Local Feature Extraction and Its Applications Using a Library of Bases

A Dissertation

Presented to the Faculty of the Graduate School

of

Yale University

in Candidacy for the Degree of

Doctor of Philosophy

by

Naoki Saito

Dissertation Director: Ronald R. Coifman

December 1994

To the Saito family

and to the memory of my mother Teruko

Contents

List of Figures

List of Tables

Acknowledgments

First of all, I would like to thank my advisor, Professor Ronald Raphy Coifman, for his support, encouragement, useful suggestions, and enthusiastic discussions. I was very fortunate to have an opportunity to work with him.

Many thanks are also due to Professor Gregory Beylkin of the University of Colorado at Boulder for introducing me to the world of wavelets and suggesting that I should pursue my Ph.D. at Yale under the supervision of Professor Coifman. Chapter 7 is a result of our continuing collaboration.

I would like to thank Professor Yves Meyer of the Université de Paris-Dauphine, and Professor Vladimir Rokhlin of the Computer Science Department at Yale for serving as the reading committee members of my thesis. I have been quite influenced by Professor Meyer's books.

I am also indebted to the work of Professor David Donoho of Stanford University and Professor Victor Wickerhauser of Washington University at St. Louis.

I would like to thank Professor Andrew Barron of the Statistics Department at Yale for helpful discussions. I also learned a lot from his course on information theory.

Many of my colleagues at Schlumberger-Doll Research gave me helpful suggestions and encouragement; in particular, I appreciate valuable inputs from: Drs. Robert Burridge, Stefan Luthi, Douglas Miller, Ram Shenoy, and Lisa Stewart. Schlumberger's management team has been supportive throughout my study at Yale. I would like to thank especially

Dr. Bill Murphy and Mr. Luis Ayestaran, the former and the current directors of the Interpretation Science Department.

On the personal side, I would like to thank Toshiki Saito, my father, and Shigeko Yamaki, my grandmother, for making this all possible through their constant encouragement and support for years. I was deeply affected by my mother Teruko who passed away when I was 11 years old. She was the first to show me how interesting mathematics is.

My parents-in-law, Masuto and Yoko Yashiki, and my aunt-in-law, Hiroko Katayama always encouraged me and supported my study.

My sons, Tomoya (now five years old) and Yuta (one year old) refreshed my mind many times.

Last but not least, I would like to give my special thanks to my wife, Mayumi. Without her, I could not have completed this thesis at all.

This research was supported in part by Schlumberger-Doll Research and by APRA ATR program.

NAOKI SAITO

Yale University

September 1994

Chapter 1

Introduction

1.1 Importance of Feature Extraction

In analyzing and interpreting signals such as musical recordings, seismic signals, or stock market fluctuations, or images such as mammograms or satellite images, extracting relevant features from them is of vital importance. Often, the important features for signal analysis, such as edges, spikes, or transients, are characterized by local information either in the time (or space) domain or in the frequency (or wave number) domain or in both[1]: for example, to discriminate seismic signals caused by nuclear explosions from the ones caused by natural earthquakes, the frequency characteristics of the primary waves, which arrive in a short and specific time window, may be a key factor; to distinguish benign and malignant tissues in mammograms, the sharpness of the edges of masses may be of critical importance.

In the present thesis, we explore how to extract relevant features from signals and discard irrelevant information for a variety of problems in signal analysis. We address five aspects of signal analysis:

Compression: how to represent and describe signals in a compact manner for information

[1]From now on, unless mentioned otherwise, time and frequency also means space and wave number (spatial frequency) respectively.

transmission and storage.

Noise removal: how to remove random noise or undesired components from signals (also called *denoising*).

Classification: how to classify signals into known categories.

Regression: how to predict a response of interest from input signals.

Edge Characterization: how to detect singularities (e.g., spikes, step edges, or "ramp" edges) in signals and characterize them.

Although the methods we develop here can be applied to many different types of signals and images, we focus our attention on those measured by sensing devices and representing certain properties of natural objects such as acoustic properties of subsurface geological formations. Also, the signals and images treated in this thesis (whether synthetic or real) are all *discrete*: they are simply vectors and matrices consisting of a finite number of real-valued samples. These signals normally have very large number of samples; e.g., a typical exploration seismic record per receiver has 1000 samples, and a typical CT scanner image has 512×512, or $262,144$ samples. Therefore, extracting only important features for the problem at hand and discarding irrelevant information (this strategy is called the *reduction of dimensionality*) are crucial; if one succeeds in doing this, the subsequent objectives can be improved both in accuracy and computational efficiency.

1.2 Historical Background on Feature Extraction

The problem of feature extraction in general has intrigued many scientists, and several fundamental ideas have been proposed. The philosophically most important and notable among them are as follows: Kolmogorov contended that the best description of data is defined as the length of the shortest computer program to generate that data (this length

is called the *Kolmogorov complexity*) [82] (see also [90]). This concept was further refined by Rissanen and lead to the *minimum description length principle* [128], which states that the best theory to infer from data is the one minimizing the sum of the length of the theory and the length of the data using the theory both encoded as binary digits. Watanabe paraphrased that "pattern recognition is a quest for the minimum entropy[2], when suitably defined" [151], [152]. Grenander proposed the *pattern theory* for understanding regular structures of patterns based on the "analysis by synthesis" approach, i.e., analyzing patterns by decomposing into and synthesizing from the elementary building blocks [64] (see also the work of Mumford [108]). All of these proposals essentially share the same theme: an efficient and compact representation or description of data suitable for the problem at hand leads to an improvement in solving the problem itself. Although these proposals have influenced many others including the author of this thesis, their numerical implementations are not always straightforward and efficient. This thesis attempts to answer partially how to implement these philosophical proposals via concrete and efficient numerical algorithms for a specific form of data, i.e., discrete signals and images.

If we look back at practical methodologies proposed for feature extraction in signal analysis, they may be grouped into two general approaches: one is the *statistical* (or *decision-theoretic*) approach (representative references are [63], [48], [103]); the other is the *structural* (or *syntactic*) approach ([62], [114]). The major accomplishments in the statistical approach—some of them are reviewed in the subsequent chapters—are the development of the Fast Fourier transform (FFT) [37] for signal representation and noise removal, the Karhunen-Loève transform (KLT) [also known as Principal Component Analysis (PCA)] for signal compression [74], [80], [91], [150], the Linear Discriminant Analysis (LDA) [59] for classification, and more generally, the Projection Pursuit (PP) [86], [61], [77], Classification and Regression Trees (CART$^{\text{TM}}$) [18], or Artificial Neural Networks (ANNs) [107],

[2]Entropy is essentially a measure of the disorder in a system. We shall define entropy more precisely in the following chapters.

[125], [24]. Each of them can be considered as a tool for providing an efficient coordinate system (whether orthogonal or not) for the specific problems. However, when we use them to extract features, especially the features whose energy is localized simultaneously in the time and frequency domains, we immediately face one or both of the following problems: computational complexity and inability to capture localized features. KLT and LDA require solving the eigenvalue systems which is an expensive operation, i.e., $O(n^3)$, where n is a number of samples in each signal. In the case of PP and ANNs, optimization of certain nonlinear functions of high dimensional variable is required; in general, they are computationally expensive, and moreover, easily get stuck in local minima of these nonlinear objective functions. CART requires searching and sorting all coordinates for the best partition of the input signal space. The second difficulty, the lack of ability to capture local features, implies that the interpretation of the features extracted by these methods becomes difficult. For example, with FFT, one can analyze signals locally in the frequency domain, but cannot obtain local information in the time domain at all. Since KLT and LDA rely on the eigenvalue systems, they are fragile to outliers and perturbations and only extract global features in either the time or frequency domain. ANNs have difficulty in interpreting the physical meaning of the weights of the connections among neurons (computational units), and so on. In short, these methods are overwhelmed by the direct input of raw signals of large dimensions; they are not originally designed as feature extractors even though they have been used as such. They are rather designed to optimize certain criteria for the specific tasks such as compression, denoising, classification, and regression for *given* features. Therefore, they can be very powerful tools if a small number of key features are supplied to them.

On the other hand, the structural approach [62], [114], is based on the philosophy of "analysis by synthesis." This is similar to the pattern theory of Grenander, which describes signals in terms of a hierarchical composition of predefined primitive components or

elementary building blocks (e.g., peaks/valleys or energy levels of certain frequency bands of signals etc.); see [62] for various examples. In particular, this approach associates the description of signals with a formal language theory: it makes the correspondence of: 1) the elementary features of the signals, 2) the signals themselves, and 3) the structural description of the signals or class of signals (normally tree structures) to: 1) the words, 2) the sentences, and 3) the syntax or grammar of languages. The fundamental problem of this approach is how to define the elementary building blocks in the first place. It does not solve the feature extraction problem either.

Considering this situation, it is worthwhile to pursue a unified approach to feature extraction to fully utilize these available statistical tools and to permit the intuitive interpretation of the results, as in the structural approach. This thesis offers a new approach by blending the statistical and structural approaches.

1.3 The Best-Basis Paradigm and a Library of Bases

The approach to the feature extraction explored in this thesis is guided by the so-called *best-basis paradigm* [35], [105]. This paradigm consists of three main steps:

1. select a "best" basis (or coordinate system) for the problem at hand from a *library of bases* (a fixed yet flexible set of bases consisting of wavelets and their relatives: wavelet packets, local trigonometric bases, and the autocorrelation functions of wavelets),

2. sort the coordinates (features) by "importance" for the problem at hand and discard "unimportant" coordinates, and

3. use the surviving coordinates to solve the problem at hand.

What is "best" and "important" clearly depends on the problem. For signal compression, a basis which provides only a few large components in the coordinate vectors should

be used since we can then discard the other components without much signal degradation. Thus, to measure the efficiency of the coordinate system for compression, an information cost such as *entropy* may be appropriate since entropy measures the number of significant coordinates in a vector. For classification, a basis through which we can "view" classes as maximally-separated point clouds in the n-dimensional space, where n is the number of samples in each signal, is a choice. In this case, the class separability index or "distances" among classes (such as relative entropy) should be used as a measure of the efficiency of the coordinate system. For regression, a basis through which we can "see" the essential relationships between the input signals and output responses of interest should be used. For this purpose, prediction error such as relative ℓ^2 error may measure an efficiency of the coordinate system. For edge characterization problems, a coordinate system which allows one to detect, characterize, and manipulate edges in a convenient manner should be used. The original best-basis paradigm was developed mainly for signal compression problems [35]. In the present thesis, this paradigm is extended to noise removal, classification, and regression problems using these basis selection criteria.

One may ask why we use the library approach. There are several answers to this question. If we confine ourselves to a single coordinate system for the problem at hand, we may lose our flexibility for handling various changes in signals; for example, KLT and LDA work perfectly if all the signals or the classes of signals obey the multivariate normal distributions, in fact, they provide the optimal coordinate system under these assumptions. However, for the cases where these assumptions cannot be guaranteed, they may fail miserably. On the other hand, if we try to seek the absolutely best coordinate system without limiting our resources and without assuming models as Kolmogorov suggested, it may take an infinite amount of time to compute or find it. In Yves Meyer's words, we cannot afford to have "the library of Babel" [105] which includes all possible coordinate systems of $\mathbb{R}^n$ in our case. Therefore, a compromise is necessary. As Rissanen correctly pointed out in his

book [128], we need to define a *language* to describe the signals as in the syntactic approach, and within that language, we should seek the best possible solution by optimizing certain criteria depending on the problem at hand. The language in our context is a collection of different coordinate systems or bases. Hence, the performance of the signal analysis tasks strongly depends on what language we use. The language must be flexible and versatile enough to describe various local features of signals such as transients and singularities but also must be computationally efficient to be practical.

This is why we use wavelets and their relatives as library members. They provide flexible coordinate systems which can capture local features in the time-frequency plane in a computationally efficient manner: e.g., to find a good coordinate system for one's problem, it costs $O(n[\log n]^p)$, where $p = 0, 1, 2$ depending on the basis type. These bases are particularly useful for the signals considered in this thesis. Some signals represent the multiscale or fractal nature of geological formations (see e.g., [83], [146]) which may be well-compressed by the standard wavelet bases including the Haar basis. If one cares about edge types (steps or ramps, etc.) in this class of signals, then the autocorrelation functions of wavelets provide a natural way to detect them and characterize them. Other signals represent responses of natural objects to the inputs of acoustic energy of oscillatory nature. These signals may be handled naturally by local trigonometric bases since they perform "local Fourier analysis" by segmenting the time axis into smaller windows. Moreover, these bases "almost diagonalize" the Green's operator for the partial differential equations describing the system responses [16]. The wavelet packets can be considered as a dual version of the local trigonometric bases; they segment the frequency axis so that the oscillatory signals can be handled efficiently. Figure 1.1 shows some of the basis functions in the library.

