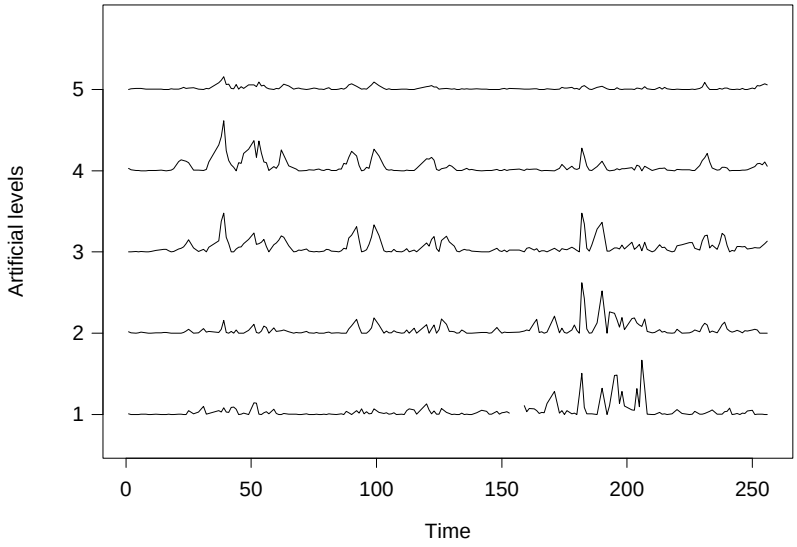

Figure 6.6: Squared average of the details at each sampled time point, within each artificial level (finest scale at the bottom, coarsest scale at the top).

its approximate location) and the sinusoidal component of the true spectrum are vaguely detected. However, there is an obvious 'leak' of power between levels, which was to be expected given the different division into scales induced by second generation wavelet constructions. An issue is therefore to try to understand the correspondence between the initial and new scale decomposition, and construct a wavelet periodogram that parallels the one of Nason *et al.* (2000).

Let us remember that scale in the second generation wavelet context appears as a concept of continuous nature (see section 3.3), and that the artificial levels were constructed to ensure similarity between the detail representation in the lifting scheme using one coefficient at a time and the classical wavelet approach.

In what follows, we decide to explore the consequences of the continuous character of scale, rather than use its artificially constructed discreteness. For each observed time location, we first investigate the magnitude of the squared detail coefficients as a function of the scale to which they are associated.

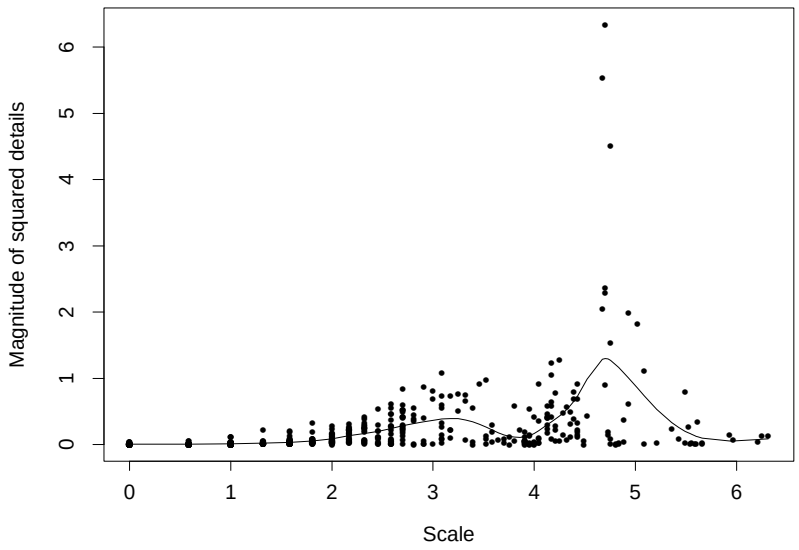

Figure 6.7: Magnitude of the squared details associated to the observation at time 15, versus ($\log_2$ of) their associated integral lengths. The superimposed curve is obtained by smoothing using a cubic spline.

In practice, to ensure that the final periodogram representation parallels the classical one, instead of the actual scale values (i.e. the integral lengths corresponding to the scaling functions) we used their $\log_2$ values.

Figure 6.7 shows that there is lot of variation in this representation, so we followed a nonparametric regression approach and estimated by means of a linear smoother the (true) function that links the magnitude of the squared details to their corresponding scale.

For each observed time location, we obtain the estimated values of the smoothed squared details corresponding to a discrete set of 'evaluation' scales (on the $\log_2$ range mentioned above). This set can be tuned as a finer or coarser division of the scale range, whichever is more appropriate to the problem at hand.

The array that gives the estimated squared detail at each time location and 'evaluation' scale is our proposed periodogram.

Figure 6.8 gives two examples of such a periodogram, each corresponding

to a different 'evaluation' scale. The range of these scales is roughly 0 to 8 (in a continuous manner, with smaller values corresponding to finer scale), and an approximate correspondence can be established with the initial discrete levels $-1, \ldots, -8$. Note though that since not all time points are associated to integral lengths spanning the whole 0 to 8 range, missing values appear at the bottom and top rows of the matrices represented in figure 6.8. However, the use of the finer scale diminishes this problem to some extent. Observe that both the burst and the (4) squared sinusoidal peaks are detected within the correct scales. However, the region approximately between times 150 to 175 does not contain much signal. This might be due to having a slightly higher proportion of missing observations from that area (figure 6.3), than from the rest.

In what follows, guided by the previous heuristic discussion, we shall embed our proposed approach into a formal framework. We will also establish a connection between the initial evolutionary wavelet spectrum and our proposed periodogram, in order to make a step towards a corrected periodogram.

6.4 Formal approach

This section, although organised as a theorem–proof environment, is not intended to provide an asymptotic theory for our construction. Its purpose is to help the understanding of the relationship between the evolutionary wavelet spectrum and our proposed estimator, and provide the grounds for future constructions which would deal with the estimator's bias and power diffusion between scales. For the future it would be interesting to set up a rigorous framework that would allow for asymptotic considerations of our approach.

We shall start with a realization of a LSW process, $(X_{t,T})_{t \in \overline{0,T-1}}$ (for some T), which features missing observations, i.e. at some time points we do not have the corresponding X values.

We define T independent identically distributed Bernoulli random vari-

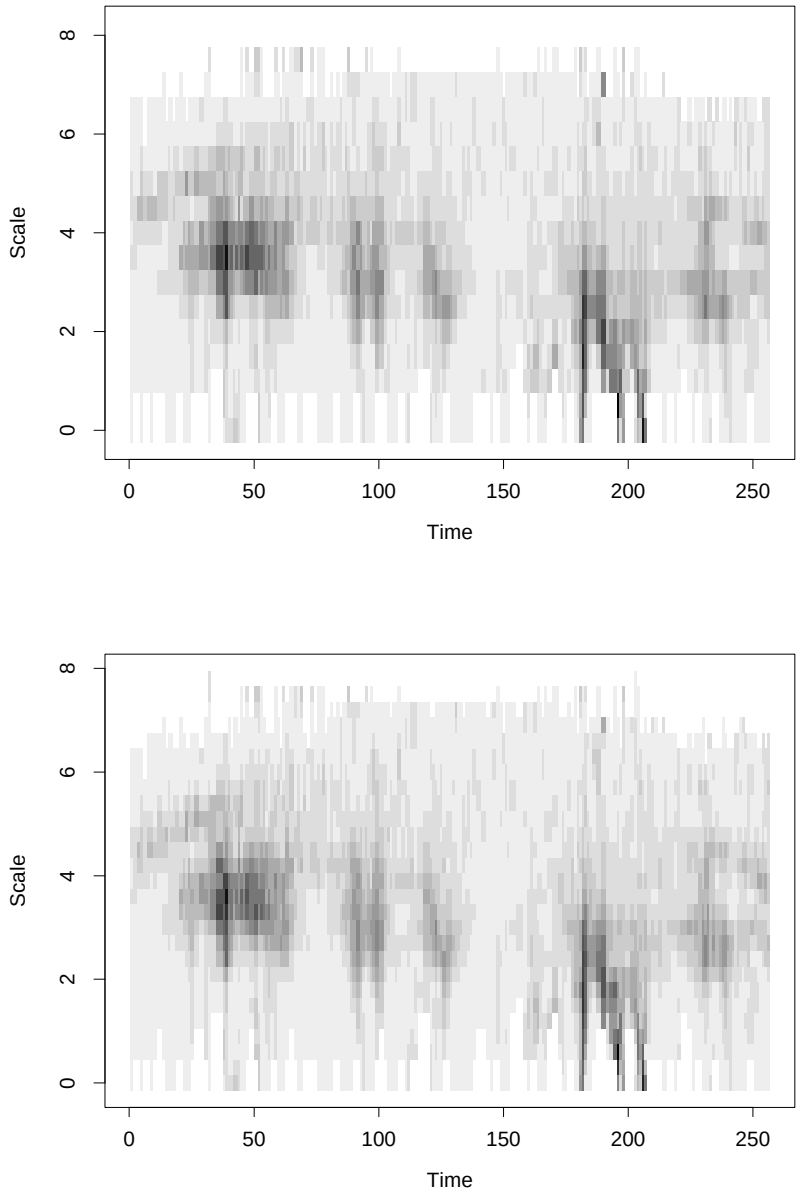

Figure 6.8: Proposed 'raw' wavelet periodograms to estimate the wavelet spectrum of figure 6.1. The smoothed squared details are represented on two different 'evaluation' scales (with 17, respectively 27 equally spaced divisions for the interval 0–8), with the bottom picture corresponding to the finer scale division. In each plot, the scale gets coarser from bottom upwards and darker pixels correspond to a higher estimated spectrum value.

ables, which model the appearance of each time point $t \in \overline{0, T-1}$ in the final set of data, i.e. $I_t \sim \text{Bernoulli}(p)$, where $(1-p)$ is the probability of each initial time point of being missing from the final collection of observations.

The number of observations on the process, let us denote it by n, is then given by $n = \sum_{t=0}^{T-1} I_t$, and consequently $n \sim \text{Bin}(T, p)$. Let us denote those $t \in \overline{0, T-1}$ for which $I_t = 1$ by $t_1, t_2, \ldots, t_n$, and the whole set of time points corresponding to observations on the process by $\mathscr{S} = \{t_1, t_2, \ldots, t_n\}$. We will use the notation $I_{\mathscr{S}}$ for the vector of $(I_{t_1}, I_{t_2}, \ldots, I_{t_n})$, and similarly for the set of missing time points, $I_{\bar{\mathscr{S}}} = (I_t)_{t \in \bar{\mathscr{S}}}$ where $\bar{\mathscr{S}} = \{0, 1, ..., T-1\} \setminus \mathscr{S}$.

Throughout this section we will be working conditional on the time locations corresponding to observations on the process being fixed. In other words, we will assume that $(I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0})$, which in practice means that we have available information at n locations, $t_1, \ldots, t_n$ and we ignore the random character of these locations.

6.4.1 Constructing a 'nondecimated' lifting transform

In what follows, we shall exploit the construction of the lifting scheme removing 'one coefficient at a time', presented in section 3.3 of chapter 3. We aim to propose a second generation wavelet method through which we obtain (at least) an empirical wavelet coefficient for each location $(t_i)_{i \in \overline{1,n}}$ at all scales, where the notion of scale is to be understood in the sense implied by the lifting scheme (see section 3.3).

We will focus on the flexibility generated by the *order* of producing the detail coefficients. In the approach introduced by Jansen *et al.* (2001), the order of transforming scaling coefficients into detail coefficients is established by using the integral lengths of the scaling functions, which account for the 'span' of each point. Here we propose to generalize this criterion, and *allow for full flexibility in choosing the order of obtaining the detail coefficients.*

We will **modify the lifting transform** to accommodate a random order

of generating the wavelet coefficients, while the prediction and update steps will be left unchanged. Then our proposal is to *repeatedly apply the modified version of the lifting transform using one coefficient at a time, every time following a different 'path'*. Here by *'path'* (or trajectory) we mean the vector that gives the order of removing the points, and consequently of generating the detail coefficients.

The n time points $t_1, ..., t_n$ can be arranged in (ordered) vectors of length n in $n!$ ways. Out of this sample space, we will randomly extract say m such orderings, which will give the 'paths' that the (modified) lifting algorithm will take.

For each selected trajectory, the modified lifting transform will generate a set of detail coefficients. Let us denote the n-dimensional (row) vector of detail coefficients by $\underline{d}^T = (d_{t_i,T})_{i\in\overline{1,n}}$.

Using the matrix representation of the wavelet transform, we can write

$$\begin{pmatrix} d_{t_1,T} \\ \cdots \\ d_{t_n,T} \end{pmatrix} = R \begin{pmatrix} X_{t_1,T} \\ \cdots \\ X_{t_n,T} \end{pmatrix},$$

where $R \in \mathscr{M}_{n,n}$ is the matrix built in the process. This is in fact the matrix $\tilde{W}$ from section 3.5.2, but here we chose not to use this notation in order to avoid any confusion with the notation of definition 6.2.6.

From the above it follows that $d_{t_i,T} = \sum_{j=1}^{n} r_{i,j} X_{t_j,T}$, $\forall i \in \overline{1,n}$, and each detail is a linear combination of the observed $X_{t_i,T}$'s, $i \in \overline{1,n}$.

The vector of details, $\underline{d}^T$ has a random character, inherited from the process $(X_{t_i,T})_{i\in\overline{1,n}}$. The elements of the matrix R depend on the prediction and update filters (see section 3.5.2), which in their turn depend only on the time locations for a linear transform, and of course on the regression order used in the prediction step. Therefore, since we work conditional on having fixed design points, $(t_k)_{k\in\overline{1,n}}$, the elements of the matrix R can be assumed non-random. Moreover, due to the characteristics of the lifting construction that

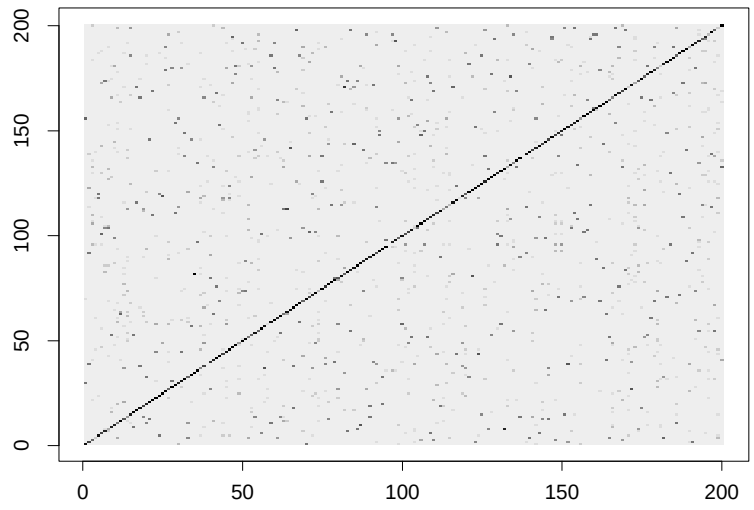

Figure 6.9: Matrix R for the process in the previous example and LP1S (see section 3.6.1). Darker pixels correspond to higher (absolute) values of the elements of R.

removes one coefficient at a time, at each stage of the transform, the prediction and update weights are non-zero only for those locations corresponding to the neighbourhoods used at each step for prediction, hence the matrix R is sparse (for an illustration refer to figure 6.9). For each of the m 'paths', we apply the lifting algorithm and generate a different matrix $R^1, ..., R^m$. Correspondingly we get m sets of wavelet vectors, $\underline{d}^{1,T}, ..., \underline{d}^{m,T}$.

Hence for each time location t_k we obtain a set of details, $\{d^{\alpha}_{t_k,T}\}_{\alpha \in \overline{1,m}}$. Each detail $d^{\alpha}_{t_k,T}$ is associated to an interval that intuitively accounts for the 'span' of time location t_k at the respective stage in the algorithm. We shall denote the length of this interval by $l^{\alpha}_{t_k,T}$, which will provide our measure of scale. Therefore, *at each time location we obtain a set of details, which we can model as a function of their scale.*

Simply in order to make a parallel with the wavelet decomposition obtained using nondecimated classical wavelets, we can consider the scale division in artificial levels, as explained in section 3.6. Then for each time location and

each artificial scale, we obtain a set of details (rather than exactly one empirical wavelet coefficient, as in the classical approach using nondecimated wavelets).

As we have already pointed out in the previous section, an approach would be to further model and investigate the distributions of details that appear for each time location and artificial scale. However, in what follows we shall not pursue this avenue, but rather exploit the continuous nature of scale that arises in the second generation wavelet context.

6.4.2 Periodogram associated with the 'nondecimated' lifting transform

In section 6.2.3 we introduced the periodogram proposed by Nason *et al.* (2000) for estimating the evolutionary wavelet spectrum. This was an array filled in with the values of the squared detail coefficients corresponding to each level and time location, where the level had the usual multiresolution meaning on a $\log_2(T)$ scale (see equation (6.10)).

Our aim is to construct an array similar to the wavelet periodogram described above. Since for second generation wavelet constructions the scale associated to each detail coefficient has a continuous character, we shall 'discretize' the scale in a different manner than when constructing the artificial levels.

To ensure comparability with the construction from Nason *et al.* (2000), we shall choose a set of 'evaluation' lengths, which we denote by $l^1, l^2, \ldots, l^{J^*}$ for some J^*. Through J^*, we are in fact tuning the proposed discreteness of the scale.

For each scale l^i with $i \in \{1, \ldots, J^*\}$ and each time location t_k with $k \in \{1, \ldots, n\}$, we need to 'manufacture' a corresponding value of the squared detail.

This can be achieved by taking for each fixed location (say t_k) a nonparametric regression approach in modelling the magnitude of the associated

squared details $(d^1_{t_k,T})^2, \ldots, (d^m_{t_k,T})^2$ as a function of the corresponding interval lengths $l^1_{t_k,T}, \ldots, l^m_{t_k,T}$ (refer to figure 6.7). For each time location t_k, we will denote by $f_{t_k,T}$ the function we want to estimate. In practice, we are interested in its values at the set of 'evaluation' lengths, $l^1, l^2, \ldots, l^{J^*}$.

In other words, for each t_k with $k \in \overline{1,n}$ we model the data as

$$(d^\alpha_{t_k,T})^2 = f_{t_k,T}(l^\alpha_{t_k,T}) + \varepsilon_\alpha, \quad \alpha \in \overline{1,m}, \tag{6.14}$$

and we want to obtain an estimate $\hat{f}_{t_k,T}(l^i)$ for each $i \in \overline{1,J^*}$.

We will estimate each $f_{t_k,T}(\cdot)$ by using a linear smoother, hence

$$\hat{f}_{t_k,T}(l^i) = \sum_{\alpha=1}^{m} K_\alpha(l^i)(d^\alpha_{t_k,T})^2, \quad \forall i \in \overline{1,J^*}, \tag{6.15}$$

where $K_\alpha(l^i)$ are weight functions that are non-zero only for those α values such that $l^\alpha_{t_k,T}$ is in a neighbourhood of l^i. We note that the weights $K_\alpha(\cdot)$ are different for each t_k, but we do not indicate it to avoid cluttering the notation.

The above value of $\hat{f}_{t_k,T}(l^i)$ is an estimate of the magnitude of the squared detail $((d^\alpha_{t_k,T})^2)$ at time t_k associated to the interval length l^i.

The matrix $(\hat{f}_{t_k,T}(l^i))_{i\in\overline{1,J^*},k\in\overline{1,n}}$ (e.g. figure 6.8) corresponds to the raw periodogram $(d^2_{j,k;T})_{j\in\overline{-J(T),-1},k\in\overline{0,T-1}}$ introduced by Nason *et al.* (2000), and it represents our proposed periodogram.

In what follows we aim to obtain $E(\hat{f}_{t_k,T}(l^i)|I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths})$, in order to highlight the relationship between our corresponding 'raw' periodogram and the (unknown) evolutionary wavelet spectrum of the process. This can be viewed as trying to establish the equivalent formula to (6.12) from the LSW approach for our setup here.

6.4.3 Relationship between the proposed periodogram and the evolutionary wavelet spectrum

We are interested in obtaining a formula that would show how the proposed lifting operations work for spectral estimation. In particular, we are interested in an analogous derivation to $E(I^j_{k,T}) = \sum_{l=-J(T)}^{-1} A_{j,l} S_l(\frac{k}{T}) + O(T^{-1})$ from Nason *et al.* (2000). The following development is only meant to parallel such a formula and is not, at this stage, a rigorous asymptotic development. It is clear from the illustrations shown thus far (e.g. figure 6.8) that some kind of blurring is present in our proposed periodogram, and the formula we derive in this section suggests that the blurring can, in principle, be corrected.

We shall first obtain the covariance structure of the wavelet coefficients as a function of the initial spectrum, both for each run of the lifting scheme, and between different runs. In other words, we are interested in $\mathrm{cov}(d^\alpha_{t_i,T}, d^\beta_{t_{i'},T} | I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths})$, $\forall i, i' \in \overline{1,n}, \forall \alpha, \beta \in \overline{1,m}$. Notice that we are conditioning on the trajectories being fixed, rather than take into account their randomness, as this would in turn induce randomness in the matrices $R^1, ..., R^m$.

As a first step, the following lemma will establish a link between the variance–covariance matrix of the detail coefficients and the (sample) autocovariance matrix of the initial LSW process at the observed time points.

Lemma 6.4.1. *Under the previous notation, the following relations hold:*

$$\begin{aligned} &\mathrm{cov}(d^\alpha_{t_i,T}, d^\beta_{t_{i'},T} | I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \textit{fixed paths}) \\ &= \sum_{j=1}^{n} \sum_{j'=1}^{n} r^\alpha_{i,j} \,\mathrm{cov}(X_{t_j,T}, X_{t_{j'},T} | I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}) r^\beta_{i',j'}, \end{aligned} \tag{6.16}$$

for $\forall \alpha, \beta \in \overline{1,m}$, $\forall i, i' \in \overline{1,n}$.

Proof.

For any $\alpha \in \overline{1,m}$, let us denote

$$\Sigma^{\alpha,T} = \mathrm{var}((\underline{d}^{\alpha,T})^T | I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths}) \in \mathscr{M}_{n,n}.$$

Also, for any $\alpha,\ \beta \in \overline{1,m}$, we shall use the notation

$$A^{\alpha,\beta,T} = \mathrm{cov}((\underline{d}^{\alpha,T})^T, (\underline{d}^{\beta,T})^T | I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths}) \in \mathscr{M}_{n,n}.$$

Therefore, the variance–covariance matrix of the vector $(\underline{d}^{\alpha,T}, \underline{d}^{\beta,T})^T \in \mathscr{M}_{2n,1}$ takes the form

$$\mathrm{var}\left(\begin{pmatrix} (\underline{d}^{\alpha,T})^T \\ (\underline{d}^{\beta,T})^T \end{pmatrix} | I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths} \right) = \begin{pmatrix} \Sigma^{\alpha,T} & A^{\alpha,\beta,T} \\ (A^{\alpha,\beta,T})^T & \Sigma^{\beta,T} \end{pmatrix}. \tag{6.17}$$

Since

$$(\underline{d}^{\alpha,T})^T = R^{\alpha}((X_{t_i,T})_{i\in\overline{1,n}})^T, \quad \alpha \in \overline{1,m}, \tag{6.18}$$

it follows that for any $\alpha, \beta \in \overline{1,m}$ we have

$$\begin{pmatrix} (\underline{d}^{\alpha,T})^T \\ (\underline{d}^{\beta,T})^T \end{pmatrix} = \begin{pmatrix} R^{\alpha} \\ R^{\beta} \end{pmatrix} ((X_{t_i,T})_{i\in\overline{1,n}})^T.$$

Hence

$$\mathrm{var}\left(\begin{pmatrix} (\underline{d}^{\alpha,T})^T \\ (\underline{d}^{\beta,T})^T \end{pmatrix} | I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths} \right) = \begin{pmatrix} R^{\alpha}\Sigma^{(T)}(R^{\alpha})^T & R^{\alpha}\Sigma^{(T)}(R^{\beta})^T \\ R^{\beta}\Sigma^{(T)}(R^{\alpha})^T & R^{\beta}\Sigma^{(T)}(R^{\beta})^T \end{pmatrix}, \tag{6.19}$$

where $\Sigma^{(T)} = \mathrm{var}(((X_{t_i,T})_{i\in\overline{1,n}})^T | I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}) = (\sigma_{j,k;T})_{j,k\in\overline{1,n}}$ is the (symmetric) variance-covariance matrix of the observed signal (with missing observations), having assumed that the missing points are deterministic rather than random quantities.