The connection of this approach to the syntactic approach can be explained as follows: as the words, i.e., the elementary building blocks, we use the basis vectors of the

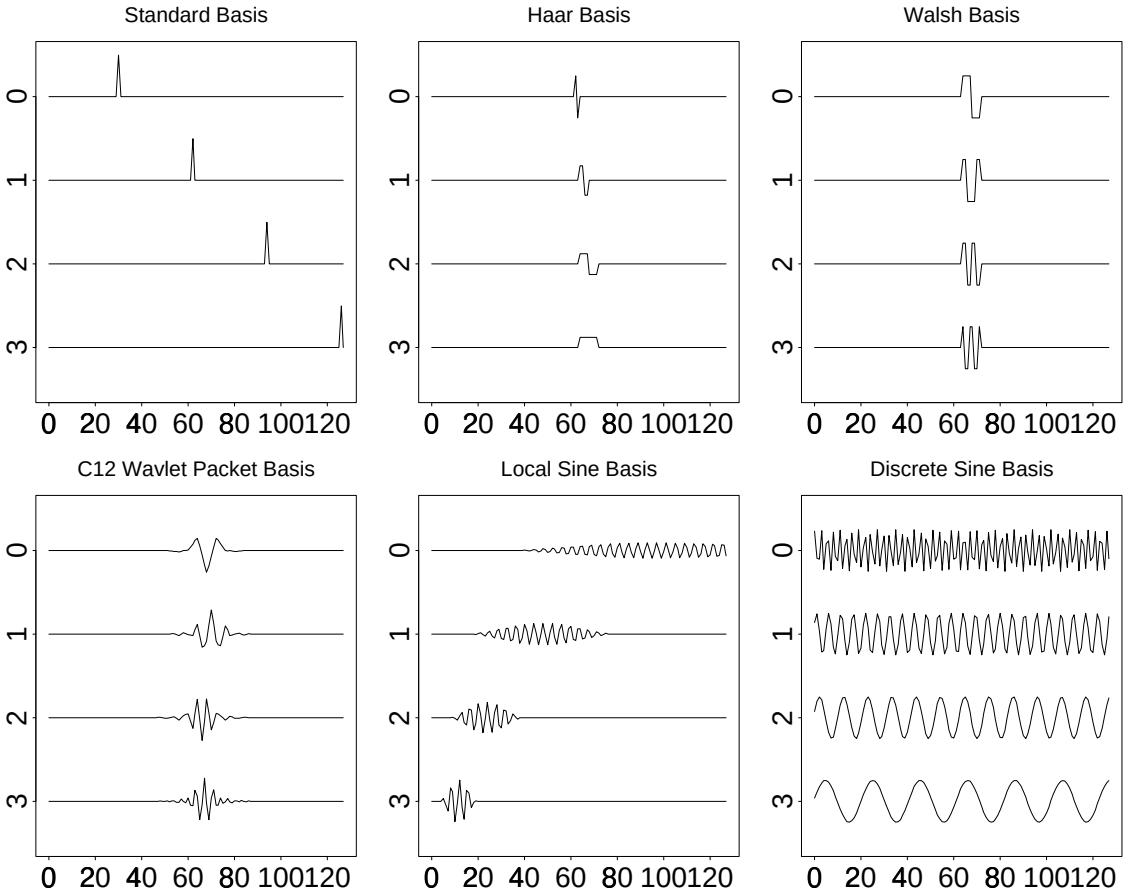


Figure 1.1: Examples of basis functions used in this thesis. Top row: from left, the standard Euclidean basis, the Haar basis, and the Walsh basis. Bottom row: the smooth wavelet packet basis, the local sine basis, and the discrete sine basis. Horizontal axes indicate time in this figure.

wavelets and their relatives, such as the ones shown in Figure 1.1. A collection of words defines a dictionary which corresponds to a collection of wavelet packet bases specified by their frequency localization characteristics, or a collection of local trigonometric bases. As shown in Figure 1.1, each dictionary has its own characteristics or flavor; the "Haar-Walsh dictionary" has various scaled and oscillatory versions of a simple piecewise constant function whereas the "local sine dictionary" contains localized smooth oscillations, etc. A library is defined as a collection of such dictionaries. Finally, the grammar for describing a signal or a class of signals is defined as a binary tree of selected subspaces or basis vectors from this library.

In summary, this paradigm (with the library of bases) provides us with an array of tools bridging between "two extremes," i.e., 1) the standard Euclidean basis and the Fourier basis, 2) the computational efficiency and the descriptive efficiency, and 3) the statistical approaches and the structural approaches. This paradigm leads us to a vastly more efficient representation, processing, and analysis of signals, compared with strategies of confining ourselves to a single basis, or of seeking the absolutely best solution without restricting the size of the library. This thesis is a record or a snapshot of the continuing effort of implementing this philosophy begun by Coifman, Meyer, and Wickerhauser [35], [105].

1.4 Overview of the Thesis

This thesis explores a variety of feature extraction and signal analysis problems using the best-basis paradigm. In particular, the original best-basis paradigm proposed by Coifman, Meyer, and Wickerhauser is extended for: simultaneous noise suppression and signal compression, classification, and regression, focusing on the real geophysical datasets. We also derive a library of non-orthogonal bases using the autocorrelation functions of wavelets for multiscale edge characterization and representation.

We first review our elementary building blocks, i.e., wavelets, wavelet packets, and local trigonometric bases, and define a dictionary and a library of orthonormal bases more precisely in the next chapter. We also review the original best-basis algorithm of Coifman and Wickerhauser, and compare it with the KLT.

The original contribution of this thesis starts from Chapter 3, where we consider how to *simultaneously* remove random noise (e.g., additive white Gaussian noise) and compress a signal component and show how naturally the concept of the minimum description length principle fits the best-basis paradigm. The key observation here is that one or more of the bases in a library of orthonormal bases can compress the signal component quite well whereas the noise component cannot be compressed efficiently by any basis in the library. Based on this observation, we derived an algorithm to estimate the signal component in the data by obtaining the "best" basis and the "best" number of terms to retain using the MDL criterion. Because of the use of the MDL criterion, this algorithm does not require the user to specify any parameter or threshold values.

Then in Chapter 4 we derive an algorithm for selecting a basis suitable for classification from the library. This algorithm uses a discrimination measure such as relative entropy as a basis selection criterion and provides a fast numerical algorithm of $O(n[\log n]^p)$, $p = 1, 2$. Once the basis for classification has been selected [we call this the *Local Discriminant Basis* (LDB)], a small number of most important features (for classification) are supplied to the conventional classifiers such as LDA or CART. Using two synthetic datasets, we demonstrate the superiority of our method over the direct application of these conventional classifiers to the raw input signals. As a further application of LDB, we also describe a method to extract signal component from data consisting of signal and textured background.

In Chapter 5, we extend the best-basis paradigm for regression problems. The proposed method uses prediction error (computed by a specified regression scheme) as a measure of the goodness of bases so that the regression scheme is integrated into the basis

selection mechanism. We call such basis *Local Regression Basis* (LRB). We show that the LRB method can also be used for the classification problems and examine its performance using the same examples as Chapter 4. The LRB method is more flexible and general than the LDB method; however, it is more computationally demanding than the LDB method.

In Chapter 6, we apply the LDB and LRB algorithms to a real geophysical regression problem, that is, prediction of geological information about subsurface formations (e.g., volume fractions of quartz, gas, etc.) from acoustic well-logging waveforms. Using these methods, we extract the useful features for predicting this information. The results, in general, agree with the explanations from the physics of wave propagation, although our use of the physics in constructing the regression rules is minimal. The best results using our methods are found to be comparable to the predictions using the physically-derived quantities.

In Chapter 7, we derive a library of non-orthogonal bases using the autocorrelation functions of wavelets with which we can explicitly extract multiscale edge information and characteristics of singularities. This library provides us shift-invariant multiresolution representations of signals which have: 1) symmetric analyzing functions, 2) shift-invariance, 3) natural and simple iterative interpolation schemes, 4) a simple algorithm for finding the locations of the multiscale edges as zero-crossings. Then we develop a non-iterative method for reconstructing signals from their zero-crossings (and slopes at these zero-crossings) in our representation. This method reduces the problem to that of solving a system of linear equations.

For certain classes of signals, such as frequency-modulated signals (often called chirps), our basis functions in the library may not be too efficient. In Chapter 8, we discuss how to handle this problem and propose an algorithm to discover the modulation laws of simple chirp signals.

Finally, we conclude in Chapter 9 with some discussion on our future projects.

We would like to note that the following chapters are the detailed and expanded version of our published materials. Chapter 3 is based on [129], [130]. Chapter 4 is based on [135], [34]. Chapter 5 is based on [34]. Chapter 7 is based on [133], [132], [134].

This chapter concludes by quoting Yves Meyer [105]:

> "Wavelets, whether they are of the time-scale or time-frequency type, will not help us to *explain* scientific facts, but they will serve to *describe* the reality around us, whether or not it is scientific."

(N.B., emphasis by the author of this thesis.)

In this thesis, we try to show that the good description sometimes makes an explanation of scientific facts easier.

Chapter 2

A Library of Orthonormal Bases

2.1 Introduction

In the previous chapter, we have defined our strategy for feature extraction: analyzing and describing signals using a library of bases, in particular, orthonormal bases of wavelets, wavelet packets, and local trigonometric transforms. In this chapter, we review the most important properties of these bases and precisely define what a dictionary and a library mean, for their applications starting from the next chapter. Since the autocorrelation functions of wavelets will not be used until Chapter 7, we defer their definitions and properties. Then, as a first step of the "best-basis paradigm," we review the "best-basis" algorithm of Coifman and Wickerhauser [35] which was developed mainly for signal compression, and compare this basis with the well-known Karhunen-Loève basis. In the last section, we briefly review the higher dimensional versions of these bases.

Historical aspects and the more details of the properties of these bases can be found in the literature, most notably, in [3], [43], [106], [105], [124], [157].

Throughout this thesis, we consider real-valued discrete signals with finite length n $(= 2^{n_0})$. To focus our attention on our main theme, we assume the periodic boundary

condition on the signals; a signal $\boldsymbol{x} = (x_k)_{k=0}^{n-1}$ is extended periodically beyond the interval $[0, n-1]$ with $x_k \equiv x_{k \pmod n}$ for any $k \in \mathbb{Z}$ if necessary. If one is concerned with the discontinuities created by the periodic boundary condition, their effects can be reduced by considering an evenly-folded version $\boldsymbol{x}' = (x_0, \ldots, x_{n-1}, x_{n-1}, \ldots, x_0)$ of period of $2n$. The compactly-supported orthonormal wavelet bases which do not assume the periodic boundary conditions have been proposed; see e.g., [28]. These can certainly be incorporated in our library of bases.

2.2 Wavelet Bases

The *wavelet transform* ([42], [43], [94], [93], [106]) can be considered as a smooth partition of the frequency axis. The signal is first decomposed into low and high frequency bands by the convolution-subsampling operations with the pair consisting of a "lowpass" filter $\{h_k\}_{k=0}^{L-1}$ and a "highpass" filter $\{g_k\}_{k=0}^{L-1}$ directly on the discrete time domain. Let $\boldsymbol{f} = \{f_k\}_{k=0}^{K-1}$ be a real-valued vector of even length K. Let H and G be the convolution-subsampling operators using these filters which are defined as:

$$(H\boldsymbol{f})_k = \sum_{l=0}^{L-1} h_l f_{2k+l}, \quad (G\boldsymbol{f})_k = \sum_{l=0}^{L-1} g_l f_{2k+l},$$

for $k = 0, 1, \ldots, K-1$. Because of the periodic boundary condition on $\boldsymbol{f}$ (whose period is K), the filtered sequences $H\boldsymbol{f}$ and $G\boldsymbol{f}$ are also periodic with period $K/2$. Their adjoint operations (i.e., upsampling-anticonvolution) H^* and G^* are defined as

$$(H^*\boldsymbol{f})_k = \sum_{0 \le k-2l < L} h_{k-2l} f_l, \quad (G^*\boldsymbol{f})_k = \sum_{0 \le k-2l < L} g_{k-2l} f_l,$$

for $k = 0, 1, \ldots, 2K-1$. The operators H and G are called *(perfect reconstruction) quadrature mirror filters* (QMFs) if they satisfy the following orthogonality (or perfect reconstruction) conditions:

$$HG^* = GH^* = 0, \quad \text{and} \quad H^*H + G^*G = I,$$

where I is the identity operator. These conditions impose some restrictions on the filter coefficients $\{h_k\}$ and $\{g_k\}$. Let m_0 and m_1 be the bounded periodic functions defined by

$$m_0(\xi) = \sum_{k=0}^{L-1} h_k \mathrm{e}^{\mathrm{i}k\xi}, \quad m_1(\xi) = \sum_{k=0}^{L-1} g_k \mathrm{e}^{\mathrm{i}k\xi}.$$

Daubechies proved in [42] that H and G are QMFs if and only if the following matrix is unitary for all $\xi \in \mathbb{R}$:

$$\begin{pmatrix} m_0(\xi) & m_0(\xi+\pi) \\ m_1(\xi) & m_1(\xi+\pi) \end{pmatrix}.$$

Various design criteria (concerning regularity, symmetry etc.) on the lowpass filter coefficients $\{h_l\}$ can be found in [43]. Once $\{h_k\}$ is fixed, we can have QMFs by setting $g_k = (-1)^k h_{L-1-k}$.

This decomposition (or expansion, or analysis) process is iterated on the low frequency bands and each time the high frequency coefficients are retained intact. At the last iteration, both low and high frequency coefficients are kept. In other words, let $\boldsymbol{x} = \{x_k\}_{k=0}^{n-1} \in \mathbb{R}^n$ be a vector to be expanded. Then, the convolution-subsampling operations transform the vector $\boldsymbol{x}$ into two subsequences $H\boldsymbol{x}$ and $G\boldsymbol{x}$ of lengths $n/2$. Next, the same operations are applied to the vector of the lower frequency band $H\boldsymbol{x}$ to obtain $H^2\boldsymbol{x}$ and $GH\boldsymbol{x}$ of lengths $n/4$. If the process is iterated J ($\leq n_0$) times, we have the discrete wavelet coefficients $(G\boldsymbol{x}, GH\boldsymbol{x}, GH^2\boldsymbol{x}, \ldots, GH^{J-1}\boldsymbol{x}, H^J\boldsymbol{x})$ of length n. As a result, the wavelet transform analyzes the data by partitioning its frequency content dyadically finer and finer toward the low frequency region (i.e., coarser and coarser in the original time domain).

If we were to partition the frequency axis sharply using the characteristic functions (or box-car functions), then we would have ended up the so-called Shannon (or Littlewood-Paley) wavelets, i.e., the difference of two sinc functions. Clearly, however, we cannot have a finite-length filter in the time domain in this case. The other extreme is the Haar basis which partitions the frequency axis quite badly but gives the shortest filter length

($L = 2$ with $h_0 = h_1 = 1/\sqrt{2}$) in the time domain and which are suitable for describing discontinuous functions. The QMFs described in detail in [43] essentially bridge between these two extreme cases; see Figure 1.1 of the previous chapter.

The reconstruction (or synthesis) process is also very simple: starting from the lowest two frequency bands $H^J\boldsymbol{x}$ and $GH^{J-1}\boldsymbol{x}$, the adjoint operations are applied and added to obtain $H^{J-1}\boldsymbol{x} = H^*H^J\boldsymbol{x} + G^*GH^{J-1}\boldsymbol{x}$. This process is iterated to reconstruct the original vector $\boldsymbol{x}$. The computational complexity of the decomposition and reconstruction process is in both cases $O(n)$ as easily seen.

Because of the perfect reconstruction condition on H and G, each decomposition step is also considered as a decomposition of the vector space into mutually orthogonal subspaces. Let $\Omega_{0,0}$ denote the standard vector space $\mathbb{R}^n$. Let $\Omega_{1,0}$ and $\Omega_{1,1}$ be mutually orthogonal subspaces generated by the application of the projection operators H and G respectively to the parent space $\Omega_{0,0}$. Then, in general, jth step of the decomposition process can be written as

$$\Omega_{j,0} = \Omega_{j+1,0} \oplus \Omega_{j+1,1} \quad \text{for } j = 0, 1, \ldots, J.$$

It is clear that $\dim \Omega_{j,\cdot} = 2^{n_0-j}$. The wavelet transform is simply a way to represent $\Omega_{0,0}$ by a direct sum of mutually orthogonal subspaces,

$$\Omega_{0,0} = \left(\bigoplus_{j=1}^{J} \Omega_{j,1} \right) \oplus \Omega_{J,0},$$

and the decomposition process is illustrated by the following figure:

We can construct the basis vector $\boldsymbol{w}_{j,k,l} \in \Omega_{j,k}$, $k = 0, 1$, at "scale" j and "position" l ($0 \le l < 2^{n_0-j}, l \in \mathbb{Z}$) simply by putting $(G^kH^{j-k}\boldsymbol{x})_i = \delta_{i,l}$, where $\delta_{i,l}$ denotes the Kronecker delta, setting all the other coefficients at the finer scales than j to zero, and synthesizing $\boldsymbol{x} = \boldsymbol{w}_{j,k,l}$ by the reconstruction algorithm. Using these basis vectors, we can express the wavelet transform in a vector-matrix form as

$$\boldsymbol{\alpha} = \boldsymbol{W}^T\boldsymbol{x},$$

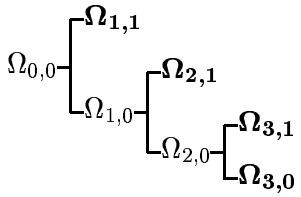


Figure 2.1: A decomposition of $\Omega_{0,0}$ into the mutually orthogonal subspaces using the wavelet transform (with $J = 3$). The symbols in bold font represent the subspaces kept intact by the wavelet transform.

where $\boldsymbol{\alpha} \in \mathbb{R}^n$ contains the wavelet coefficients and $\boldsymbol{W} \in \mathbb{R}^{n \times n}$ is an orthogonal matrix consisting of column vectors $\boldsymbol{w}_{j,k,l}$. This basis vector has the following important properties:

- *vanishing moments*: $\sum_{i=0}^{n-1} i^m \boldsymbol{w}_{j,k,l}(i) = 0$ for $m = 0, 1, \ldots, M-1$.

The higher the degrees of vanishing moments the basis has, the better it compresses the smooth part of the signal. In the original construction of Daubechies [42], it turns out that $L = 2M$. There are several other possibilities. One of them is a family of the so-called "coiflets" with $L = 3M$ [43] which are less asymmetric than the original wavelets of Daubechies.

- *regularity*: $|\boldsymbol{w}_{j,k,l}(i+1) - \boldsymbol{w}_{j,k,l}(i)| \leq c\, 2^{-j\rho}$,

where $c > 0$ is a constant and $\rho > 0$ is called the *regularity* of the wavelets. The larger the value of ρ is, the smoother the basis vector becomes. This property may be important if one requires high compression rate since the shapes of the basis vectors become "visible" in those cases and one might want to avoid fractal-like shapes in the compressed signals/images [122].

- *compact support*: $\boldsymbol{w}_{j,k,l}(i) = 0 \quad \text{for } i \notin [2^j l, 2^j l + (2^j - 1)(L-1)]$.

The compact support property is important for efficient and exact numerical implementation.

Because of these properties, wavelet bases generate very efficient and simple representations for piecewise smooth signals and images.

2.3 Wavelet Packet Bases

For oscillating signals such as acoustic signals, however, the analysis by the wavelet transform is not always efficient because it only partitions the frequency axis finely toward the low frequency. The *wavelet packet transform* ([31], [32], [105], [157]) is a generalized version of the wavelet transform: it decomposes even the high frequency bands which are kept intact in the wavelet transform. Examples of the wavelet packet basis vectors were already shown in Figure 1.1. They are much more oscillatory compared to the wavelet basis vectors. The first level decomposition generates $H\boldsymbol{x}$ and $G\boldsymbol{x}$ just like in the wavelet transform. The second level generates four subsequences, $H^2\boldsymbol{x}, GH\boldsymbol{x}, HG\boldsymbol{x}, G^2\boldsymbol{x}$. If we repeat this process for J times, we end up having Jn expansion coefficients. It is easily seen that the computational cost of this whole process is about $O(Jn) \leq O(n \log_2 n)$. This iterative process naturally generates subspaces of $\mathbb{R}^n$ of a binary tree structure where the nodes of the tree represent subspaces with different frequency localization characteristics. The root node of this tree is again $\Omega_{0,0}$. The node $\Omega_{j,k}$ splits into the two orthogonal subspaces $\Omega_{j+1,2k}$ and $\Omega_{j+1,2k+1}$ by the operators H and G, respectively:

$$\Omega_{j,k} = \Omega_{j+1,2k} \oplus \Omega_{j+1,2k+1} \quad \text{for } j = 0, 1, \ldots, J,\ k = 0, \ldots, 2^j - 1.$$

The following figure shows the binary tree of the subspaces of $\Omega_{0,0}$:

Clearly, we have a redundant set of subspaces in the binary tree. In fact, it is easily proved that there are more than $2^{2^{(J-1)}}$ possible orthonormal bases in this binary tree; see e.g. [157]. This binary tree is our main tool in this thesis:

Definition 2.1. A *dictionary of orthonormal bases* $\mathfrak{D}$ for $\mathbb{R}^n$ is a binary tree if it satisfies:

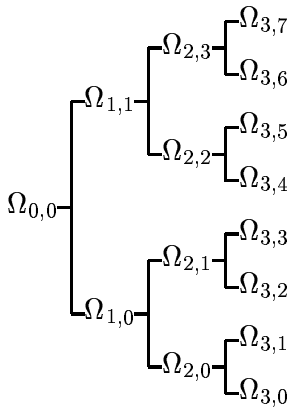


Figure 2.2: A decomposition of $\Omega_{0,0}$ into the tree-structured subspaces using the wavelet packet transform (with $J = 3$).

(a) Subsets of basis vectors can be identified with subintervals of $I = [0, n)$ of the form $I_{j,k} = [2^{n_0-j}k, 2^{n_0-j}(k+1))$, for $j = 0, 1, \ldots, J,\ k = 0, 1, \ldots, 2^j - 1$, where $J \leq n_0$.

(b) Each basis in the dictionary corresponds to a disjoint cover of I by intervals $I_{j,k}$.

(c) If $\Omega_{j,k}$ is the subspace identified with $I_{j,k}$, then $\Omega_{j,k} = \Omega_{j+1,2k} \oplus \Omega_{j+1,2k+1}$.

Each subspace $\Omega_{j,k}$ is spanned by 2^{n_0-j} basis vectors $\{\boldsymbol{w}_{j,k,l}\}_{l=0}^{2^{n_0-j}-1}$. In the wavelet packet dictionary, the parameters k and l roughly indicate frequency bands[1] and the location of the center of wiggles, respectively: the vector $\boldsymbol{w}_{j,k,l}$ is roughly centered at $2^j l$, has length of support $\approx 2^j$, and oscillates $\approx k$ times. Note that for $j = 0$, we have the standard Euclidean basis of $\mathbb{R}^n$. By specifying a pair of QMFs, we obtain one dictionary which contains a large number of orthonormal bases of $\mathbb{R}^n$: we have a large number of coordinate systems to "view our signals" at our disposal. An important question is how to select the best coordinate system efficiently for the problem at hand from this dictionary. This is the main theme of this thesis.

[1] The original binary tree generated by successive applications of H and G is called "Paley ordered" and the frequency band of $\Omega_{j,k}$ is not monotonically increasing as a function of k. This behavior is corrected by the so-called "Gray code" permutation; see [157] for the details.

2.4 Local Trigonometric Bases

Local trigonometric transforms ([33], [3], [105], [157]) or lapped orthogonal transform ([99], [100]) can be considered as conjugates of wavelet packet transforms: they partition the time axis smoothly. In fact, Coifman and Meyer [33] showed that it is possible to partition the real-line into any disjoint intervals smoothly and construct orthonormal bases on each interval. Each basis function on an interval uses the signal values on the interval itself and on the adjacent intervals; hence it is named the "lapped" orthogonal transform. It is also "local" since this essentially performs the Fourier analysis on the short intervals. Figure 1.1 in the previous chapter clearly showed this capability. Since it partitions the time axis smoothly, these local cosine and sine transforms (LCT/LST), have less edge (or blocking) effects than the conventional discrete cosine/sine transforms (DCT/DST).

For definiteness we use a particular symmetric window (or "bell") function

$$b(t) \triangleq \begin{cases} \sin\frac{\pi}{4}(1+\sin\pi t) & \text{if } -\frac{1}{2} < t < \frac{3}{2}, \\ 0 & \text{otherwise.} \end{cases}$$

This function generates a partition of unity:

$$\sum_{k=-\infty}^{\infty} b^2(t+k) = 1 \quad \text{for any } t \in \mathbb{R}.$$

This function is also symmetric about $t = 1/2$, smooth on $(-1/2, 3/2)$ with vanishing derivatives at the boundary points, so that it has a continuous derivative on $\mathbb{R}$; see [157] for more general bell functions. Now let us consider the interval of integers $I = \{0, 1, \ldots, n-1\}$. The lapped orthogonal functions are mainly supported on I but also take some values on $\{-n/2, \ldots, -1\}$ and $\{n, \ldots, 3n/2 - 1\}$. For integers $k \in I$, we can define the *local sine basis functions* on the interval I:

$$S_k(l) \triangleq \frac{1}{\sqrt{2n}} b\left(\frac{l+\frac{1}{2}}{n}\right) \sin\left(\pi\left(k+\frac{1}{2}\right)\left[\frac{l+\frac{1}{2}}{n}\right]\right).$$

By replacing sin by cos, we obtain the local cosine basis function $C_k(l)$ as well. Apart from the bell function factor, these are exactly the basis functions for the so-called DST-IV (and DCT-IV for cosines) [121]. The orthogonality condition certainly holds since

$$\sum_{l=-n/2}^{3n/2-1} S_k(l) S_{k'}(l) = \delta_{k,k'}.$$

The bell function allows sines on adjacent intervals to overlap while remaining orthogonal. For example, the function $S_k(l+n)$ is centered over the range $l \in \{-n, -n+1, \ldots, -1\}$ and overlaps with the function $S_{k'}(l)$ at $l \in \{-n/2, -n/2+1, \ldots, n/2-1\}$. Yet these two are orthogonal since

$$\sum_{l=-n/2}^{n/2} S_k(l+n) S_{k'}(l) = 0 \quad \text{for any } k, k' \in \{0, 1, \ldots, n-1\}.$$

In the numerical implementation of these transforms, we should not calculate the actual inner products of data and the basis functions. Instead, we should preprocess the data so that the standard DST-IV (or DCT-IV for LCT) algorithms may be used. This is accomplished by "folding" the overlapping parts of the bells back into the interval. This folding can be transposed onto the data and results in the disjoint intervals of samples over which the DST-IV algorithms can be applied. These folded disjoint intervals can also be "unfolded" to produce smooth overlapping segments. The details of these operations can be found in [3], [156], [157].