Using relation (6.17), we obtain

$$\Sigma^{\alpha,T} = R^{\alpha}\Sigma^{(T)}(R^{\alpha})^T, \quad \forall \alpha \in \overline{1,m}, \tag{6.20}$$

$$A^{\alpha,\beta,T} = R^{\alpha}\Sigma^{(T)}(R^{\beta})^T, \quad \forall \alpha,\beta \in \overline{1,m}. \tag{6.21}$$

Written explicitly, equation (6.21) takes the form

$$\operatorname{cov}(d^{\alpha}_{t_i,T}, d^{\beta}_{t_{i'},T} | I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths}) = \sum_{j=1}^{n}\sum_{j'=1}^{n} r^{\alpha}_{i,j} \operatorname{cov}(X_{t_j,T}, X_{t_{j'},T} | I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}) r^{\beta}_{i',j'}, \tag{6.22}$$

for $\forall \alpha, \beta \in \overline{1,m}$, $\forall i, i' \in \overline{1,n}$. ■

So far we expressed $\operatorname{cov}(d^{\alpha}_{t_i,T}, d^{\beta}_{t_{i'},T} | I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths})$ as a linear combination of *sample* autocovariances of the initial process $(X_{t,T})_{t\in\overline{0,T-1}}$ *involving only the observed locations.*

Next, we will extend this relation and link $\operatorname{cov}(d^{\alpha}_{t_i,T}, d^{\beta}_{t_{i'},T} | I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths})$ to the *local* autocovariance of the process.

Proposition 6.4.2. *For $\forall \alpha,\ \beta \in \overline{1,m}$, $\forall i,\ i' \in \overline{1,n}$ we have*

$$\operatorname{cov}(d^{\alpha}_{t_i,T}, d^{\beta}_{t_{i'},T} | I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \textit{fixed paths}) = \sum_{j=1}^{n}\sum_{j'=1}^{n} r^{\alpha}_{i,j} c(\frac{t_j}{T}, t_{j'} - t_j) r^{\beta}_{i',j'} + \tilde{R}_T, \tag{6.23}$$

where $\tilde{R}_T$ is a term of order $O(T^{-1})$.

Proof.
If we let $z_j = \frac{t_j}{T} \in (0,1)$, then the process autocovariance can be written as

$$\operatorname{cov}(X_{t_j,T}, X_{t_{j'},T} | I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}) = \operatorname{cov}(X_{\lfloor z_j T \rfloor}, X_{\lfloor z_j T \rfloor + (t_{j'} - t_j)} | I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}).$$

Therefore, we can write

$$\operatorname{cov}(X_{t_j,T}, X_{t_{j'},T} | I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}) = c_T(\frac{t_j}{T}, t_{j'} - t_j), \tag{6.24}$$

where $c_T(\cdot\,,\cdot)$ is the autocovariance of the LSW process $(X_{t,T})_{t\in\overline{0,T-1}}$, introduced in section 6.2.3, and we assume the conditioning still holds.

From the result in the previous lemma and equation (6.24), we obtain

$$\operatorname{cov}(d^{\alpha}_{t_i,T}, d^{\beta}_{t_{i'},T} | I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths}) = \sum_{j=1}^{n}\sum_{j'=1}^{n} r^{\alpha}_{i,j} c_T(\frac{t_j}{T}, t_{j'} - t_j) r^{\beta}_{i',j'}.$$

Nason *et al.* (2000) proved that the process autocovariance and local autocovariance functions are linked through $c_T(z,\tau) = c(z,\tau) + R_T$, for any rescaled time location z and lag τ, where R_T is a term of magnitude $O(T^{-1})$. From the above relation, the following becomes apparent:

$$\operatorname{cov}(d^{\alpha}_{t_i,T}, d^{\beta}_{t_{i'},T} | I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths}) = \sum_{j=1}^{n}\sum_{j'=1}^{n} r^{\alpha}_{i,j} c(\frac{t_j}{T}, t_{j'} - t_j) r^{\beta}_{i',j'} + \\ + \sum_{j=1}^{n}\sum_{j'=1}^{n} r^{\alpha}_{i,j} R_T r^{\beta}_{i',j'},$$

for $\forall \alpha,\ \beta \in \overline{1,m}$, $\forall i,\ i' \in \overline{1,n}$.

Denote $\tilde{R}_T = R_T \sum_{j=1}^{n}\sum_{j'=1}^{n} r^{\alpha}_{i,j} r^{\beta}_{i',j'}$. Therefore, $\tilde{R}_T = R_T \sum_{j=1}^{n} r^{\alpha}_{i,j} \sum_{j'=1}^{n} r^{\beta}_{i',j'}$. We have pointed out that the matrices associated to a lifting transform removing one coefficient at a time have a sparse character, so for a fixed i the sums of the type $\sum_{j=1}^{n} r^{\alpha}_{i,j}$ only involve a finite number of elements, independent of the magnitude of n. If more data is collected, then there is a chance that the new observations will be involved in $\sum_{j=1}^{n} r^{\alpha}_{i,j}$ for a fixed i, but the combination will still be sparse, hence $\sum_{j=1}^{n} r^{\alpha}_{i,j} := C^{\alpha} < \infty$. As $R_T = O(T^{-1})$, it follows that $\exists\, k < \infty$ such that $|\tilde{R}_T| \leq kT^{-1}C^{\alpha}C^{\beta} < \infty$, so $\tilde{R}_T$ has magnitude $O(T^{-1})$. ■

Using the definition of the local autocovariance (6.8) and equation (6.23), we can obtain an expression of $\operatorname{cov}(d^{\alpha}_{t_i,T}, d^{\beta}_{t_{i'},T} | I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths})$ in terms of the evolutionary wavelet spectrum of the LSW process, $\{S_l(\cdot)\}_l$, and the discrete autocorrelation wavelets, $\{\Psi_l(\cdot)\}_l$ from definition 6.2.5.

More exactly, by replacing the local autocovariance

$$c(z,\tau) = \sum_{l=-\infty}^{-1} S_l(z)\Psi_l(\tau)$$

in equation (6.23), we obtain

$$\mathrm{cov}(d^{\alpha}_{t_i,T}, d^{\beta}_{t_{i'},T}|I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths}) = \sum_{l=-\infty}^{-1}\sum_{j=1}^{n}\sum_{j'=1}^{n} r^{\alpha}_{i,j} r^{\beta}_{i',j'} \Psi_l(t_j - t_{j'}) S_l(\frac{t_j}{T}) + \tilde{R}_T, \tag{6.25}$$

for $\forall \alpha,\ \beta \in \overline{1,m}$, $\forall i,\ i' \in \overline{1,n}$. In the above formula we used the symmetry around 0 of $\{\Psi_l(\cdot)\}_l$. Fryzlewicz (2003) observes that in order to achieve the convergence of the autocovariance $c_T(\cdot,\cdot)$ (proposition 1 of Nason *et al.* (2000)), the 'tail' of the sequence $\{S_j(\cdot)\}_{j\leq -1}$ needs to be controlled. An approach to this would be to allow non-zero contributions to $\{S_j(\cdot)\}_{j\leq -1}$ only from levels say $j \in \{-J', \ldots, -1\}$ for a large enough J', which would in turn mean that l would have a finite range in the above formula (and therefore in the next ones).

Equation (6.25) links $\mathrm{cov}(d^{\alpha}_{t_i,T}, d^{\beta}_{t_{i'},T}|I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths})$, and implicitly $E(d^{\alpha}_{t_i,T} d^{\beta}_{t_{i'},T}|I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths})$, to the (unknown) wavelet spectrum at the (rescaled) locations corresponding to the observed time points, $\{S_l(\frac{t_j}{T})\}_{l,j}$, by involving only tractable coefficients.

We shall now re-write the previous expression in terms of $E(d^{\alpha}_{t_i,T} d^{\beta}_{t_{i'},T}|I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths})$.

Proposition 6.4.3. *For $\forall \alpha,\ \beta \in \overline{1,m}$, $\forall i,\ i' \in \overline{1,n}$ we have*

$$E(d^{\alpha}_{t_i,T} d^{\beta}_{t_{i'},T}|I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \textit{fixed paths}) = \sum_{l=-\infty}^{-1}\sum_{j=1}^{n}\sum_{j'=1}^{n} r^{\alpha}_{i,j} r^{\beta}_{i',j'} \Psi_l(t_j - t_{j'}) S_l(\frac{t_j}{T}) + \tilde{R}_T, \tag{6.26}$$

where $\tilde{R}_T$ is a term of magnitude $O(T^{-1})$.

Proof.

In the LSW model, the sequence of processes $(X_{t,T})_{t\in\overline{0,T-1}}$, $T = 1, 2, \ldots$ is assumed to have zero mean, i.e. $E(X_{t,T}) = 0$, $\forall t \in \overline{0, T-1}$, $\forall T$.

As $E(d^{\alpha}_{t_i,T} d^{\beta}_{t_{i'},T}) = \mathrm{cov}(d^{\alpha}_{t_i,T}, d^{\beta}_{t_{i'},T}) + E(d^{\alpha}_{t_i,T}) E(d^{\beta}_{t_{i'},T})$ and $E(d^{\alpha}_{t_i,T}|I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths}) = \sum_{j=1}^{n} r^{\alpha}_{i,j} E(X_{t_j,T})$, from formula (6.25) we obtain the desired equation. ■

We are now in position to obtain $E(\hat{f}_{t_k,T}(l^i)|I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths})$, which will give us an insight into the relationship between our proposed periodogram and the wavelet spectrum of the process.

Theorem 6.4.4. *For the wavelet periodogram estimators $\hat{f}_{t_k,T}(\cdot)$ constructed in (6.15), and for $\forall\, i \in \overline{1, J^*}$, $k \in \overline{1, n}$ we obtain the following*

$$E(\hat{f}_{t_k,T}(l^i)|I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \textit{fixed paths}) = \textit{Trace}(A^{l^i,k} S^T) + \tilde{R}_T^*, \qquad (6.27)$$

where $\tilde{R}_T^* = O(T^{-1})$, $S = (S_{l,j})_{l \leq -1, j \in \overline{1,n}}$ with $S_{l,j} = S_l(\frac{t_j}{T})$, $A^{l^i,k} = (a_{l,j}^{l^i,k})_{l \leq -1, j \in \overline{1,n}}$ with $a_{l,j}^{l^i,k} = \sum_{j'=1}^{n} \left\{ \sum_{\alpha=1}^{m} K_\alpha(l^i) r_{k,j}^\alpha r_{k,j'}^\alpha \right\} \Psi_l(t_j - t_{j'})$ and $\{K_\alpha(\cdot)\}_\alpha$ are as defined in equation (6.15).

Proof.

Since $\hat{f}_{t_k,T}(l^i) = \sum_{\alpha=1}^{m} K_\alpha(l^i)(d_{t_k,T}^\alpha)^2$, $\forall i \in \overline{1, J^*}$, $\forall k \in \overline{1, n}$, it follows that

$$E(\hat{f}_{t_k,T}(l^i)|I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths}) = \sum_{\alpha=1}^{m} K_\alpha(l^i) E((d_{t_k,T}^\alpha{}^2)|I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths}).$$

By taking $\alpha = \beta$ and $i = i' := k$ in (6.26), we obtain

$$E(\hat{f}_{t_k,T}(l^i)|I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths}) =$$
$$= \sum_{\alpha=1}^{m} K_\alpha(l^i) \left\{ \sum_{l=-\infty}^{-1} \sum_{j=1}^{n} \sum_{j'=1}^{n} r_{k,j}^\alpha r_{k,j'}^\alpha \Psi_l(t_j - t_{j'}) S_l(\frac{t_j}{T}) + \tilde{R}_T \right\},$$
$$= \sum_{l=-\infty}^{-1} \sum_{j=1}^{n} \left[\sum_{j'=1}^{n} \left\{ \sum_{\alpha=1}^{m} K_\alpha(l^i) r_{k,j}^\alpha r_{k,j'}^\alpha \right\} \Psi_l(t_j - t_{j'}) \right] S_l(\frac{t_j}{T}) + \tilde{R}_T \sum_{\alpha=1}^{m} K_\alpha(l^i),$$

$\forall i \in \overline{1, J^*}$, $\forall k \in \overline{1, n}$.

As $a_{l,j}^{l^i,k} = \sum_{j'=1}^{n} \left\{ \sum_{\alpha=1}^{m} K_\alpha(l^i) r_{k,j}^\alpha r_{k,j'}^\alpha \right\} \Psi_l(t_j - t_{j'})$, the above equation can be equivalently written as

$$E(\hat{f}_{t_k,T}(l^i)|I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths}) = \sum_{l=-\infty}^{-1} \sum_{j=1}^{n} a_{l,j}^{l^i,k} S_l(\frac{t_j}{T}) + \tilde{R}_T \sum_{\alpha=1}^{m} K_\alpha(l^i). \qquad (6.28)$$

Therefore,

$$E(\hat{f}_{t_k,T}(l^i)|I_{\mathscr{S}} = \underline{1}, I_{\bar{\mathscr{S}}} = \underline{0}, \text{fixed paths}) = \sum_{l=-\infty}^{-1} (A^{l^i,k} S^T)_{l,l} + \tilde{R}_T \sum_{\alpha=1}^{m} K_\alpha(l^i)$$
$$= \text{Trace}(A^{l^i,k} S^T) + \tilde{R}_T \sum_{\alpha=1}^{m} K_\alpha(l^i).$$

Observe that in order to obtain $(A^{l^i,k} S^T)_{l,l}$ for a fixed time t_k and scale l^i, only the terms corresponding to time locations t_j, $t_{j'}$ such that $(t_j - t_{j'})$ does not exceed the support of the autocorrelation wavelet $\Psi_l(\cdot)$ are contributing to the sum.

Let us denote $\tilde{R}^*_T = \tilde{R}_T \sum_{\alpha=1}^{m} K_\alpha(l^i)$. For finite m, $\tilde{R}^*_T$ has magnitude $O(T^{-1})$ as $\tilde{R}_T$ has magnitude $O(T^{-1})$ from the previous proposition. ■

The result in the previous theorem corresponds to (6.12) in the development of Nason *et al.* (2000). However, our result is conditional on the time locations corresponding to the observations on the process being fixed, and on ignoring the randomness in the lifting trajectories. For further work, it would be interesting to try and eliminate these restrictions, as well as rigorously set a framework in which to investigate the asymptotic behaviour of our estimator.

Equation (6.27) shows that our proposed 'raw' periodogram is not an unbiased estimator for the wavelet spectrum, and it therefore needs correction. This does not come as a surprise, given the similar result that follows from (6.12) for the simpler case of observing a LSW process with no missing observations. Formula (6.27) also highlights that the used smoother will influence the amount of bias.

6.4.4 Matrix formulation

Relation (6.27) indicates a way for proposing a better estimator (than the raw periodogram) for the spectrum matrix S curtailed down to $J(T)$ rows, $S = (S_l(\frac{t_j}{T}))_{l \in \overline{-J(T),-1}, j \in \overline{1,n}}$.

To achieve this, we shall first re-write the unknown spectrum values S into vector format,

$$\underline{\tilde{s}} = ((S_{-1,j})_{j\in\overline{1,n}} \mid (S_{-2,j}))_{j\in\overline{1,n}} \mid \cdots \mid (S_{-J(T),j})_{j\in\overline{1,n}}) \in \mathscr{M}_{1,J(T)\times n}.$$

On the same principle as above, let us put each matrix $A^{l^i,k}$ into vector format, as follows

$$\underline{\tilde{a}}^{l^i,k} = ((a^{l^i,k}_{-1,j})_{j\in\overline{1,n}} \mid (a^{l^i,k}_{-2,j})_{j\in\overline{1,n}} \mid \cdots \mid (a^{l^i,k}_{-J(T),j})_{j\in\overline{1,n}}),$$

where for each $l \in \overline{-J(T),-1}$, $(a^{l^i,k}_{l,j})_{j\in\overline{1,n}}$ is an n-dimensional row vector, hence $\underline{\tilde{a}}^{l^i,k} \in \mathscr{M}_{1,J(T)\times n}$, $\forall k \in \overline{1,n}$, $\forall i \in \overline{1,J^*}$.

For each observed point t_k, i.e. for each $k \in \overline{1,n}$, define the associated matrix

$$\tilde{A}^k = \begin{pmatrix} \underline{\tilde{a}}^{l^1,k} \\ \underline{\tilde{a}}^{l^2,k} \\ \cdots \\ \underline{\tilde{a}}^{l^{J^*},k} \end{pmatrix} \in \mathscr{M}_{J^*,J(T)\times n}.$$

Also define for each $k \in \overline{1,n}$

$$\underline{\hat{f}}^k = (\hat{f}_{t_k,T}(l^1), \hat{f}_{t_k,T}(l^2), \ldots, \hat{f}_{t_k,T}(l^{J^*})) \in \mathscr{M}_{1,J^*}.$$

In this notation, an estimator for the vector of wavelet spectrum values corresponding to the observed locations, $\underline{\tilde{s}}$, can be obtained by solving the following system with $J(T)\times n$ unknowns and $J^*\times n$ equations

$$\begin{pmatrix} \tilde{A}^1 \\ \tilde{A}^2 \\ \cdots \\ \tilde{A}^n \end{pmatrix} \underline{\tilde{s}}^T = \begin{pmatrix} (\underline{\hat{f}}^1)^T \\ (\underline{\hat{f}}^2)^T \\ \cdots \\ (\underline{\hat{f}}^n)^T \end{pmatrix}. \tag{6.29}$$

As a direction for future research, it would be very interesting to establish

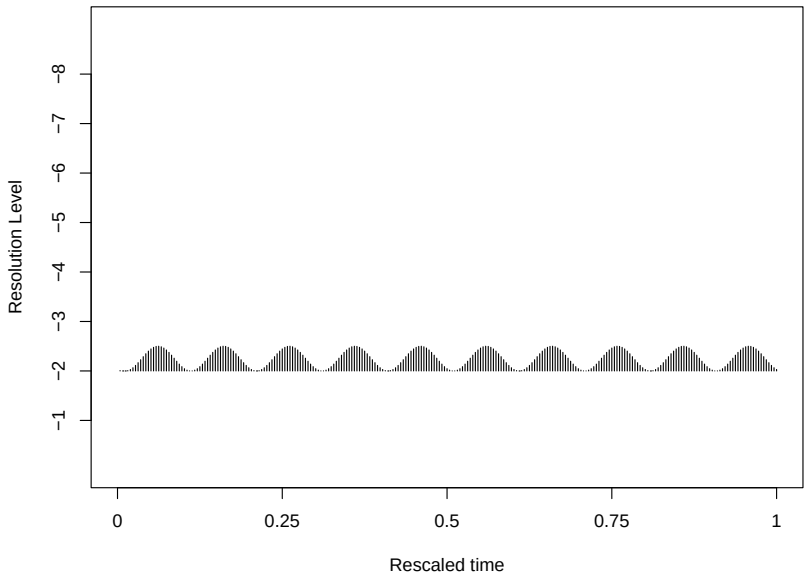

Figure 6.10: Evolutionary wavelet spectrum. The scale runs from finest (bottom) to coarsest (top).

various possible solutions for the above system, and investigate their properties as estimators for S.

In what follows we shall give two examples, to furthermore illustrate our construction.

Examples.

1. Let us start with a LSW process corresponding to a evolutionary wavelet spectrum that at the (finer) level -2 has a squared sinusoidal character, as in figure 6.10 (see equation 6.30). A realization of length $T = 256$ of such a process is displayed in figure 6.11 (top).

$$S_j(z) = \begin{cases} \sin^2(10\pi z), & \text{for } j = -2, \\ 0, & \text{otherwise.} \end{cases} \tag{6.30}$$

In order to simulate a process with missing observations, we shall take a random sample of $n = 200$ time points out of the 256, and record the pro-

cess observations corresponding to these time locations. The sampled process now has an irregular time spacing, and is represented in figure 6.11 (bottom). Figure 6.12 shows the roughly uniform distribution of the missing values (remember that our process times were only running up to 256, hence the shorter last bar in the plot).

To construct our proposed periodogram, we shall first take a simple random sample of $m = 1500$ trajectories out of the total 256!, and for each trajectory obtain a set of corresponding empirical wavelet coefficients by applying the modified lifting scheme using linear regression with an intercept and with 2 symmetrical neighbours.

The algorithm generates 5 artificial levels, and in this example all time points are represented within all artificial levels.

Therefore, each time point is associated with a distribution of wavelet coefficients at each artificial level. We will investigate the observation at time 183. This location has associated wavelet coefficients as follows: in the first artificial level (the finest level) it appeared 929 times, in the second level 271 times, in the third level 148 times, and 52 and 89 times in the last two levels respectively. Figure 6.13 represents the frequency of appearance of each point within each artificial level.

Figure 6.14 shows the histograms associated to the same time point at the first, fourth and fifth artificial levels. Again, note how the characteristics of the details change with the level. Here we also display the squared details from the finest (artificial) level corresponding to the same time point. In what follows, rather than try to model their distribution within each level, we will investigate their magnitude as a function of the associated integral lengths.

As a note, figure 6.15 shows for each time point and each artificial level, the squared average of the corresponding empirical wavelet coefficients. The squared sinusoidal behaviour is somewhat detected, and again we notice a diffusion of power between the new scales, although the (true) activity is located at level -2. This shows that there must be a re-distribution of the initial levels

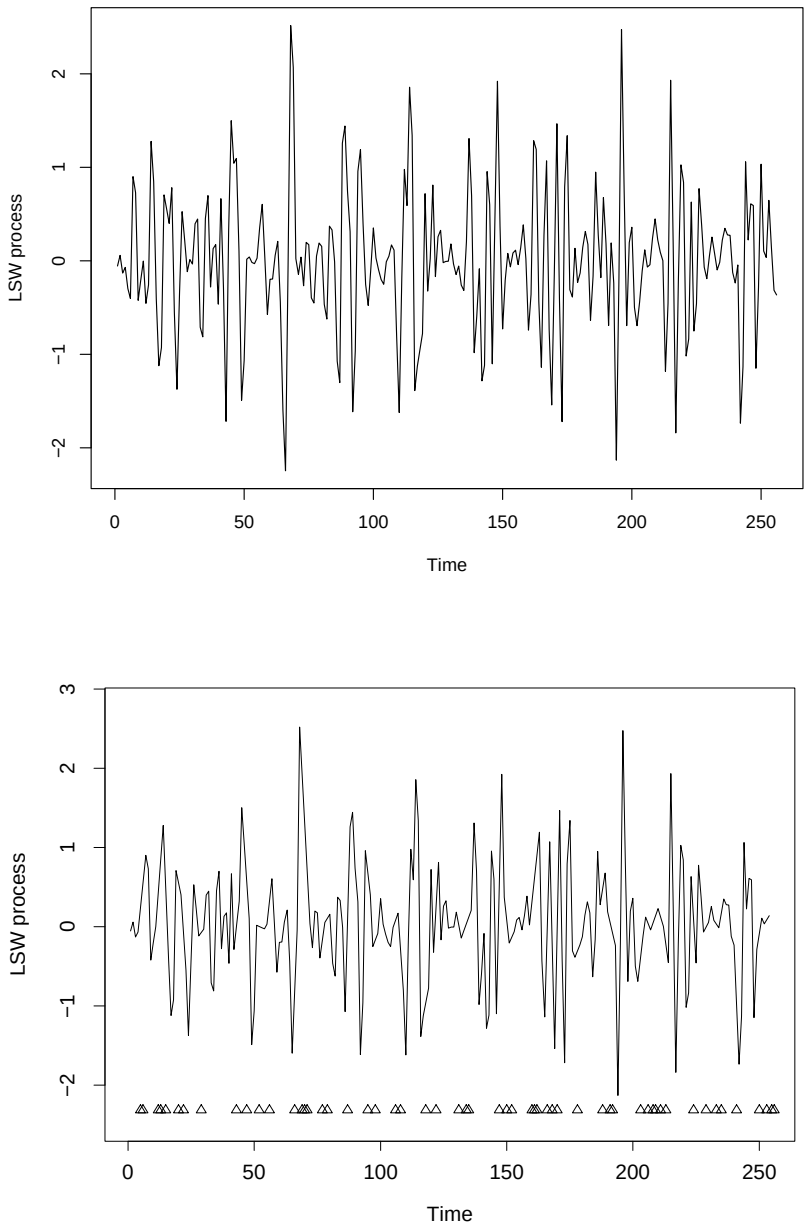

Figure 6.11: Top: simulated LSW process with spectrum as in figure 6.10. Bottom: the LSW process of above featuring missing observations; the locations of the missing observations are indicated by triangles.