Because we can segment an interval into arbitrary disjoint intervals, it is natural to segment the whole interval into dyadic subintervals recursively. This segmentation makes the number of signal samples contained in each subinterval a dyadic number (2^{n_0-j} at step j) if the length of the original signal is also a dyadic number (2^{n_0}); This enables one to utilize the fast DCT/DST-IV algorithm [121]. By this segmentation, the original interval $I = [0, n)$ is split into $[0, n/2)$ and $[n/2, n)$, and each subinterval is further split into half in a recursive manner. Let us set $I_{0,0} = I$ and let $I_{j,k}$ be a subinterval of I after jth iteration

of the splitting process. Then we have a familiar relation

$$I_{j,k} = I_{j+1,2k} \cup I_{j+1,2k+1} \quad \text{for } j = 0, 1, \ldots, J, \ k = 0, 1, \ldots, 2^j - 1.$$

Now we can consider the subspaces $\Omega_{j,k}$ associated with the interval $I_{j,k}$. Then we obtain a binary tree of the subspaces with the same tree structure as the one shown in Figure 2.2. Each subspace is spanned by the basis vectors $\{\boldsymbol{w}_{j,k,l}\}_{l=0}^{2^{n_0-j}-1}$ where

$$\boldsymbol{w}_{j,k,l}(m) = \frac{1}{\sqrt{2^{n_0-j+1}}} b\left(\frac{m+\frac{1}{2}}{2^{n_0-j}} - k\right) \sin\left(\pi\left(l+\frac{1}{2}\right)\left[\frac{m+\frac{1}{2}}{2^{n_0-j}} - k\right]\right),$$

for LST. The LCT version can be obtained in an obvious manner. Note that the triplet (j, k, l) now corresponds to scale, location (or window index) and frequency, respectively. For $j = 0$, this reduces to a simple DCT/DST. Hence, we can obtain two additional dictionaries of orthonormal bases using LCT/LST. The computational complexity to obtain this dictionary (or expanding a signal into this dictionary) is about $O(n[\log_2 n]^2)$; see e.g., [157]. Once obtained the dictionary of orthonormal bases, the question is again how to select the best basis for the problem at hand from the collection of bases. In the next section, we review the best-basis algorithm of Coifman-Wickerhauser for signal compression.

2.5 Selection of a "Best Basis" from a Library of Orthonormal Bases

2.5.1 Information cost functions

An efficient coordinate system for representing a signal should give large magnitudes along a few axes and negligible magnitudes along most axes when the signal is expanded into the associated basis. We then need a measure to evaluate and compare the efficiency of many bases. Let $\mathcal{I}$ denote this measure which is often called "information cost" function. There are several choices for $\mathcal{I}$; see e.g., [157], [63, Chapter 9], [112]. All of them essentially measure

the "energy concentration" of the coordinate vector. A natural choice for this measure is the *Shannon entropy* of the coordinate vector [35], [157]. Let us define the entropy of a nonnegative sequence $\boldsymbol{p} = \{p_i\}$ with $\sum_i p_i = 1$ by

$$H(\boldsymbol{p}) \stackrel{\Delta}{=} -\sum_i p_i \log_2 p_i, \tag{2.1}$$

with the convention $0 \cdot \log 0 = 0$. (From now on, we use "log" for the logarithm of base 2.) For a signal $\boldsymbol{x}$, we set $p_i = (|x_i|/\|\boldsymbol{x}\|_r)^r$ where $\|\cdot\|_r$ is the ℓ^r norm and $1 \leq r < \infty$ and define

$$H_r(\boldsymbol{x}) \stackrel{\Delta}{=} -\sum_i \frac{|x_i|^r}{\|\boldsymbol{x}\|_r^r} \log \frac{|x_i|^r}{\|\boldsymbol{x}\|_r^r}. \tag{2.2}$$

Often $r = 1$ or $r = 2$ is used. In this thesis, we always use $r = 2$.

2.5.2 Best basis selection from a dictionary of orthonormal bases

The "best-basis" algorithm of Coifman and Wickerhauser [35] was developed mainly for signal compression. This method first expands a given *single* signal into a specified dictionary of orthonormal bases. Then a complete basis called a *best basis* (BB) which minimizes a certain information cost function such as entropy (2.2) is searched in this binary tree using the divide-and-conquer algorithm. More precisely, let $B_{j,k}$ denote a set of basis vectors belonging to the subspace $\Omega_{j,k}$ arranged as a matrix

$$B_{j,k} = (\boldsymbol{w}_{j,k,0}, \ldots, \boldsymbol{w}_{j,k,2^{n_0-j}-1})^T. \tag{2.3}$$

Now let $A_{j,k}$ be the best basis for the signal $\boldsymbol{x}$ restricted to the span of $B_{j,k}$ and let $\mathcal{I}$ be an information cost function measuring the goodness of nodes (subspaces) for compression. The following best-basis algorithm essentially "prunes" this binary tree by comparing efficiency of each parent node and its two children nodes:

Algorithm 2.2 (The Best-Basis Algorithm [35]). *Given a vector* $\boldsymbol{x}$,

Step 0: *Choose a dictionary of orthonormal bases $\mathfrak{D}$ (i.e., specify QMFs for a wavelet packet dictionary or decide to use either the local cosine dictionary or the local sine dictionary) and specify the maximum depth of decomposition J and an information cost $\mathcal{I}$.*

Step 1: *Expand $\boldsymbol{x}$ into the dictionary $\mathfrak{D}$ and obtain coefficients $\{B_{j,k}\boldsymbol{x}\}_{0\le j\le J,\ 0\le k\le 2^j-1}$.*

Step 2: *Set $A_{J,k} = B_{J,k}$ for $k = 0, \ldots, 2^J - 1$.*

Step 3: *Determine the best subspace $A_{j,k}$ for $j = J-1, \ldots, 0,\ k = 0, \ldots, 2^j - 1$ by*

$$A_{j,k} = \begin{cases} B_{j,k} & \text{if } \mathcal{I}(B_{j,k}\boldsymbol{x}) \le \mathcal{I}(A_{j+1,2k}\boldsymbol{x} \cup A_{j+1,2k+1}\boldsymbol{x}), \\ A_{j+1,2k} \oplus A_{j+1,2k+1} & \text{otherwise.} \end{cases} \tag{2.4}$$

To make this algorithm fast, the cost functional $\mathcal{I}$ needs to be *additive*:

Definition 2.3. A map $\mathcal{I}$ from sequences $\{x_i\}$ to $\mathbb{R}$ is said to be *additive* if $\mathcal{I}(0) = 0$ and $\mathcal{I}(\{x_i\}) = \sum_i \mathcal{I}(x_i)$.

Thus, if $\mathcal{I}$ is additive, then in (2.4) we have

$$\mathcal{I}(A_{j+1,2k}\boldsymbol{x} \cup A_{j+1,2k+1}\boldsymbol{x}) = \mathcal{I}(A_{j+1,2k}\boldsymbol{x}) + \mathcal{I}(A_{j+1,2k+1}\boldsymbol{x}).$$

This implies that a simple addition suffices instead of computing the cost of union of the nodes. Although (2.1) is additive with respect to $\boldsymbol{p}$, $H_r(\boldsymbol{x})$ is not additive with respect to $\boldsymbol{x}$. But it is easy to show that minimizing the additive measure

$$h_r(\boldsymbol{x}) \stackrel{\Delta}{=} -\sum_i |x_i|^r \log |x_i|^r \tag{2.5}$$

implies minimizing $H_r(\boldsymbol{x})$ since $H_r(\boldsymbol{x}) = h_r(\boldsymbol{x})/\|\boldsymbol{x}\|_r^r + \log \|\boldsymbol{x}\|_r^r$.

With the additive information cost function, we have the following proposition:

Proposition 2.4 (Coifman & Wickerhauser [35]). *Algorithm 2.2 yields the best basis relative to $\mathcal{I}$ if $\mathcal{I}$ is additive.*

See [35], [157] for the proof.

The computational complexity of computing the best basis from a dictionary is $O(n \log n)$ for a wavelet packet dictionary and $O(n[\log n]^2)$ for a local trigonometric dictionary; it is dominated by the expansion of a signal into the dictionary and the cost for searching the best basis is about $O(n)$ because of the use of the divide-and-conquer algorithm. The reconstruction of the original vector from the best-basis coefficients has the same computational complexity.

2.5.3 Best basis selection from a library of orthonormal bases

We now consider a "meta" algorithm for the best basis selection.

Definition 2.5. A *library of orthonormal bases* for $\mathbb{R}^n$ is a collection of the dictionaries of orthonormal bases for $\mathbb{R}^n$.

This library of bases is more adaptable and versatile for representing various transient signals than a single dictionary of bases is. For example, if the signal consists of blocky functions such as acoustic impedance profiles of subsurface structure, the Haar-Walsh dictionary captures those discontinuous features both accurately and efficiently. If the signal consists of piecewise polynomial functions of order p, then the Daubechies wavelets/wavelet packets with filter length $L \geq 2(p+1)$ or the coiflets with filter length $L \geq 3(p+1)$ would be efficient because of the vanishing moment property. If the signal has a sinusoidal shape or highly oscillating characteristics, the local trigonometric bases would do the job. Moreover, computational efficiency of this library is also attractive; the most expensive expansion in this library, i.e., the local trigonometric expansion, costs about $O(n[\log n]^2)$.

How can we choose the best dictionary from this library? The strategy of Coifman and Majid [30] is very simple: pick the one giving the minimum entropy among them. (We note that the purpose of [30] is not the compression but the noise removal. More on this aspect in the next chapter.) More precisely, let $\mathfrak{L} = \{\mathfrak{D}_1, \ldots, \mathfrak{D}_M\}$ denote a library

of orthonormal bases where $\mathfrak{D}_m$ represents a dictionary of orthonormal bases. For each dictionary $\mathfrak{D}_m$, the best basis $\mathfrak{B}_m$ of the signal $\boldsymbol{x}$ is computed by Algorithm 2.2. This generates M different sets of the expansion coefficients $\{\boldsymbol{\alpha}_m\}_{m=1}^M$ of the signal. For each expansion coefficient set, entropy $h_2(\boldsymbol{\alpha}_m)$ defined in (2.5) is computed and then the basis which gives the minimum entropy among M entropy values is selected as the "best of the best bases". We consider a closely related but different criterion based on the Rissanen's minimum description length principle [128] in the next chapter.

2.6 Joint Best Basis and Karhunen-Loève Basis

To compress a given set of signals $\{\boldsymbol{x}_i\}_{i=1}^N \subset \mathcal{X} \subset \mathbb{R}^n$ rather than a single signal, one of the well-known traditional methods is the *Karhunen-Loève transform* (KLT) [150], [1], [63], [112]. Let X denote the data matrix: $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) \in \mathbb{R}^{n \times N}$. Then, the Karhunen-Loève basis (KLB) for a finite number of real-valued discrete signals of finite lengths is defined as the eigenvectors of the symmetric positive definite matrix called the *sample autocorrelation matrix*

$$R_X \stackrel{\Delta}{=} \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i \boldsymbol{x}_i^T = \frac{1}{N} X X^T.$$

The KLB satisfies a number of optimality criteria, and in particular, it is *the* minimum entropy basis in $\mathbb{R}^n$ among all the orthonormal bases associated with orthogonal transformations in $\mathrm{O}(n)$ [150]. To measure the efficiency of the coordinate systems for a set of signals, the entropy of the total energy distribution to the coordinate axes is used. This energy distribution is defined as the normalized diagonal vector of the sample autocorrelation matrix:

$$\boldsymbol{\gamma}_X \stackrel{\Delta}{=} \frac{\mathrm{diag}(R_X)}{\| \mathrm{diag}(R_X) \|}.$$

Thus, $\boldsymbol{\gamma}_X(i)$ represents the energy distribution to the ith coordinate axis of the signals in the original basis which is normally either the standard Euclidean basis or the discrete

trigonometric basis. Then, $H(\boldsymbol{\gamma}_X)$, the entropy (2.1) for the vector $\boldsymbol{\gamma}_X$ is well-defined. Now we rotate the axes or apply an orthogonal transformation $U \in \mathrm{O}(n)$ to the set of signals. This generates a set of transformed signals $\boldsymbol{y}_i = U^T \boldsymbol{x}_i$ for $i = 1, \ldots, N$. Let $Y = U^T X$. Then, the energy distribution of the new coordinate system is given by $\boldsymbol{\gamma}_Y = \mathrm{diag}(R_Y) = \mathrm{diag}(R_{U^T X})$. S. Watanabe proved the following theorem:

Theorem 2.6 (Watanabe [150], [152]). *The orthogonal matrix $U \in \mathrm{O}(n)$ is the Karhunen-Loève basis U_{KL} if and only if*

$$H(\boldsymbol{\gamma}_{U_{KL}^T X}) = \min_{U \in \mathrm{O}(n)} H(\boldsymbol{\gamma}_{U^T X}).$$

The main problem of the KLB is its computational cost $O(n^3)$ for diagonalizing R_X. In fact, its dependence on the eigenvalue system creates more problems; the sensitivity to the alignment of the signals and difficulty in capturing local features in the signals.

Wickerhauser proposed a method to overcome these problems of the KLB using the "best-basis paradigm," which is an extension to the best-basis method [155], [157]. Let us fix a dictionary $\mathfrak{D}$. Then, the idea is to use the energy distribution of the set of the signals to the coordinate axes in $\mathfrak{D}$ by computing $\sum_{i=1}^{N} (\boldsymbol{w}_{j,k,l}^T \boldsymbol{x}_i)^2$ for each (j, k, l) and organize them into a binary tree so that the divide-and-conquer algorithm can search a basis minimizing the entropy of the energy distribution from the tree-structured subspaces corresponding to the the tree of energy distribution. Such a best basis is called the *joint best basis* (JBB) for $\{\boldsymbol{x}_i\}_{i=1}^N$ In this thesis, we will also use the term "best basis" as a joint best basis for simplicity. The following is the algorithm for obtaining such a best basis derived by Wickerhauser. We also show the computational cost at each step.

Algorithm 2.7 (The Joint-Best-Basis Algorithm [155]). *Given a set of signals $\{\boldsymbol{x}_i\}_{i=1}^N$,*

Step 0: *Choose a dictionary of orthonormal bases $\mathfrak{D}$ (i.e., specify QMFs for a wavelet packet dictionary or decide to use either the local cosine dictionary or the local sine*

dictionary) and specify maximum depth of decomposition J and an additive information cost $\mathcal{I}$.

Step 1: *Expand $\{\boldsymbol{x}_i\}_{i=1}^N$ into the dictionary $\mathfrak{D}$; $O(Nn\log n)$.*

Step 2: *Sum the squares of the expansion coefficients into the tree of energy distribution; $O(n\log n)$.*

Step 3: *Search the tree for a joint best basis; $O(n)$.*

Step 4: *Sorting the best basis vectors into decreasing order of importance; $O(n\log n)$.*

Step 5: *Use the top k ($\leq n$) best-basis vectors for representing the signals.*

One can further apply the KLT to the top k best-basis vectors to compress more with additional cost $O(k^3)$; however, since a good amount of compression is usually achieved at Step 5, this final KLT may not be necessary.

We note that a joint best basis as well as a KLB can also be computed after subtracting the mean $(1/N)\sum_{i=1}^N \boldsymbol{x}_i$ from each signal.

The entropy criterion used in the best-basis algorithm is good for signal compression; however, it may not be necessarily good for other problems. We extend this algorithm for classification in Chapter 4 and for regression in Chapter 5.

2.7 Extension to Images

For images or multidimensional signals, we can easily extend the above algorithms by using the multidimensional version of the wavelets, wavelet packets, and local trigonometric transforms. In this section, we briefly summarize the two-dimensional (2D) versions of these transforms. For the 2D wavelets, there are several different approaches. The first one, which we call the sequential method, is the tensor product of the one-dimensional (1D) wavelets: applying the wavelet expansion algorithm separately along two axes t_1 and t_2 corresponding

to column (vertical) and row (horizontal) directions respectively. Let $\boldsymbol{x} \in \mathbb{R}^{n_1 \times n_2}$ and H_i, G_i be the 1D convolution-subsampling operators defined on matrices along axis $t_i, i = 1, 2$. Then this version of the 2D wavelet transform first applies the convolution-subsampling operations along the t_1 axis to obtain $\boldsymbol{x}_1 = (G_1\boldsymbol{x}, G_1H_1\boldsymbol{x}, \ldots, G_1H_1^{J_1}\boldsymbol{x})$, then applies the convolution-subsampling operations along the t_2 axis to get the final 2D wavelet coefficients $(G_2\boldsymbol{x}_1, G_2H_2\boldsymbol{x}_1, \ldots, G_2H_2^{J_2}\boldsymbol{x}_1)$ of length $n_1 \times n_2$, where J_1 ($\leq \log n_1$) and J_2 ($\leq \log n_2$) are maximum levels of decomposition along t_1 and t_2 axes respectively. We note that one can choose different 1D wavelet bases for t_1 and t_2 axes independently. Given M different QMF pairs, there exist M^2 possible 2D wavelets using this approach.

The second approach is the basis generated from the tensor product of the multiresolution analysis. This decomposes an image $\boldsymbol{x}$ into four different sets of coefficients, $H_1H_2\boldsymbol{x}$, $G_1H_2\boldsymbol{x}$, $H_1G_2\boldsymbol{x}$, and $G_1G_2\boldsymbol{x}$, corresponding to "low-low", "high-low", "low-high", "high-high" frequency bands of the image, respectively. The decomposition is iterated on the "low-low" frequency band and this ends up in a "pyramid" structure of coefficients. Transforming the digital images by these wavelets to obtain the 2D wavelet coefficients are described in detail in e.g., [94], [43].

There are also 2D wavelet bases which do not have a tensor-product structure, such as wavelets on the hexagonal grids and wavelets with matrix dilations; see e.g., [84], [65] for the details.

There has been some argument as to which version of the 2D wavelet bases should be used for various applications [13], [43]. Our strategy toward this problem is this: we can put as many versions of these bases in the library as we can afford it in terms of computational resources. Then we select the best possible basis for the problem at hand.

As for the 2D version of the wavelet packet transform, the sequential method may be generalized, but it is not easily interpreted; the 1D best-bases may be different from column to column so that the resultant coefficients viewing along the row direction may not

share the same frequency bands and scales unlike the 2D wavelet bases. This also makes the reconstruction algorithm complicated. Therefore, we should use the other tensor-product 2D wavelet approach for the construction of the 2D wavelet packet best-basis: we recursively decompose not only the "low-low" frequency band but also the other three frequency bands. This process produces the "quad-tree" structure of subspaces instead of the "binary-tree" structure for 1D wavelet packets.

The 2D version of the local trigonometric transforms can be constructed similarly: the original image is smoothly folded and segmented into 4 subimages recursively, and in each subimage the separable DCT/DST-IV is applied. This process results in the quad-tree structure of the subspaces.

Thus, these 2D wavelet packet transforms and local trigonometric transforms generate dictionaries of orthonormal bases of the quad-tree structure. The 2D best basis can be selected in the same manner as the 1D version. As for the computational complexity for an image of $n = n_1 \times n_2$ pixels, it costs approximately $O(n)$, $O(n \log_4 n)$, $O(n[\log_4 n]^2)$ to compute a 2D wavelet, a 2D wavelet packet best-basis, a 2D local trigonometric best-basis, respectively; see [157] and [156] for the details of the 2D version of the best-basis algorithms.

Chapter 3

A Simultaneous Noise Suppression and Signal Compression Algorithm

3.1 Introduction

This chapter attempts to bridge between the "best-basis paradigm" and Rissanen's minimum description length (MDL) principle [128]. In particular, we propose an algorithm to *simultaneously* reduce random noise in signals and compress the "structural" component of the signals using a library of orthonormal bases and the MDL criterion.

Wavelet transforms and their relatives have given a major impact on data compression field; see e.g., [49], [17], [156], [138], [73]. Meanwhile, several researchers have claimed that wavelets and their relatives are also useful for reducing noise in (or denoising) signals/images [53], [51], [30], [36], [97], [92]. In this chapter, we take advantage of both sides: we show compression of signals leads to random noise suppression in the signals. In other words, we try to "kill two birds with one stone."

Throughout this chapter, we consider a simple degradation model: observed data consists of a signal component and additive white Gaussian noise although it is possible

to extend to more general noise models. The key motivation here is that the structural component in signals can often be efficiently represented by one or more of the bases in the library whereas the noise component cannot be represented efficiently by any basis in the library. The use of the MDL criterion frees us from any subjective parameter setting such as threshold selection. This is particularly important for real field data where the noise level is difficult to obtain or estimate *a priori*.

The organization of this chapter is as follows. In Section 3.2, we formulate our problem. We view the problem of simultaneous noise suppression and signal compression as a model selection problem out of models generated by the library of orthonormal bases. In Section 3.3, we review the MDL principle which plays a critical role in this thesis. We also give some simple examples to help understand its concept. In Section 3.4, we develop an actual algorithm of simultaneous noise suppression and signal compression. We also give the computational complexity of our algorithm. In Section 3.5, we apply our algorithm to several geophysical datasets, both synthetic and real, and compare the results with other competing methods. We discuss the connection of our algorithm with other approaches in Section 3.6.

3.2 Problem Formulation

Let us consider a discrete degradation model

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

where $\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\epsilon} \in \mathbb{R}^n$ and $n = 2^{n_0}$. The vector $\boldsymbol{y}$ represents the noisy observed data and $\boldsymbol{x}$ is the unknown true signal to be estimated. The vector $\boldsymbol{\epsilon}$ is white Gaussian noise (WGN), i.e., $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$. Let us assume that σ^2 is unknown.

We now consider an algorithm to estimate $\boldsymbol{x}$ from the noisy observation $\boldsymbol{y}$. First, we prepare the library of orthonormal bases for $\boldsymbol{x}$ described in the previous chapter. In this

chapter, we use the library $\mathfrak{L} = \{\mathfrak{B}_1, \mathfrak{B}_2, \ldots, \mathfrak{B}_M\}$, where $\mathfrak{B}_m$ represents either the best basis selected from a dictionary of orthonormal bases or a non-adaptable basis such as the wavelet basis or the discrete sine (or cosine) basis. We can certainly add other orthonormal bases to this library such as the ones we will develop in the subsequent chapters. The number of bases M in this library typically ranges from 5 to 20 depending on the computational resources and the *a priori* knowledge on the signal $\boldsymbol{x}$. Since the bases in the library $\mathfrak{L}$ compress signals/images very well, we make a strong assumption here: we suppose the unknown signal $\boldsymbol{x}$ can be *completely* represented by k ($< n$) elements of a basis $\mathfrak{B}_m$, i.e.,

$$\boldsymbol{x} = \boldsymbol{W}_m \boldsymbol{\alpha}_m^{(k)}, \tag{3.1}$$

where $\boldsymbol{W}_m \in \mathbb{R}^{n \times n}$ is an orthogonal matrix whose column vectors are the basis elements of $\mathfrak{B}_m$, and $\boldsymbol{\alpha}_m^{(k)} \in \mathbb{R}^n$ is the vector of expansion coefficients of $\boldsymbol{x}$ with only k non-zero coefficients. At this point, we do not know the actual value of k and the basis $\mathfrak{B}_m$. We would like to emphasize that in reality the signal $\boldsymbol{x}$ might not be strictly represented by (3.1). We regard (3.1) as a *model at hand* rather than a rigid physical model exactly *explaining* $\boldsymbol{x}$ and we will try our best under this assumption. (This is often the case if we want to fit polynomials to some data.) Now the problem of simultaneous noise suppression and signal compression can be stated as follows: *find the "best" k and m given the library $\mathfrak{L}$*. In other words, we translate the estimation problem into a model selection problem where models are the bases $\mathfrak{B}_m$ and the number of terms k under the additive WGN assumption.

For the purpose of data compression, we want to have k as small as possible. At the same time, we want to minimize the distortion between the estimate and the true signal by choosing the most suitable basis $\mathfrak{B}_m$, keeping in mind that the larger k normally gives smaller value of error. How can we satisfy these seemingly conflicting demands?

3.3 The Minimum Description Length Principle

To satisfy the above mentioned conflicting demands, we need a model selection criterion. One of the most suitable criteria for our purpose is the so-called *Minimum Description Length* (MDL) criterion proposed by Rissanen [126], [127], [128]. The MDL principle suggests that the "best" model among the given collection of models is the one giving the shortest description of the data *and* the model itself. For each model in the collection, the length of description of the data is counted as the codelength of encoding the data using that model in binary digits (bits). The length of description of a model is the codelength of specifying that model, e.g., the number of parameters and their values if it is a parametric model.

To help understand what "code" or "encoding" means, we give some simple examples. We assume that we want to transmit data by first encoding (mapping) them into a bitstream by an encoder, then receive the bitstream by a decoder, and finally try to reconstruct the data. Let $L(\boldsymbol{x})$ denote the codelength (in bits) of a vector $\boldsymbol{x}$ of deterministic or probabilistic parameters which are either real-valued, integer-valued, or taking values in a finite alphabet.

Example 3.1. *Codelength of symbols drawn from a finite alphabet.*
Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ be a string of symbols drawn from a finite alphabet $\mathcal{X}$, which are independently and identically distributed (i.i.d.) with probability mass function $p(x)$, $x \in \mathcal{X}$. In this case, clearly the frequently occurring symbols should have shorter codelengths than rarely occurring symbols for efficient communication. This leads to the so-called Shannon code [40] whose codelength (if we ignore the integer requirement for the codelength) can be written as

$$L(x) = -\log p(x) \qquad \text{for } x \in \mathcal{X}.$$

The Shannon code has the shortest codelength *on the average*, and satisfies the so-called

Kraft inequality [40]:

$$\sum_{x \in \mathcal{X}} 2^{-L(x)} \leq 1, \tag{3.2}$$

which is necessary and sufficient for the existence of an instantaneously decodable code, i.e., a code such that there is no codeword which is the prefix of any other codeword in the coding system. The shortest codelength on the average for the whole sequence $\boldsymbol{x}$ becomes

$$L(\boldsymbol{x}) = \sum_{i=1}^{n} L(x_i) = -\sum_{i=1}^{n} \log p(x_i).$$

Example 3.2. *Codelength of deterministic integers.*

For a deterministic parameter $j \in \mathbb{Z}_n = (0, 1, \ldots, n-1)$ (i.e., both the encoder and decoder know n), the codelength of describing j is written as $L(j) = \log n$ since $\log n$ bits are required to index n integers. This can also be interpreted as a codelength using Shannon code for a sample drawn from the uniform distribution over $(0, 1, \ldots, n-1)$.