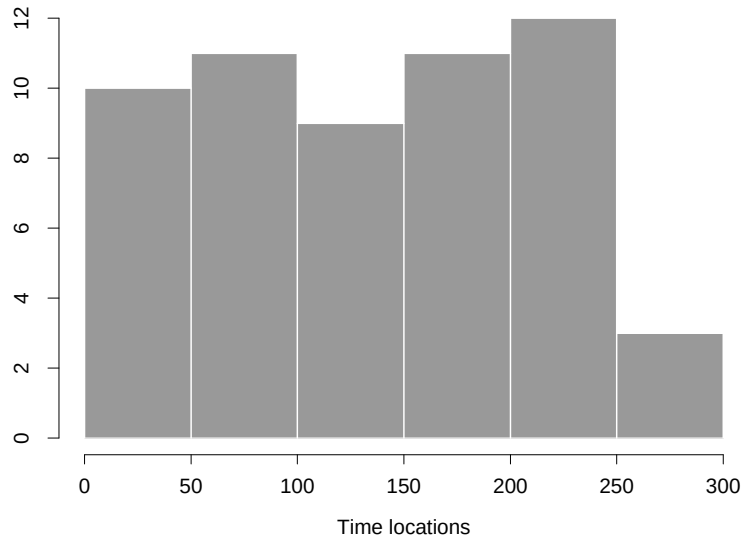

Figure 6.12: Distribution of the missing time points.

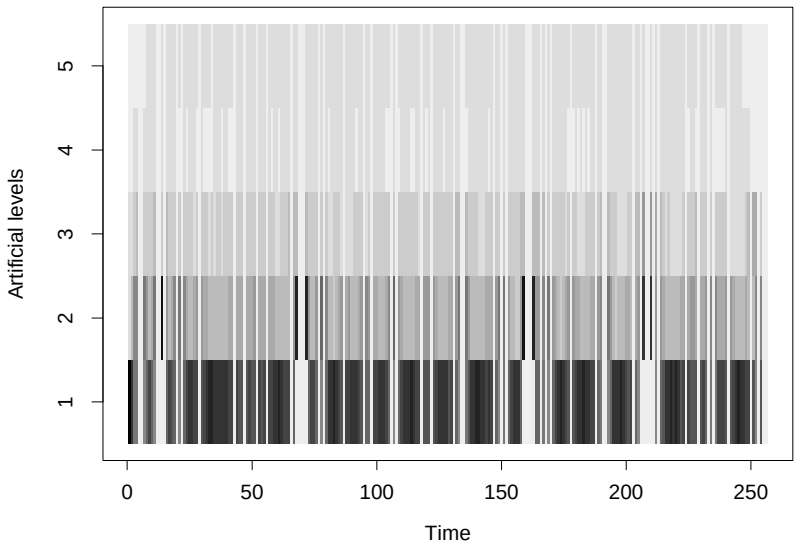

Figure 6.13: Number of times each time point is represented within each artificial level. The artificial levels are from finest (1, bottom) to coarsest (5, top). The intensity (darkness) of a pixel corresponds to a higher number of appearances. White lines extending over all artificial levels correspond to missing time points.

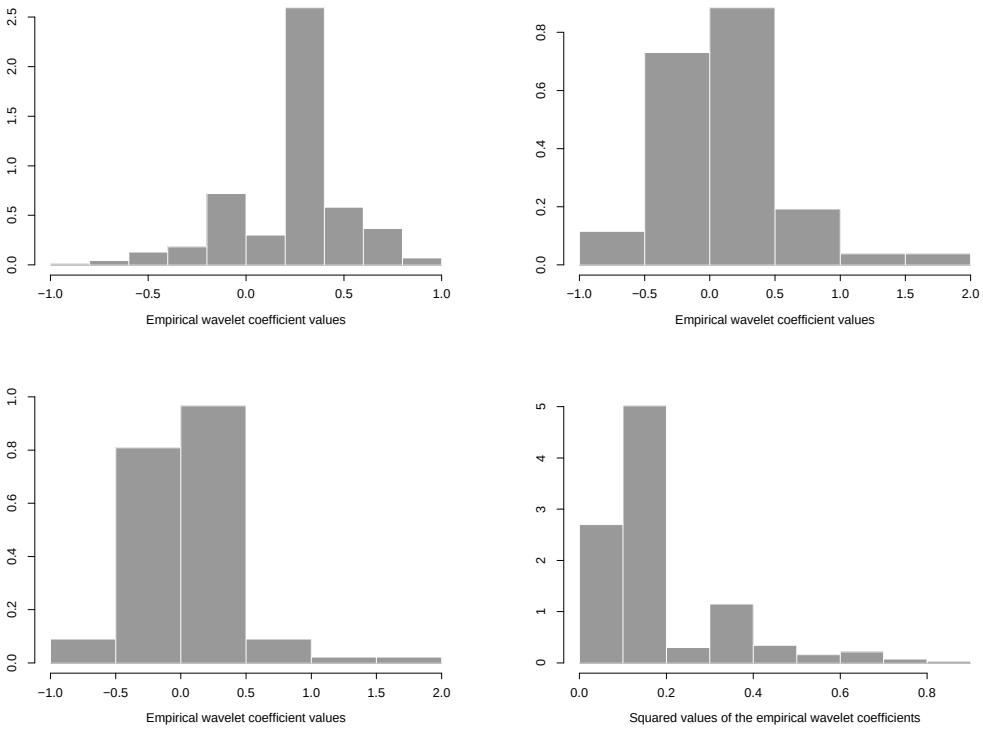

Figure 6.14: Top left, right, bottom left respectively: histograms of the details associated to observation at time 183 within artificial levels 1,4 respectively 5. Bottom right: histogram of the squared details in artificial level 1, associated to the same time point.

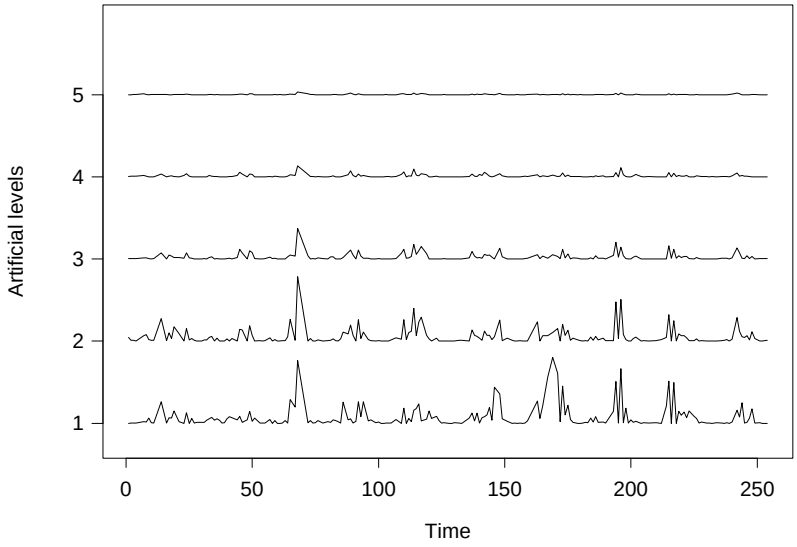

Figure 6.15: Squared average of the details at each sampled time point, within each artificial level (finest scale at the bottom, coarsest scale at the top).

into the new artificial ones. However, as we have already seen, we chose not explore this avenue any further for now.

As previously explained, we take a nonparametric regression approach to estimate the values of the squared details corresponding to each time point and to a set of scale values (see figure 6.16 for an example). Here we used scale values on a $\log_2$ range of the integral lengths and two levels of division (hence we generated two scales spanning approximately the range 0 to 8, one of them with a finer division).

The matrix that gives the estimated squared detail at each time location and 'evaluation' scale is our proposed periodogram, and it appears in figure 6.17. At each evaluation scale l^i, the intensity (darkness) of a pixel at time t_k is higher if it corresponds to a higher value $\hat{f}_{t_k,T}(l^i)$.

Note that the (10) peaks of activity at level -2 are detected in the correct region, although closer investigation is needed into the correspondence between our continuous scale and the initial discrete scale of Nason *et al.* (2000). Also, we have seen that our proposed periodograms are biased estimators of the true

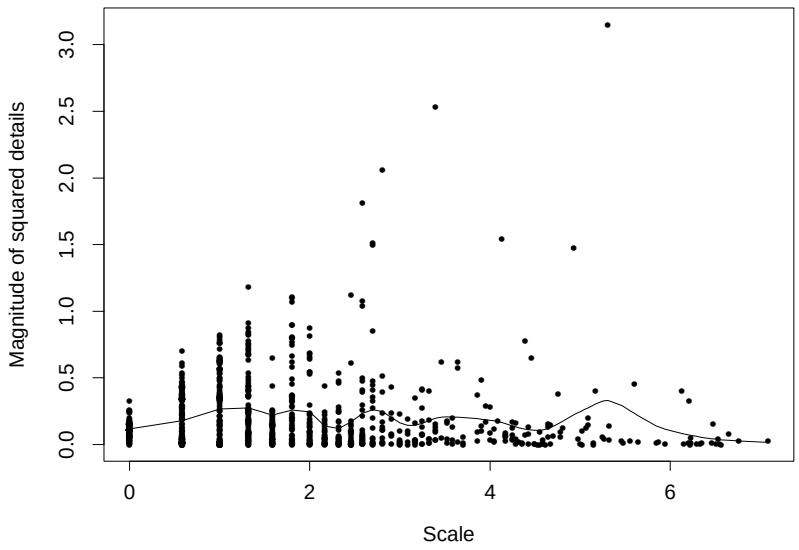

Figure 6.16: Magnitude of the squared details associated to the observation at time 183, versus ($\log_2$ of) their associated integral lengths. The superimposed curve is obtained by smoothing using a cubic spline.

evolutionary wavelet spectrum. In these circumstances, correction is expected to improve the periodograms in figure 6.17.

2. Let us now investigate a LSW process corresponding to a wavelet spectrum that at each of the four finest levels has a burst of activity spanning 64 time points, that does not overlap any of the other bursts (see equation (6.31) and figure 6.18).

$$S_j(z) = \begin{cases} 1, & \text{for } j = -1,\ z \in (0, \frac{1}{4}), \\ 1, & \text{for } j = -2,\ z \in (\frac{1}{4}, \frac{1}{2}), \\ 1, & \text{for } j = -3,\ z \in (\frac{1}{2}, \frac{3}{4}), \\ 1, & \text{for } j = -4,\ z \in (\frac{3}{4}, 1), \\ 0, & \text{otherwise.} \end{cases} \tag{6.31}$$

As we have done so far, we take a random sample of 200 time points out of the 256, and use this to simulate a LSW process with missing observations.