Example 3.3. *Codelength of an integer (universal prior for an integer).*

Suppose we do not know how large a natural number j is. Rissanen [126] proposed that the code of such j should be the binary representation of j, preceded by the code describing its length $\log j$, preceded by the code describing the length of the code for $\log j$, and so forth. This recursive strategy leads to

$$L^*(j) = \log^* j + \log c_0 = \log j + \log\log j + \cdots + \log c_0,$$

where the sum involves only the non-negative terms and the constant $c_0 \approx 2.865064$ which was computed so that equality holds in (3.2), i.e., $\sum_{j=1}^{\infty} 2^{-L^*(j)} = 1$. This can be generalized for an integer j by defining

$$L^*(j) = \begin{cases} 1 & \text{if } j = 0, \\ \log^* |j| + \log 4c_0 & \text{otherwise.} \end{cases} \tag{3.3}$$

(We can easily see that (3.3) satisfies $\sum_{j=-\infty}^{\infty} 2^{-L^*(j)} = 1$.)

The following two examples are important for pruning the tree-based classification and regression rules used in Chapters 4, 5, and 6; see Appendix A for the details.

Example 3.4. *Codelength of a binary string with specified number of 1s.*
Let $\boldsymbol{x}$ be a binary string of length n containing exactly k 1s. Then, we must describe: 1) the integer k which requires $log(n+1)$ bits since $0 \leq k \leq n$, and 2) the index of this string in the list of all possible strings of length n with k 1s, which costs $\log \binom{n}{k}$ bits [128], [117]. Hence the total description length is

$$L(n,k) = \log(n+1) + \log \binom{n}{k} = \log \frac{(n+1)!}{k!\,(n-k)!} \tag{3.4}$$

bits. Notice that $L(n,k)$ does not depend on the position of 1s but just on the number of 1s. This encoding scheme wins over the obvious code (i.e., just sending it as is, which costs n bits) when k is small.

As a generalization of this example,

Example 3.5. *Codelength of a d-ary string with specified numbers of symbols*
Let $\boldsymbol{x}$ be a string of symbols from d-ary alphabet, say, $\{0, 1, \ldots, d-1\}$ and let n be the length of $\boldsymbol{x}$ and let n_i be the number of occurrences of symbol i in $\boldsymbol{x}$, i.e., $n = n_0 + \cdots + n_{d-1}$. To encode this string, we need to specify: 1) the description length of numbers of occurrences of symbols, $(n_0, \ldots, n_{d-1})$, which requires $\log \binom{n+d-1}{d-1}$ bits (think of how to assign n apples to d children), and 2) the index of this string $\boldsymbol{x}$ in the list of all possible strings with n_0 0s, ... ,and n_{d-1} $(d-1)$s, which needs $\log \binom{n}{n_0 \cdots n_{d-1}}$. Thus, we need

$$L(n; n_0, \ldots, n_{d-1}) = \log \binom{n+d-1}{d-1} + \log \binom{n}{n_0 \cdots n_{d-1}} = \log \frac{(n+d-1)!}{(d-1)!\,n_0! \cdots n_{d-1}!}. \tag{3.5}$$

Let us now turn to real-valued parameters:

Example 3.6. *Codelength of a truncated real-valued parameter.*
For a deterministic real-valued parameter $v \in \mathbb{R}$, the exact code generally requires infinite

length of bits. Thus, in practice, some truncation must be done for transmission. Let δ be the precision and v_δ be the truncated value, i.e., $|v - v_\delta| < \delta$. Then, the number of bits required for v_δ is the sum of the codelength of its integer part $[v]$ and the number of fractional binary digits of the truncation precision δ, i.e.,

$$L(v_\delta) = L^*([v]) + \log(1/\delta). \tag{3.6}$$

Having gone through the above examples, now we can state the MDL principle more clearly. Let $\mathcal{M} = \{\boldsymbol{\theta}_m : m = 1, 2, \dots\}$ be a class or collection of models at hand. The integer m is simply an index of a model in the list. Let $\boldsymbol{x}$ be a sequence of observed data. Assume that we do not know the true model $\boldsymbol{\theta}$ generating the data $\boldsymbol{x}$. As in [128], [110], given the index m, we can write the codelength for the whole process as

$$L(\boldsymbol{x}, \boldsymbol{\theta}_m, m) = L(m) + L(\boldsymbol{\theta}_m \mid m) + L(\boldsymbol{x} \mid \boldsymbol{\theta}_m, m). \tag{3.7}$$

This equation says that the codelength to rewrite the data is the sum of the codelengths to describe: (i) the index m, (ii) the model $\boldsymbol{\theta}_m$ given m, and (iii) the data $\boldsymbol{x}$ using the model $\boldsymbol{\theta}_m$. The MDL criterion suggests picking the model $\boldsymbol{\theta}_{m^*}$ which gives the minimum of the total description length (3.7).

The last term of the right-hand side (RHS) of (3.7) is the length of the Shannon code of the data assuming the model $\boldsymbol{\theta}_m$ is the true model, i.e.,

$$L(\boldsymbol{x} \mid \boldsymbol{\theta}_m, m) = -\log p(\boldsymbol{x} \mid \boldsymbol{\theta}_m, m), \tag{3.8}$$

and the maximum likelihood (ML) estimate $\widehat{\boldsymbol{\theta}}_m$ minimizes (3.8) by the definition:

$$L(\boldsymbol{x} \mid \widehat{\boldsymbol{\theta}}_m, m) = -\log p(\boldsymbol{x} \mid \widehat{\boldsymbol{\theta}}_m, m) \leq -\log p(\boldsymbol{x} \mid \boldsymbol{\theta}_m, m). \tag{3.9}$$

We should consider a further truncation of $\widehat{\boldsymbol{\theta}}_m$ as shown in Example 3.6 above to check that additional savings in the description length is possible. The finer truncation precision we use, the smaller the term (3.9), but the larger the term $L(\widehat{\boldsymbol{\theta}}_m \mid m)$ becomes. Suppose

that the model $\boldsymbol{\theta}_m$ has k_m real-valued parameters, i.e., $\boldsymbol{\theta}_m = (\theta_{m,1}, \ldots, \theta_{m,k_m})$. Rissanen showed in [126], [128] that the optimized truncation precision (δ^*) is of order $1/\sqrt{n}$ and

$$\begin{aligned}
&\min_{\delta} L(\boldsymbol{x}, \boldsymbol{\theta}_{m,\delta}, m, \delta) \\
&\quad = \quad L(m) + L(\widehat{\boldsymbol{\theta}}_{m,\delta^*} \mid m) + L(\boldsymbol{x} \mid \widehat{\boldsymbol{\theta}}_{m,\delta^*}, m) + O(k_m) \\
&\quad \approx \quad L(m) + \sum_{j=1}^{k_m} L^*([\widehat{\theta}_{m,j}]) + \frac{k_m}{2} \log n + L(\boldsymbol{x} \mid \widehat{\boldsymbol{\theta}}_m, m) + O(k_m), \qquad (3.10)
\end{aligned}$$

where $\widehat{\boldsymbol{\theta}}_m$ is the optimal non-truncated value given m, $\widehat{\boldsymbol{\theta}}_{m,\delta^*}$ is its optimally truncated version, and $L^*(\cdot)$ is defined in (3.6). We note that the last term $O(k_m)$ in the approximation in (3.10) includes the penalty codelength necessary to describe the data $\boldsymbol{x}$ using the truncated ML estimate $\widehat{\boldsymbol{\theta}}_{m,\delta^*}$ instead of the true ML estimate $\widehat{\boldsymbol{\theta}}_m$. In practice, we rarely need to obtain the optimally truncated value $\widehat{\boldsymbol{\theta}}_{m,\delta^*}$ and we should compute $\widehat{\boldsymbol{\theta}}_m$ up to the machine precision, say, 10^{-15}, and use that value as the true ML estimate in (3.10). For sufficiently large n, the last term may be omitted, and instead of minimizing the ideal codelength (3.7), Rissanen proposed to minimize

$$MDL(\boldsymbol{x}, \widehat{\boldsymbol{\theta}}_m, m) = L(m) + \sum_{j=1}^{k_m} L^*([\widehat{\theta}_{m,j}]) + \frac{k_m}{2} \log n + L(\boldsymbol{x} \mid \widehat{\boldsymbol{\theta}}_m, m). \qquad (3.11)$$

The minimum of (3.11) gives the best compromise between the low complexity in the model and high likelihood on the data.

The first term of the RHS of (3.11) can be written as

$$L(m) = -\log p(m), \qquad (3.12)$$

where $p(m)$ is the probability of selecting m. If there is prior information about m as to which m is more likely, we should reflect this in $p(m)$. Otherwise, we assume each m is equally likely, i.e., $p(m)$ is a uniform distribution.

Remark 3.7. Even though the list of models $\mathcal{M}$ does not include the true model, the MDL

method achieves the best result among the available models. See Barron and Cover [7] for detailed information on the error between the MDL estimate and the true model.

We also would like to note that the MDL principle does not attempt to find the absolutely minimum description of the data. The MDL always requires an available collection of models and simply suggests picking the best model from that collection. In other words, the MDL can be considered as an "oracle" for model selection [110]. This contrasts with the algorithmic complexities such as the Kolmogorov complexity which gives the absolutely minimum description of the data; however, in general, it is impossible to obtain [126].

Before deriving our simultaneous noise suppression and signal compression algorithm in the context of the MDL criterion, let us give a closely related example:

Example 3.8. *A curve fitting problem using polynomials.*
Given n points of data $(t_i, y_i) \in \mathbb{R}^2$, consider the problem of fitting a polynomial through these points. The model class we consider is a set of polynomials of orders $0, 1, \ldots, n-1$. In this case, $\boldsymbol{\theta}_m = (a_0, a_1, \ldots, a_m)$ represents the $m+1$ coefficients of a polynomial of order m. We also assume that the data is contaminated by the additive WGN with known variance σ^2, i.e.,

$$y_i = x(t_i) + \epsilon_i,$$

where $x(\cdot)$ is an unknown function to be estimated by the polynomial models, and $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. To invoke the MDL formalism, we pose this question in the information transmission setting. First we prepare an encoder which computes the ML estimate of the coefficients of the polynomial, $(\widehat{a}_0, \ldots, \widehat{a}_m)$, of the given degree m from the data. (In the additive WGN assumption the ML estimate coincides with the least squares estimate.) This encoder transmits these m coefficients as well as the estimation errors. We also prepare a decoder which receives the coefficients of the polynomial and residual errors and reconstruct the data. (We assume that the abscissas $\{t_i\}_{i=1}^n$ and the noise variance σ^2 are known to both the encoder and the decoder.) Then we ask how many bits of information should be

transmitted to reconstruct the data. If we used polynomials of degree $n-1$, we could find a polynomial passing through all n points. In this case, we could describe the data extremely well. In fact, there is no error between the observed data and those reconstructed by the decoder; however, we do not gain anything in terms of data compression/transmission since we also have to encode the model which requires n coefficients of the polynomial. In some sense, we did not "learn" anything in this case. If we used the polynomial of degree 0, i.e., a constant, then it would be an extremely efficient model, but we would need many bits to describe the deviations from that constant. (Of course, if the underlying data is really a constant, then the deviation would be 0.)

Let us assume there is no prior preference on the order m. Then we can easily see that the total codelength (3.11) in this case becomes

$$\begin{aligned} MDL(\boldsymbol{y}, \widehat{\boldsymbol{\theta}}_m, m) &= \log n + \sum_{j=0}^{m} L^*([\widehat{a}_j]) + \frac{m+1}{2} \log n \\ &\quad + \frac{n}{2} \log 2\pi\sigma^2 + \frac{\log \mathrm{e}}{2\sigma^2} \sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{m} \widehat{a}_j {x_i}^j \right)^2 . \end{aligned}$$

The MDL criterion suggests picking the "best" polynomial of order m^* by minimizing this approximate codelength.

The MDL criterion has been successfully used in various fields such as signal detection [153], image segmentation [88], and cluster analysis [149] where the optimal number of signals, regions, and clusters, respectively, should be determined. If one knows *a priori* the physical model to explain the observed data, that model should definitely be used, e.g., the complex sinusoids in [153]. In general, however, as a descriptor of real-life signals which are full of transients or edges, the library of wavelets, wavelet packets, and local trigonometric transforms is more flexible and efficient than the set of polynomials or sinusoids.