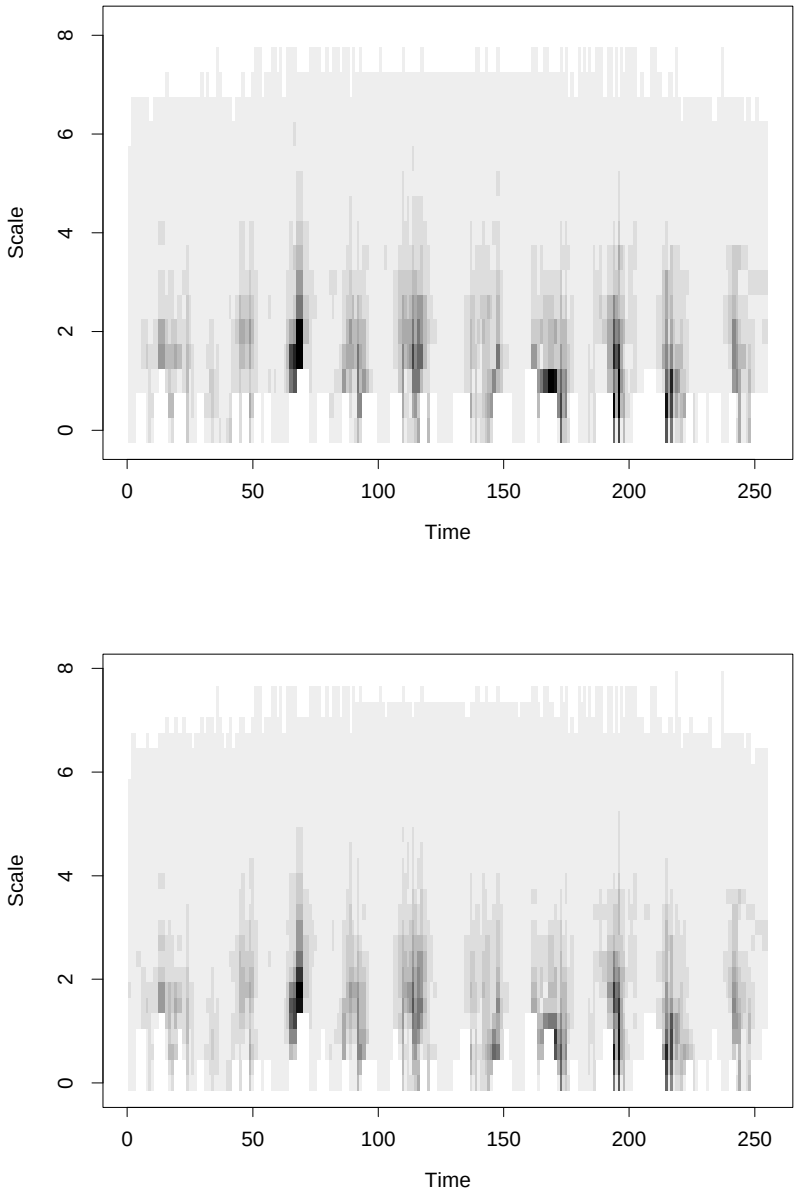

Figure 6.17: Proposed 'raw' wavelet periodograms to estimate the wavelet spectrum of figure 6.10. The smoothed squared details are represented on two different 'evaluation' scales (with 17, respectively 27 equally spaced divisions for the interval 0–8), with the bottom picture corresponding to the finer scale division. In each plot, the scale gets coarser from bottom upwards and darker pixels correspond to a higher estimated spectrum value.

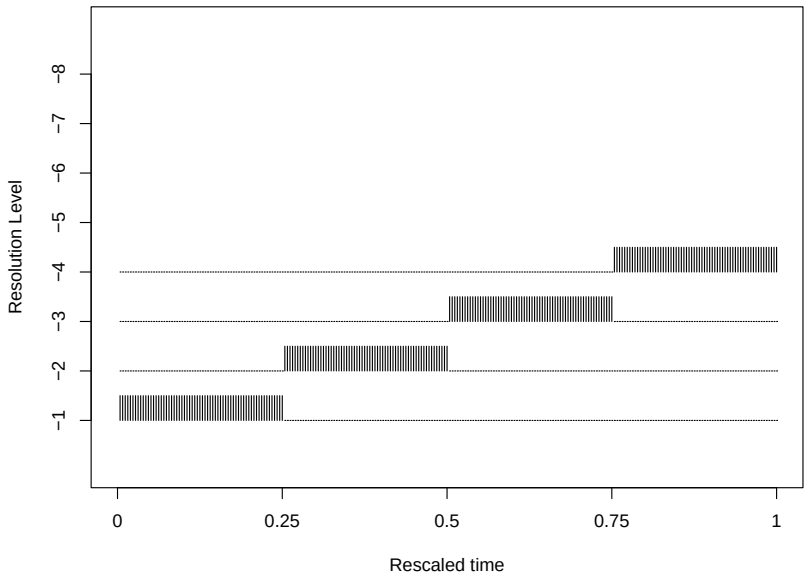

Figure 6.18: Evolutionary wavelet spectrum. The scale runs from finest (bottom) to coarsest (top).

A realization of such a LSW process, and its sampled version appear in figure 6.19.

We then use 1000 randomly selected trajectories and obtain for each one a set of empirical wavelet coefficients by applying the modified lifting scheme using prediction with linear regression with a neighbourhood of size 2 in symmetric configuration and with an intercept. The lifting algorithm generates 5 artificial levels and all time points are represented within each artificial level, except for time 147 which does not appear in the first artificial scale (see figure 6.20).

Each time point is now associated to a set of empirical wavelet coefficients, which in their turn correspond to a set of scales. If the artificial levels are used as a scale division, then the squared average of the details associated to each time point and each level appear in figure 6.21. We notice that while activity is identified at the finest four scales, it is not correctly localized within each scale. The same diffusion of power between levels that we have seen in the

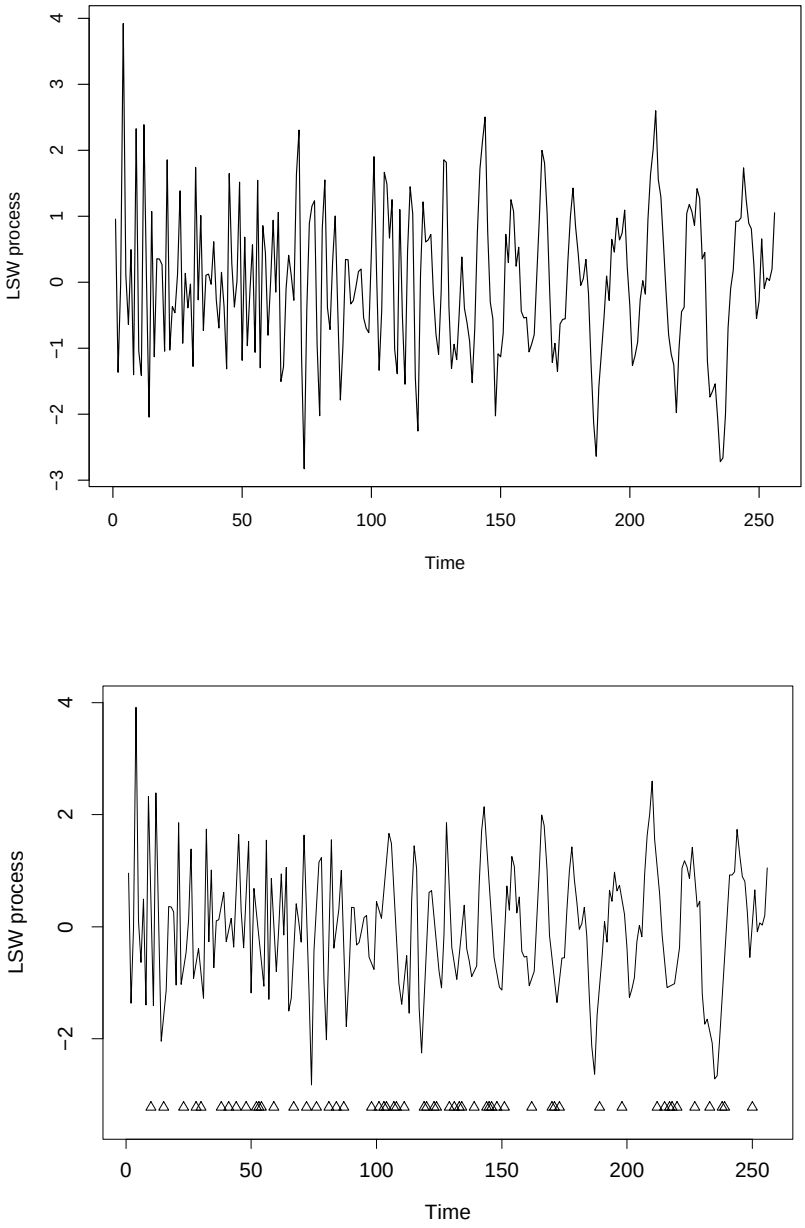

Figure 6.19: Top: simulated LSW process with spectrum as in figure 6.18. Bottom: the LSW process of above featuring missing observations; the locations of the missing observations are indicated by triangles.

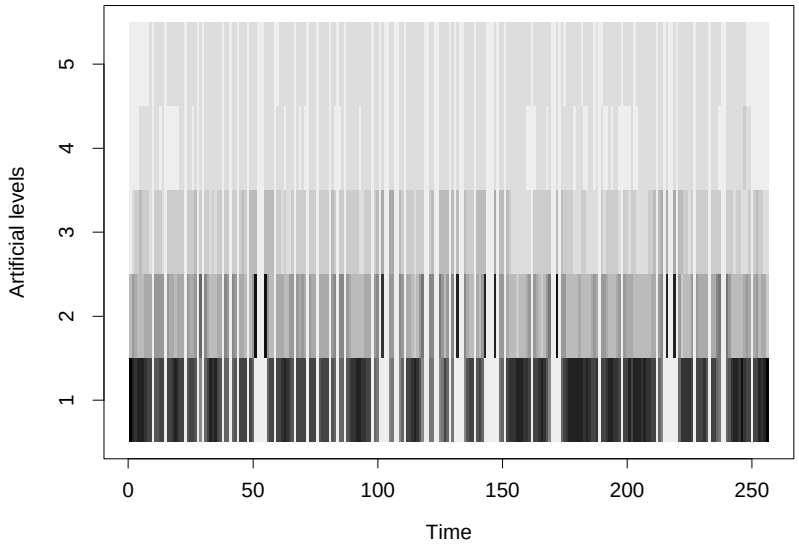

Figure 6.20: Number of times each time point is represented within each artificial level. The artificial levels are from finest (1, bottom) to coarsest (5, top). The intensity (darkness) of a pixel corresponds to a higher number of appearances. White lines extending over all artificial levels correspond to missing time points.

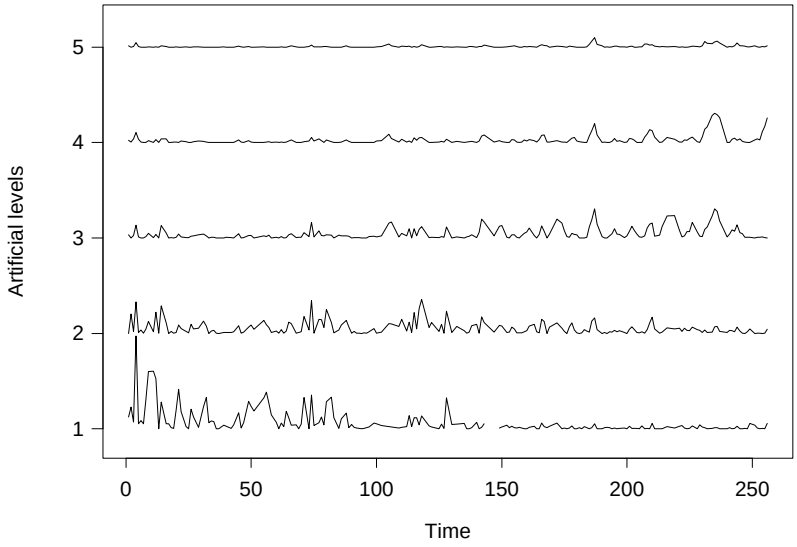

Figure 6.21: Squared average of the details at each sampled time point, within each artificial level (finest scale at the bottom, coarsest scale at the top).

previous examples appears here too, and one of the explanations is that the artificial scales provide a different scale division to the one used for defining the wavelet spectrum.

By estimating the magnitude of the squared detail that corresponds to each time point and (continuous) scale, we obtain the matrix that gives our periodogram, see figure 6.22. As before, we used two scale divisions.

Note that the activity is detected in the correct regions. However, due to the same power leak between scales that we noticed so far, there is not a clear 'cut' between levels. For the future, further investigation is needed into the correspondence between the proposed periodogram and the true evolutionary wavelet spectrum, both in terms of scale diffusion and correction for localisation (bias).