3.4 A Simultaneous Noise Suppression and Signal Compression Algorithm

We carry on our development of the algorithm based on the information transmission setting as the polynomial curve fitting problem described in the previous section. We consider again an encoder and a decoder for our problem. Given (k, m) in (3.1), the encoder expands the data $\boldsymbol{y}$ in the basis $\mathfrak{B}_m$, then transmits the number of terms k, the specification of the basis m, and k expansion coefficients, the variance of the WGN model σ^2, and finally the estimation errors. The decoder receives this information in bits and tries to reconstruct the data $\boldsymbol{y}$.

In this case, the total codelength to be minimized may be expressed as the sum of the codelengths of: (i) two natural numbers (k, m), (ii) $(k+1)$ real-valued parameters $(\boldsymbol{\alpha}_m^{(k)}, \sigma^2)$ given (k, m), and (iii) the deviations of the observed data $\boldsymbol{y}$ from the (estimated) signal $\boldsymbol{x} = \boldsymbol{W}_m \boldsymbol{\alpha}_m^{(k)}$ given $(k, m, \boldsymbol{\alpha}_m^{(k)}, \sigma^2)$. The approximate total description length (3.11) now becomes

$$\begin{aligned} & MDL(\boldsymbol{y}, \widehat{\boldsymbol{\alpha}}_m^{(k)}, \widehat{\sigma}^2, k, m) \\ &= \quad L(k, m) + L(\widehat{\boldsymbol{\alpha}}_m^{(k)}, \widehat{\sigma}^2 \mid k, m) + L(\boldsymbol{y} \mid \widehat{\boldsymbol{\alpha}}_m^{(k)}, \widehat{\sigma}^2, k, m), \end{aligned} \tag{3.13}$$

where $\widehat{\boldsymbol{\alpha}}_m^{(k)}$ and $\widehat{\sigma}^2$ are the ML estimates of $\boldsymbol{\alpha}_m^{(k)}$ and σ^2, respectively.

Let us now derive these ML estimates. Since we assumed the noise component is additive WGN, the probability of observing the data given all model parameters is

$$P(\boldsymbol{y} \mid \boldsymbol{\alpha}_m^{(k)}, \sigma^2, k, m) = (2\pi\sigma^2)^{-n/2} \exp\left(-\frac{\|\boldsymbol{y} - \boldsymbol{W}_m \boldsymbol{\alpha}_m^{(k)}\|^2}{2\sigma^2} \right), \tag{3.14}$$

where $\|\cdot\|$ is the standard Euclidean norm on $\mathbb{R}^n$. For the ML estimate of σ^2, first consider the log-likelihood of (3.14)

$$\ln p(\boldsymbol{y} \mid \boldsymbol{\alpha}_m^{(k)}, \sigma^2, k, m) = -\frac{n}{2} \ln 2\pi\sigma^2 - \frac{\|\boldsymbol{y} - \boldsymbol{W}_m \boldsymbol{\alpha}_m^{(k)}\|^2}{2\sigma^2}. \tag{3.15}$$

Taking the derivative with respect to σ^2 and setting it to zero, we easily obtain

$$\widehat{\sigma}^2 = \frac{1}{n}\|\boldsymbol{y} - \boldsymbol{W}_m \boldsymbol{\alpha}_m^{(k)}\|^2. \tag{3.16}$$

Insert this equation back to (3.15) to get

$$\ln p(\boldsymbol{y} \mid \boldsymbol{\alpha}_m^{(k)}, \widehat{\sigma}^2, k, m) = -\frac{n}{2}\ln\left(\frac{2\pi}{n}\|\boldsymbol{y} - \boldsymbol{W}_m \boldsymbol{\alpha}_m^{(k)}\|^2\right) - \frac{n}{2}. \tag{3.17}$$

Let $\widetilde{\boldsymbol{y}}_m = \boldsymbol{W}_m^T \boldsymbol{y}$ denote the vector of the expansion coefficients of $\boldsymbol{y}$ in the basis $\mathfrak{B}_m$. Since this basis is orthonormal, i.e., $\boldsymbol{W}_m$ is orthogonal, and we use the ℓ^2 norm, we have

$$\|\boldsymbol{y} - \boldsymbol{W}_m \boldsymbol{\alpha}_m^{(k)}\|^2 = \|\boldsymbol{W}_m(\boldsymbol{W}_m^T \boldsymbol{y} - \boldsymbol{\alpha}_m^{(k)})\|^2 = \|\widetilde{\boldsymbol{y}}_m - \boldsymbol{\alpha}_m^{(k)}\|^2. \tag{3.18}$$

From (3.17), (3.18), and the monotonicity of the ln function, we find that maximizing (3.17) is equivalent to minimizing

$$\|\widetilde{\boldsymbol{y}}_m - \boldsymbol{\alpha}_m^{(k)}\|^2. \tag{3.19}$$

Considering that the vector $\boldsymbol{\alpha}_m^{(k)}$ only contains k nonzero elements, we can easily conclude that the minimum of (3.19) is achieved by taking the largest k coefficients in magnitudes of $\widetilde{\boldsymbol{y}}_m$ as the ML estimate of $\boldsymbol{\alpha}_m^{(k)}$, i.e.,

$$\widehat{\boldsymbol{\alpha}}_m^{(k)} = \boldsymbol{\Theta}^{(k)} \widetilde{\boldsymbol{y}}_m = \boldsymbol{\Theta}^{(k)}(\boldsymbol{W}_m^T \boldsymbol{y}), \tag{3.20}$$

where $\boldsymbol{\Theta}^{(k)}$ is a thresholding operation which keeps the k largest elements in absolute value intact and sets all other elements to zero. Finally, inserting (3.20) into (3.16), we obtain

$$\widehat{\sigma}^2 = \frac{1}{n}\|\boldsymbol{W}_m^T \boldsymbol{y} - \boldsymbol{\Theta}^{(k)} \boldsymbol{W}_m^T \boldsymbol{y}\|^2 = \frac{1}{n}\|(\boldsymbol{I} - \boldsymbol{\Theta}^{(k)})\boldsymbol{W}_m^T \boldsymbol{y}\|^2, \tag{3.21}$$

where $\boldsymbol{I}$ represents the n dimensional identity operator (matrix).

Let us further analyze (3.13) term by term. If we do not have any prior information on (k, m), then the cost $L(k, m)$ is the same for all cases, i.e., we can drop the first term of (3.13) for minimization purpose. However, if one has some preference about the choice of

basis by some prior information about the signal $\boldsymbol{x}$, $L(k,m)$ should reflect this information. For instance, if we happen to know that the original function $\boldsymbol{x}$ consists of a linear combination of dyadic blocks, then we clearly should use the Haar basis. In this case, we may use the Dirac distribution, i.e., $p(m) = \delta_{m,m_0}$, where m_0 is the index for the Haar basis in the library $\mathfrak{L}$. By (3.12), this leads to

$$L(k,m) = \begin{cases} L(k) & \text{if } m = m_0, \\ +\infty & \text{otherwise.} \end{cases}$$

On the other hand, if we either happen to know *a priori* or want to force the number of terms retained (k) to satisfy $k_1 \le k \le k_2$, then we may want to assume the uniform distribution for this range of k, i.e.,

$$L(k,m) = \begin{cases} L(m) + \log(k_2 - k_1 + 1) & \text{if } k_1 \le k \le k_2, \\ +\infty & \text{otherwise.} \end{cases} \tag{3.22}$$

As for the second term of (3.13), which is critical for our algorithm, we have to encode k expansion coefficients $\widehat{\boldsymbol{\alpha}}_m^{(k)}$ and $\widehat{\sigma}^2$, i.e., $(k+1)$ real-valued parameters. In this case, however, by normalizing the whole sequence by $\|\boldsymbol{y}\|$, we can safely assume that the magnitude of each coefficient in $\widehat{\boldsymbol{\alpha}}^{(k)}$ is strictly less than one; in other words, the integer part of each coefficient is simply zero. Hence we do not need to encode the integer part as in (3.11) if we transmit the real-valued parameter $\|\boldsymbol{y}\|$. Now the description length of $(\widehat{\boldsymbol{\alpha}}_m^{(k)}, \widehat{\sigma}^2)$ given (k,m) becomes approximately $\frac{k+2}{2}\log n + L^*([\widehat{\sigma}^2]) + L^*([\|\boldsymbol{y}\|])$ bits since there are $k+2$ real-valued parameters: k nonzero coefficients, $\widehat{\sigma}^2$, and $\|\boldsymbol{y}\|$. After normalizing by $\|\boldsymbol{y}\|$, we clearly have $\widehat{\sigma}^2 < 1$ (see (3.21)), so that $L^*([\widehat{\sigma}^2]) = 1$ (see (3.3)). For each expansion coefficient, however, we still need to specify the index of the coefficient, i.e., where the k non-zero elements are in the vector $\widehat{\boldsymbol{\alpha}}_m^{(k)}$. This requires $k \log n$ bits. As a result, we have

$$L(\widehat{\boldsymbol{\alpha}}_m^{(k)}, \widehat{\sigma}^2 \mid k, m) = \frac{3}{2} k \log n + c, \tag{3.23}$$

where c is a constant independent of (k, m).

Since the probability of observing $\boldsymbol{y}$ given all model parameters is given by (3.14), we have for the last term in (3.13)

$$L(\boldsymbol{y} \mid \widehat{\boldsymbol{\alpha}}_m^{(k)}, \widehat{\sigma}^2, k, m) = \frac{n}{2} \log \|(\boldsymbol{I} - \boldsymbol{\Theta}^{(k)}) \boldsymbol{W}_m^T \boldsymbol{y}\|^2 + c', \tag{3.24}$$

where c' is a constant independent of (k, m).

Finally we can state our simultaneous noise suppression and signal compression algorithm. Let us assume that we do not have any prior information on (k, m) for now. Then, from (3.13), (3.23), and (3.24) with ignoring the constant terms c and c', our algorithm can be stated as:

Pick the index (k^*, m^*) *such that*

$$AMDL(k^*, m^*) = \min_{\substack{0 \le k < n \\ 1 \le m \le M}} \left(\frac{3}{2} k \log n + \frac{n}{2} \log \|(\boldsymbol{I} - \boldsymbol{\Theta}^{(k)}) \boldsymbol{W}_m^T \boldsymbol{y}\|^2 \right). \tag{3.25}$$

Then reconstruct the signal estimate

$$\widehat{\boldsymbol{x}} = \boldsymbol{W}_{m^*} \boldsymbol{\alpha}_{m^*}^{(k^*)}. \tag{3.26}$$

Let us call the objective function to be minimized in (3.25), the approximate MDL (AMDL) since we ignored the constant terms. Let us now show a typical behavior of the AMDL value as a function of the number of terms retained (k) in Figure 3.1. (In fact, this curve is generated using Example 3.9 below.) We see that the log(residual energy) always decreases as k increases. By adding the penalty term of retaining the expansion coefficients, i.e., $(3/2)k \log n$ (which is just a straight line), we have the AMDL curve which typically decreases for the small k, then starts increasing because of the penalty term, then finally decreases again at some large k near from $k = n$ because the residual error becomes very small. Now what we really want is the value of k achieving the minimum at the beginning of the k-axis, and we want to avoid searching for k beyond the maximum occurring for k near n. So, we can safely assume that $k_1 = 0$ and $k_2 = n/2$ in (3.22) to avoid searching

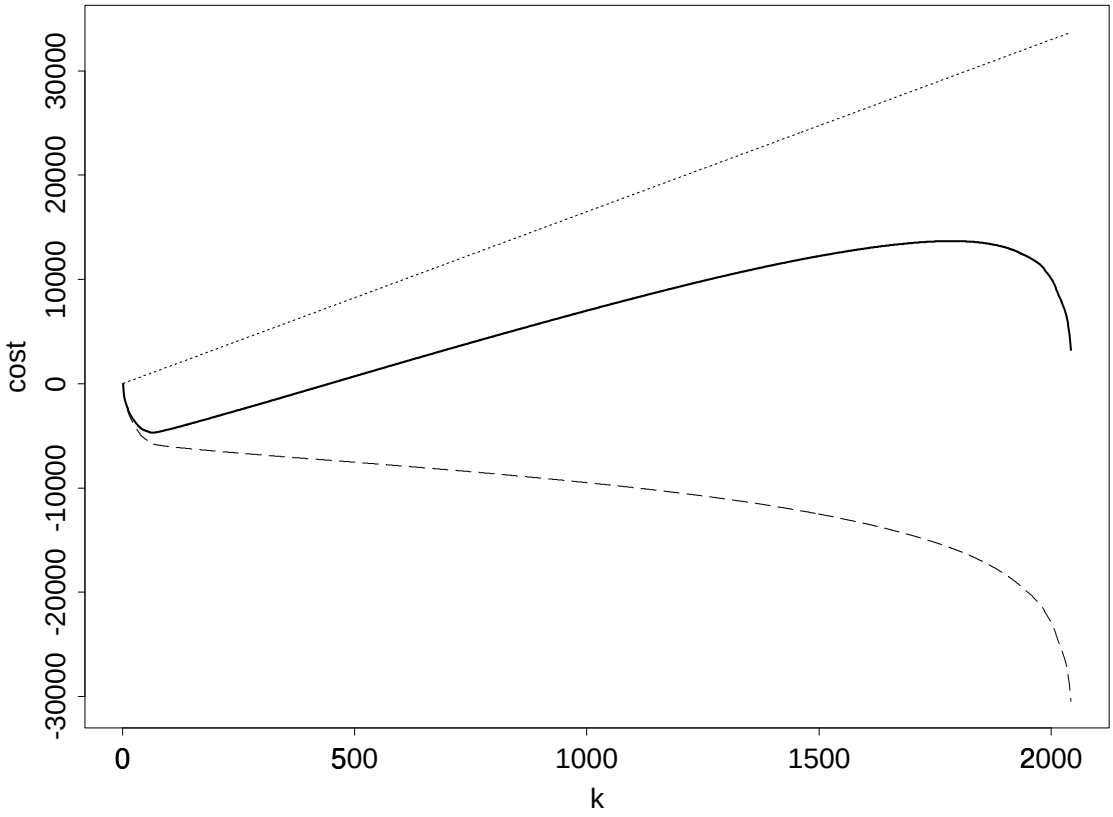


Figure 3.1: Graphs of AMDL versus k: AMDL [solid line] which is the sum of the $(3/2)k \log n$ term [dotted line] and the $(n/2)$ log(residual energy) term [dashed line].

more than necessary. (In fact, setting $k_2 > n/2$ does not make much sense in terms of data compression either.)