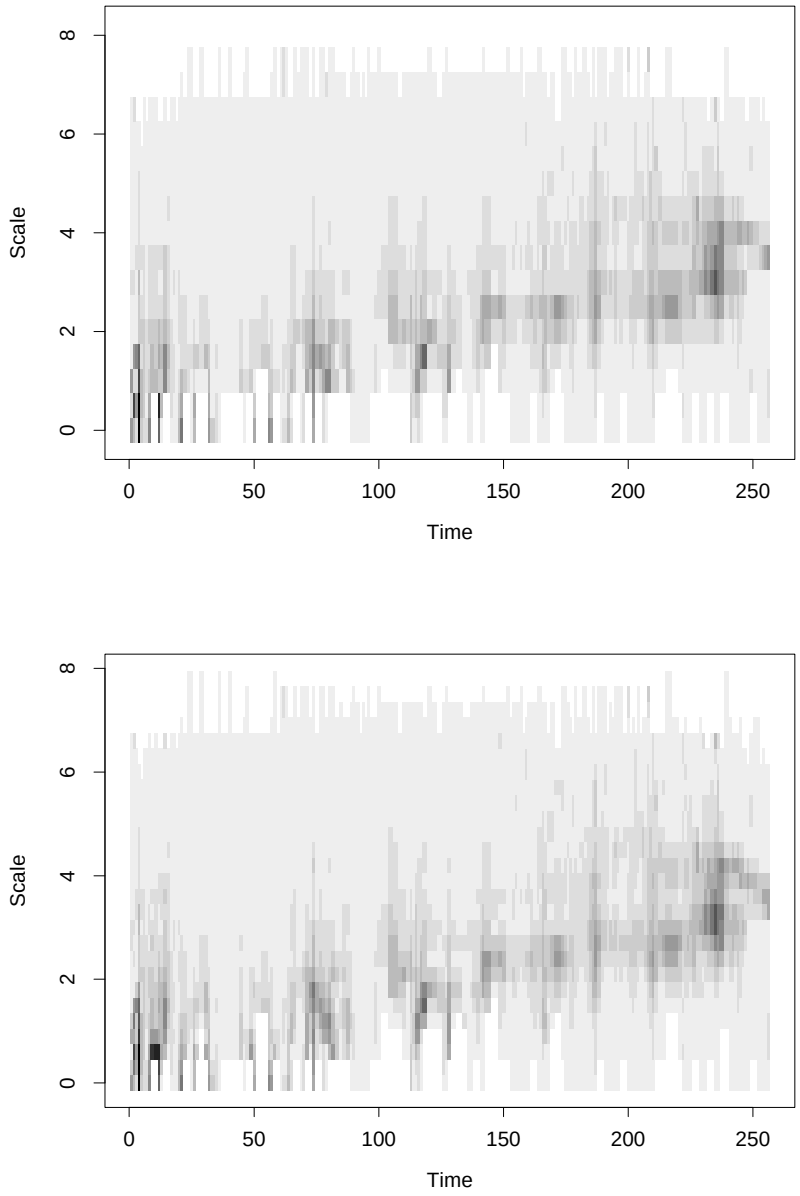

Figure 6.22: Proposed 'raw' wavelet periodograms to estimate the wavelet spectrum of figure 6.18. The smoothed squared details are represented on two different 'evaluation' scales (with 17, respectively 27 equally spaced divisions for the interval 0–8), with the bottom picture corresponding to the finer scale division. In each plot, the scale gets coarser from bottom upwards and darker pixels correspond to a higher estimated spectrum value.

6.5 Conclusions and further work

In this chapter we addressed the problem of spectral estimation for a non-stationary process that exhibits missing observations. Nonstationarity was understood here as local stationarity, and the wavelet model introduced by Nason *et al.* (2000) was adopted. In this context we proposed a 'nondecimated' lifting transform which ensured that a set of empirical wavelet coefficients is available at each (observed) time location throughout a continuous distribution of scales. Exploiting the flexibility behind the continuous nature of scale in second generation wavelet approaches, we proposed a 'raw' periodogram for estimating the wavelet spectrum at the (rescaled) observed locations. We have shown that the proposed periodogram is not an unbiased estimator for the evolutionary wavelet spectrum, and made a first step towards constructing a corrected periodogram. For the future, it would be interesting to work out the correction and the properties of the corresponding estimator, as well as its asymptotic behaviour. Also, to the moment we have not investigated the consequences of using adaptive lifting in our development, which would give our method the potential of not having to choose the wavelet basis. A challenge would be to set up a locally stationary wavelet type model, that would directly handle the problem of irregular data.

Chapter 7

Conclusions and further work

This chapter summarizes the main contributions of the novel work introduced in this thesis through chapters 3 to 6. Each section encapsules an idea and its development, together with its positive and negative aspects, as well as with directions for future research.

7.1 Adaptive second generation wavelets

Classical wavelet constructions are based on a few assumptions: the signal length is of the form 2^J, the observations are regularly spaced and there are no multiple observations at one location. For a successful application of wavelet shrinkage in nonparametric regression problems, the smoothness of the wavelet basis has to be suitably chosen for each particular dataset, and so should the primary resolution level.

Second generation wavelets are designed to work on sequences of any length, with observations at irregularly spaced locations. Stemming from the lifting scheme which removes one coefficient at a time of Jansen *et al.* (2001, 2004), we constructed second generation wavelet functions that adaptively adjust to the signal features (chapter 3). As a consequence, the smoothness of each wavelet function gets tuned adaptively and automatically to the varying smoothness of the data. Also, our approach allows for a natural way of handling multi-

ple data, hence the only remaining issue will be the equivalent of setting the primary resolution level.

We proposed two degrees of adaptiveness, which resulted in two adaptive lifting algorithms, which we call AdaptPred and AdaptNeigh. A detailed simulation study (chapter 4) showed that our adaptive transforms produce sparse wavelet representations and have competitive denoising properties for irregularly spaced datasets, when compared to both established wavelet and non-wavelet based regression techniques.

Therefore, our construction enables the work on general datasets, unrestricted by their size or by the existence of multiple observations at the same location or irregularities in locations. In nonparametric regression problems, when compared to classical wavelet methods, our approach also circumvents the issue of having to choose the wavelet smoothness.

However, the adaptivity in the transform makes the theoretical assessment of its statistical properties very difficult, as each application of the algorithm becomes dependent on the dataset it is used on. The work we have done so far is conditional on the local structure of the data. A future challenge would be to obtain unconditional results on the statistical properties of the empirical wavelet coefficients.

Second generation wavelet constructions based on the lifting scheme have been shown in the literature to generally display stability problems (see Simoens and Vandewalle (2003) for instance). Since the adaptiveness of our construction induces a dependence on the signal being analysed, assessing the stability of our transforms is more complicated. When done computationally, we must bear in mind that for each transform, the degree of irregularity in observations influences the result, but so does the test function, hence the results cannot easily be generalised. An interesting line for future research would be to investigate a possible transformation that would bring our bases closer to orthogonality, and hence to stability.

7.2 Transmembrane segment prediction along a protein sequence

Classical wavelet methods have been so far used in predicting hydrophobic segments along the sequence of a transmembrane protein (Lio and Vannucci (2000)). As with all classical wavelet constructions, certain assumptions are needed and consequently for this particular problem the residues along the protein chain are modelled as regularly spaced.

We challenged this assumption and constructed family-based dissimilarity matrices for estimating the distance between consecutive residues (chapter 5). In order to do this, we made use of the resolved 3D information available for proteins similar to the protein of interest.

Introducing irregularities in the residues called for wavelets capable to work on such data. Guided by the simulation results obtained in chapter 4, we proposed using two of our adaptive algorithms to predict transmembrane segments.

By comparing our method to the classical wavelet approach, we have shown (chapter 5) that by introducing irregular distances in the residues we improved prediction both in terms of the existence of predicted segments compared to experimentally determined ones, and also the proportion of correctly predicted segments.

A downside of our technique is that it requires more information (the resolved 3D structures of similar proteins) than the method involving classical wavelets (which makes use only of the length of protein sequence), and therefore our approach is more computationally expensive. Also, its real strength relies in the cases when there is available 3D information in similar proteins, which might not always be the case.

So far we modelled the protein sequence as a straight chain. An interesting direction for further research would be to model the 3D shape of the protein. This would also call for developing/generalizing an adaptive wavelet

construction able to work on 3D objects.

7.3 Spectral estimation for LSW processes with missing observations

Due to their capacity of delivering time–scale representations, first generation wavelets have been used for spectral estimation of both stationary and nonstationary time series (Chiann and Morettin (1999), Nason *et al.* (2000)). When the process realization features missing observations, which frequently happens in practice, classical wavelet methods are unable to work unless the regularity in time locations has been restored, for instance by somehow imputing the missing data.

We first proposed a 'nondecimated' lifting transform which ensures that a set of empirical wavelet coefficients is available at each observed time location, throughout a continuous distribution of scales (chapter 6). Based on this, we constructed a second generation wavelet periodogram for estimating the wavelet spectrum that corresponds to a class of nonstationary wavelet processes introduced by Nason *et al.* (2000) and assumed to feature missing observations (chapter 6).

Our approach is based upon second generation wavelets constructed using a modified variant of the linear form of the lifting scheme that removes 'one coefficient at a time' of Jansen *et al.* (2001, 2004). The modified lifting scheme we propose accommodates a random order of generating the wavelet coefficients. The random order is referred to as path or trajectory. Conditional on the observed time points being fixed and not taking into consideration the randomness of the trajectories, the proposed second generation wavelet periodogram is shown to be biased, and therefore it needs a correction. Computationally, the raw periodogram visually performs well in estimating the evolutionary wavelet spectrum, although more research is needed into establishing a more exact

correspondence between the two quantities.

For future research it would be very interesting to specifically work out the correction needed to produce an unbiased estimator of the spectrum, and the statistical properties of the corrected estimator. Also, so far we have not investigated the consequences of using our adaptive lifting transforms on the properties of our raw periodogram. An asymptotic development for the lifting construction in spectral analysis is another challenging direction arising from this work.

Bibliography

Abramovich, F., Bailey, T.C. and Sapatinas, T. (2000) Wavelet analysis and its statistical applications. *J. Roy. Statist. Soc.* D, **49**, 1–29.

Abramovich, F. and Benjamini, Y. (1995) Thresholding of wavelet coefficients as multiple hypotheses testing procedure. In *Wavelets and Statistics*, Antoniadis, A. and Oppenheim, G. (eds), Lect. Notes Statist., Springer-Verlag, New York, **103**, 5–14.

Abramovich, F., Sapatinas, T. and Silverman, B.W. (1998) Wavelet thresholding via a Bayesian Approach. *J. Roy. Statist. Soc.* B, **60**, 725–749.

Altschul, S.F., Gish, W., Miller, W., Myers, E.W. and Lipman, D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Antoniadis, A. and Fan, J. (2001) Regularization of wavelet approximations. *J. Amer. Statist. Assoc.*, **96**, 939–967.

Barber, S. and Nason, G.P. (2004) Real nonparametric regression using complex wavelets. *J. Roy. Statist. Soc.* B, **66**, 927–939.

Bos, R., de Waele, S. and Broersen, P.M.T. (2002) Autoregressive spectral estimation by application of the Burg algorithm to irregularly sampled data. *IEEE Trans. Instrum. Meas.*, **51**, 1289–1294.

Boulgouris, N.V., Tzovaras, D. and Strintzis, M.G. (2001) Lossless image compression based on optimal prediction, adaptive lifting and conditional arithmetic coding. *IEEE Trans. Im. Proc.*, **10**, 1–14.

Brockwell, P.J. and Davis, R.A. (1991) *Time Series: Theory and Methods.* Springer-Verlag, New York.

Cai, T.T. and Brown, L.D. (1998) Wavelet shrinkage for nonequispaced samples. *Ann. Statist.*, **26**, 1783–1799.

Cai, T.T. and Brown, L.D. (1999) Wavelet estimation for samples with random uniform design. *Statist. Prob. Lett.*, **42**, 313–321.

Chatfield, C. (2004) *The Analysis of Time Series.* Chapman & Hall/CRC.

Chiann, C. and Morettin, P.A. (1999) A wavelet analysis for time series. *J. Nonparam. Statist.*, **10**, 1–46.

Chipman, H.A., Kolaczyk, E.D. and McCulloch, R.E. (1997) Adaptive Bayesian wavelet shrinkage. *J. Amer. Statist. Assoc.*, **92**, 1413–1421.

Claypoole, R.L., Baraniuk, R.G. and Nowak, R.D. (1998) Adaptive wavelet transforms via lifting. In *Transactions of the International Conference on Acoustics, Speech and Signal Processing*, Seattle.

Claypoole, R.L., Davis, G.M., Sweldens, W. and Baraniuk, R.G. (2003) Nonlinear wavelet transforms for image coding via lifting. *IEEE Trans. Im. Proc.*, **12**, 1449–1459.

Clinger, W. and Van Ness, J.W. (1976) On unequally spaced time points in time series. *Ann. Statist.*, **4**, 736–745.

Cohen, A., Daubechies, I. and Feauveau, J. (1992) Bi-orthogonal bases of compactly supported wavelets. *Commun. Pure Appl. Math.*, **45**, 485–560.

Cohen, A., Daubechies, I. and Vial, P. (1993) Wavelets on the interval and fast wavelet transforms. *Appl. Comput. Harmon. Anal.*, **1**, 54–81.

Coifman, R.R., Meyer, Y., Quake, S. and Wickerhauser, M.V. (1989) Signal processing and compression with wave packets. In *Proceedings of the International Conference on Wavelets*, Meyer, Y. (ed), Paris:Masson.

Coifman, R.R. and Wickerhauser, M.V. (1992) Entropy-based algorithms for best basis selection. *IEEE Trans. Inform. Theory*, **38**, 713–718.

Comte, F. and Rozenholc, Y. (2004) A new algorithm for fixed design regression and denoising. *Ann. Inst. Statist. Math.*, **56**, 449–473.

Dahlhaus, R. (1997) Fitting time series models to nonstationary processes. *Ann. Statist.*, **25**, 1–37.

Daubechies, I. (1988) Orthonormal bases of compactly supported wavelets. *Commun. Pure Appl. Math.*, **41**, 909–996.

Daubechies, I. (1992) *Ten Lectures on Wavelets*. Philadelphia: SIAM.

Delouille, V., Franke, J. and von Sachs, R. (2001) Nonparametric stochastic regression with design-adapted wavelets. *Sankhya*, A, **63**, 328–366.

Delouille, V., Simoens, J. and von Sachs, R. (2004) Smooth design-adapted wavelets for nonparametric stochastic regression. *J. Amer. Statist. Assoc.*, **99**, 643–658.

Donoho, D.L. and Johnstone, I.M. (1994) Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, **81**, 425–455.

Donoho, D.L. and Johnstone, I.M. (1995a) Adapting to unknown smoothness via wavelet shrinkage. *J. Amer. Statist. Assoc.*, **90**, 1200–1224.

Donoho, D.L., Johnstone, I.M., Kerkyacharian, G. and Picard, D. (1995b) Wavelet shrinkage: asymptopia? (with discussion) *J. Roy. Statist. Soc.* B, **57**, 301–369.

Eisenberg, D., Schwarz, E., Komaromy, M. and Wall, R. (1984) Analysis of membrane and surface protein sequences with the hydrophobic moment plot. *J. Mol. Biol.*, **179**, 125–142.

Engelman, D.M., Steitz, T.A. and Goldman, A. (1986) Identifying nonpolar transbilayer helices in amino acid sequences of membrane proteins. *Annu. Rev. Biophys. Biophysical Chem.*, **15**, 321–353.

Engle, R.F. (2000) The econometrics of ultra-high-freqency data. *Econometrica*, **68**, 1–22.

Fischer, P., Baudoux, G. and Wouters, J. (2003) WAVPRED: a wavelet-based algorithm for the prediction of transmembrane proteins. *Comm. Math. Sci.* **1**, 44–56.

Friedman, J.H. (1984) A variable span scatterplot smoother. *Technical Report*, No. 5, Laboratory for Computational Statistics, Stanford University, Stanford, CA, USA.

Fryzlewicz, P.Z. (2003) Wavelet Techniques for Time Series and Poisson Data. *PhD Thesis*, University of Bristol.

Green, P.J. and Silverman, B.W. (1994) *Nonparametric Regression and Generalized Linear Models.* Chapman and Hall: London.

Hall, P., Fisher, N.I. and Hoffmann, B. (1994) On the nonparametric estimation of covariance functions. *Ann. Statist.*, **22**, 2115–2134.

Hall, P. and Nason, G.P. (1997) On choosing a non-integer resolution level when using wavelet methods. *Statist. Prob. Lett.*, **34**, 5–11.

Henikoff, S. and Henikoff, J.G. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA*, **89**, 10915–10919.

Henikoff, S. and Henikoff, J.G. (1993) Performance evaluation of amino acid substitution matrices. *Proteins*, **17**, 49–61.

Hirakawa, H., Muta, S. and Kuhara, S. (1999) The hydrophobic cores of proteins predicted by wavelet analysis. *Bioinformatics*, **15**, 141–148.

Jansen, M., Nason, G.P. and Silverman, B.W. (2001) Scattered data smoothing by empirical Bayesian shrinkage of second generation wavelet coefficients. In *Proceedings of SPIE, Wavelet applications in signal and image processing*, Unser, M. and Aldroubi, A. (eds), **4478**, 87–97.

Jansen, M., Nason, G.P. and Silverman, B.W. (2004) Multivariate nonparametric regression using lifting. *Technical Report* 04:17, Statistics Group, Department of Mathematics, University of Bristol, UK.

Jawerth, B. and Sweldens, W. (1994) An overview of wavelet based multiresolution analyses. *SIAM Review*, **36**, 377–412.

Johnstone, I.M. and Silverman, B.W. (1997) Wavelet threshold estimators for data with correlated noise. *J. Roy. Statist. Soc.* B, **59**, 319–351.

Johnstone, I.M. and Silverman, B.W. (2004a) Needles and straw in haystacks: Empirical Bayes estimates of possibly sparse sequences. *Ann. Statist.*, **32**, 1594–1649.

Johnstone, I.M. and Silverman, B.W. (2004b) EbayesThresh: R programs for Empirical Bayes thresholding. *J. Statist. Soft.*, **12**, 1–38.

Johnstone, I.M. and Silverman, B.W. (2005) Empirical Bayes selection of wavelet thresholds. *Ann. Statist.*, **33**, 1700–1752.

Jones, R.H. (1962) Spectral analysis with regularly missed observations. *Ann. Math. Statist.*, **33**, 455–461.

Knight, M.I. and Nason, G.P. (2004) Improving prediction of hydrophobic segments along a transmembrane protein sequence using adaptive multiscale lifting. *Technical Report* 04:19, Statistics Group, Department of Mathematics, University of Bristol, UK. To appear in *SIAM Multisc. Model. & Sim.*

Kovac, A. (1998) Wavelet Thresholding for Unequally Spaced Data. *PhD Thesis*, University of Bristol.

Kovac, A. and Silverman, B.W. (2000) Extending the scope of wavelet regression methods by coefficient-dependent thresholding. *J. Amer. Statist. Assoc.*, **95**, 172–183.

Kyte, J. and Doolittle, R.F. (1982) A simple method for displaying the hydropathic character of a protein. *J. Mol. Biol.*, **157**, 105–132.

Lio, P. and Vannucci, M. (2000) Wavelet change-point prediction of transmembrane proteins. *Bioinformatics*, **16**, 376–382.

Lipman, D.J. and Pearson, W.R. (1990) Rapid and sensitive protein similarity searches. *Science*, **227**, 1435–1441.

Loader, C. (1997) Locfit: an introduction. *Statist. Comput. Graph. News.*, **8**, 11-17.

Loader, C. (1999) *Local Regression and Likelihood.* Springer: New York.

Mallat, S.G. (1989a) Multiresolution approximations and wavelet orthonormal bases of $L^2(\mathbb{R})$. *Trans. Amer. Math. Soc.*, **315**, 69–87.

Mallat, S.G. (1989b) A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattn. Anal. Mach. Intell.*, **11**, 674–693.

Mallat, S.G. (1999) *A Wavelet Tour of Signal Processing.* Academic Press.

Meyer, Y. (1992) *Wavelets and Operators.* Cambridge University Press: Cambridge.

Mikosch, T. and Starica, C. (2004) Nonstationarities in financial time series, the long–range dependence, and the IGARCH effects. *The Rev. Econ. Statist.*, **86**, 378–390.

Nason, G.P. (1996) Wavelet shrinkage using cross-validation. *J. Roy. Statist. Soc.* B, **58**, 463–479.

Nason, G.P. (1998) WaveThresh Software. Department of Mathematics, University of Bristol, UK.

Nason, G.P. (2002) Choice of wavelet smoothness, primary resolution and threshold in wavelet shrinkage. *Statist. Comput.*, **12**, 219–227.

Nason, G.P., Sapatinas, T. and Sawczenko, A. (2001) Wavelet packet modelling of infant sleep state using heart rate data. *Sankhya* B, **63**, 199–217.

Nason, G.P. and Silverman, B.W. (1994) The discrete wavelet transform in S. *J. Comput. Graph. Statist.*, **3**, 163–191.

Nason, G.P. and von Sachs, R. (1999) Wavelets in time series analysis. *Phil. Trans. Roy. Soc. London* A, **357**, 2511–2526.

Nason, G.P., von Sachs, R. and Kroisandt, G. (2000) Wavelet processes and adaptive estimation of the evolutionary wavelet spectrum. *J. Roy. Statist. Soc.* B, **62**, 271–292.

Nunes, M.A. and Nason, G.P. (2005) Stopping time in adaptive lifting. *Technical Report* 05:15, Statistics Group, Department of Mathematics, University of Bristol, UK.

Nunes, M.A., Knight, M.I. and Nason, G.P. (2004) Adaptive lifting in nonparametric regression. *Technical Report* 04:20, Statistics Group, Department of Mathematics, University of Bristol, UK. To appear in *Statist. Comput.*.

Ogden, R.T. and Parzen, E. (1996a) Change-point approach to data analytic wavelet thresholding. *Statist. Comput.*, **6**, 93–99.

Ogden, R.T. and Parzen, E. (1996b) Data dependent wavelet thresholding in nonparametric regression with change-point applications. *Comput. Statist. Data Anal.*, **22**, 53–70.

Pensky, M. and Vidakovic, B. (2001) On non-equally spaced wavelet regression. *Ann. Inst. Statist. Math.*, **53**, 681–690.

Percival, D.B. and Walden, A.T. (2000) *Wavelet Methods for Time Series Analysis.* Cambridge University Press, Cambridge.

Piella, G. and Heijmans, H.J.A.M. (2002) Adaptive lifting schemes with perfect reconstruction. *IEEE Trans. Sig. Proc.*, **50**, 1620–1630.

Priestley, M.B. (1965) Evolutionary spectra and non-stationary processes. *J. Roy. Statist. Soc.* B, **27**, 204–237.

Priestley, M.B. (1981) *Spectral Analysis and Time Series.* Academic Press.

Ramsay, J.O. and Silverman, B.W. (1997) *Functional Data Analysis.* Springer-Verlag, New York.

Rost, B., Casadio, R., Fariselli, P. and Sander, C. (1995) Transmembrane helices predicted at 95% accuracy. *Protein Science* **4**, 521–533.

Rost, B., Fariselli, P. and Casadio, R. (1996) Topology prediction for helical transmembrane proteins at 86% accuracy. *Protein Science* **5**, 1704–1718.

Rost, B. and Sander, C. (1993) Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol.*, **232**, 584–599.

Rudin, W. (1973) *Functional Analysis.* McGraw, New York.

Sardy, S., Percival, D.B., Bruce, A.G., Gao, H.-Y. and Stuetzle, W. (1999) Wavelet shrinkage for unequally spaced data. *Statist. Comput.*, **9**, 65–75.

Silverman, B.W. (1985) Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *J. Roy. Statist. Soc.* B, **47**, 1–52.

Simoens, J. and Vandewalle, S. (2003) A stabilized construction of wavelets on irregular meshes on the interval. *SIAM J. Scient. Comput.*, **24**, 1356–1378.

Simonoff, J.S. (1996) *Smoothing Methods in Statistics.* Springer, New York.

Stein, C.M. (1981) Estimation of the mean of a multivariate normal distribution. *Ann. Statist.*, **9**, 1135–1151.

Stryer, L. (1988) *Biochemistry.* W. H. Freeman and Company, New York.

Sweldens, W. (1996) Wavelets and the lifting scheme: A 5 minute tour. *Z. Angew. Math. Mech.*, **76**, 41–44.

Sweldens, W. (1998) The lifting scheme: a construction of second generation wavelets. *SIAM J. Math. Anal.*, **29**, 511–546.

Trappe, W. and Liu, K.J.R. (2000) Denoising via adaptive lifting schemes. In *Proceedings of SPIE, Wavelet applications in signal and image processing VIII*, Aldroubi, A., Laine, M.A. and Unser, M.A. (eds), **4119**, 302–312.

Vanraes, E., Jansen, M. and Bultheel, A. (2002) Stabilised wavelet transforms for non-equispaced data smoothing. *Sig. Proc.*, **82**, 1979–1990.

Vidakovic, B. (1999) *Statistical Modeling by Wavelets.* Wiley: New York.

Watson, G.S. (1964) Smooth regression analysis. *Sankhya* A, **26**, 359–372.

Zemla, A., Venclovas, C., Fidelis, K. and Rost, B. (1999) A modified definition of *Sov*, a segment based measure for protein secondary structure assesment. *Proteins: Struct., Funct. and Genetics* **34**, 220–223.