We briefly examine below the computational complexity of our algorithm. To obtain (k^*, m^*), we proceed as follows:

Step 1: Expand the data $\boldsymbol{y}$ into bases $\mathfrak{B}_1, \ldots, \mathfrak{B}_M$. Each expansion (including the best-basis selection procedure) costs about $O(n)$ for wavelets, $O(n \log n)$ for wavelet packet best bases, and $O(n[\log n]^2)$ for local trigonometric best bases.

Step2: For $k_1 \le k \le k_2$, $1 \le m \le M$, compute the expression in the parenthesis of the RHS in (3.25). This costs approximately $O(n + 3MK)$ multiplications and MK calls to the log function where $K(= k_2 - k_1 + 1)$ denote the length of the search range for k.

Step 3: Search the minimum entry in this table, which costs MK comparisons.

Step 4: Reconstruct the signal estimate (3.26), which costs $O(n)$ for wavelets, $O(n \log n)$ for wavelet packet best bases, and $O(n[\log n]^2)$ for local trigonometric best bases.

For images or multidimensional signals, we can easily extend our algorithm using the multidimensional version of the wavelets, wavelet packets, and local trigonometric transforms reviewed in Chapter 2. We can put as many orthonormal bases into the library as we can afford it in terms of computational resources. In particular, we can put different versions of 2D wavelets (e.g., two different versions of the tensor-product-based wavelets [13], [43], nonseparable wavelets [65], [84]) in the library so that there will be no issue such as which version should be used or not. Minimizing the AMDL values automatically selects the most suitable one for our purpose.

3.5 Examples

In this section, we give several examples to show the usefulness of our algorithm.

Example 3.9. *The Synthetic Piecewise Constant Function of Donoho-Johnstone.*
We compared the performance of our algorithm in terms of the visual quality of the estimation and the relative ℓ^2 error with Donoho-Johnstone's method using the piecewise constant function used in their experiments [53]. The results are shown in Figure 3.2. The true signal is the piecewise constant function with $n = 2048$, and its noisy observation was created by adding the WGN sequence with $\|\boldsymbol{x}\|/\|\boldsymbol{\epsilon}\| = 7$. The library $\mathfrak{L}$ for this example

consisted of 18 different bases: the standard Euclidean basis of $\mathbb{R}^n$, the wavelet packet best bases created with D02, D04, ..., D20, C06, C12, ..., C30, and the local cosine and sine best bases (Dm represents the m-tap QMF of Daubechies and Cm represents the m-tap coiflet filter). In the Donoho-Johnstone method, we used the C06, i.e., 6-tap coiflet with 2 vanishing moments. We also specified the scale parameter $J = 7$, and supplied the *exact* value of σ^2. Next, we *forced* the Haar basis (D02) to be used in their method. Finally, we applied our algorithm without specifying anything. In this case, the Haar-Walsh best basis with $k^* = 63$ was automatically selected. The relative ℓ^2 errors are 0.116, 0.089, 0.051, respectively. Although the visual quality of our result is not too different from Donoho and Johnstone's (if we *choose* the appropriate basis for their method), our method generated the estimate with the smallest relative ℓ^2 error and slightly sharper edges. (See Section 3.6 for more about the Donoho-Johnstone method and its relation to our method.)

Example 3.10. *A Pure White Gaussian Noise.*
We generated a synthetic sequence of WGN with $\sigma^2 = 1.0$ and $n = 4096$. The same library as in Example 3.9 (with the best bases adapted to this pure WGN sequence) was used. We also set the upper limit of search range $k_2 = n/2 = 2048$. Figure 3.3 shows the AMDL curves versus k for all bases in the library. As we can see, there is no single minimum in the graphs, and our algorithm satisfactorily decided $k^* = 0$, i.e., there is nothing to "learn" in this dataset.

Example 3.11. *A Natural Radioactivity Profile of Subsurface Formation.*
We tested our algorithm on the actual field data which are measurement of natural radioactivity of subsurface formations obtained at an oil-producing well. The length of the data is $n = 1024$. Again, the same library was used as in the previous examples. The results are shown in Figure 3.4. In this case, our algorithm selected the D12 wavelet packet best basis (Daubechies's 12-tap filter with 6 vanishing moments) with $k^* = 77$. The residual error is shown in Figure 3.4 (c) which consists mostly of a WGN-like high frequency component.

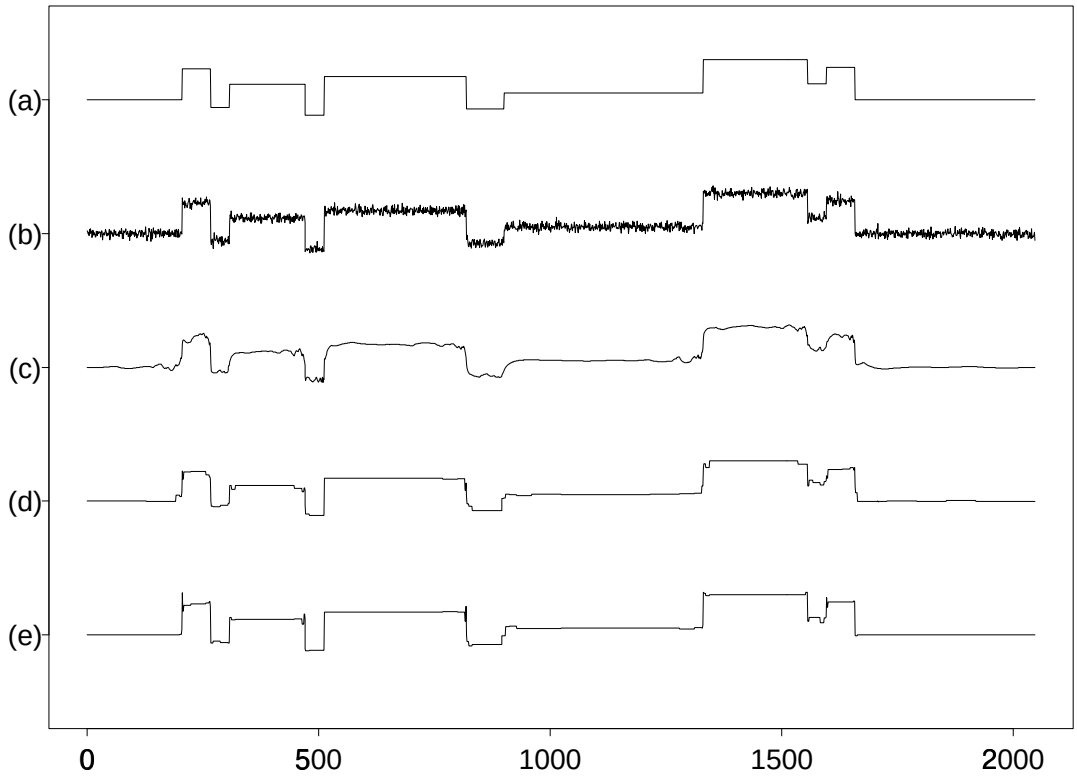


Figure 3.2: Results for the synthetic piecewise constant function: (a) Original piecewise constant function. (b) Noisy observation with (signal energy)/(noise energy) = 7^2. (c) Estimation by the Donoho-Johnstone method using coiflets C06. (d) Estimation by the Donoho-Johnstone method using Haar basis. (e) Estimation by the proposed method.

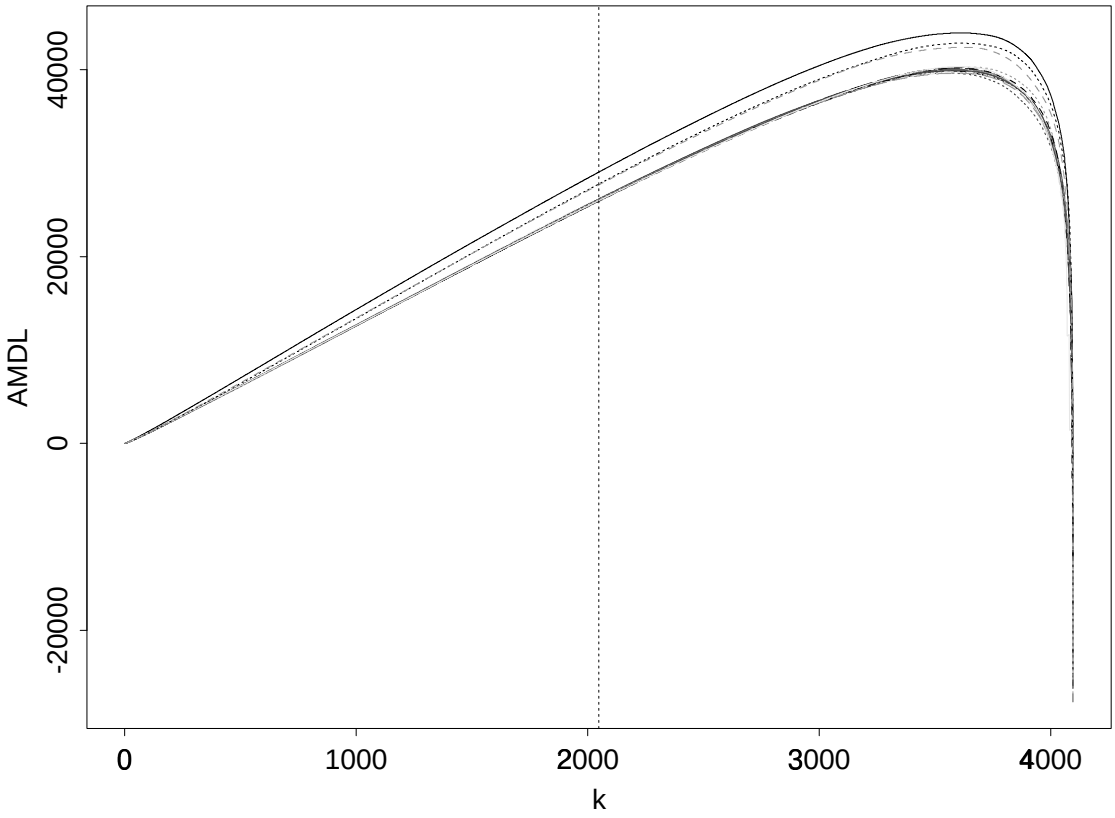


Figure 3.3: The AMDL curves of the White Gaussian Noise data for all bases. For each basis, $k = 0$ is the minimum value. The vertical dotted line indicates the upper limit of the search range for k.

The compression ratio is $1024/77 \approx 13.3$. To be able to reconstruct the signal from the surviving coefficients, we still need to record the indices of those coefficients.

Suppose we can store each index by b_i bytes of memory and the precision of the original data is b_f bytes per sample. Then the *storage reduction ratio* R_s can be computed by

$$R_s = \frac{n/r \times (b_f + b_i)}{n \times b_f} = \frac{1}{r}(1 + \frac{b_i}{b_f}), \tag{3.27}$$

where r is a compression ratio. The original data precision was $b_f = 8$ (bytes) in this case. Since it is enough to use $b_i = 2$ (bytes) for indices and $r = 13.3\%$, we have $R_s \approx 9.40\%$, i.e., 90.60% of the original data can be discarded.

Example 3.12. *A Migrated Seismic Section.*
In this example, the data is a migrated seismic section as shown in Figure 3.5 (a). The data consist of 128 traces of 256 time samples. We selected six 2D wavelet packet best bases (D02, C06, C12, C18, C24, C30) as the library. Figure 3.5 (b) shows the estimate by our algorithm. It automatically selected the filter C30 and the number of terms retained as $k^* = 1611$. If we were to choose a good threshold in this example, it would be fairly difficult since we do not know the accurate estimate of σ^2. The compression rate, in this case, is $(128 \times 256)/1611 \approx 20.34$. The original data precision was $b_f = 8$ as in the previous example. In this case we have to use $b_i = 3$ (1 byte for row index, 1 byte for column index, and 1 byte for scale level). If we put these and $r = 20.34\%$ into (3.27), we have $R_s \approx 6.76\%$, i.e., 93.24% of the original data can be discarded. Figure 3.5 (c) shows the residual error between the original and the estimate. We can clearly see the random noise and some strange high frequency patterns (which are considered to be numerical artifacts from the migration algorithm applied).

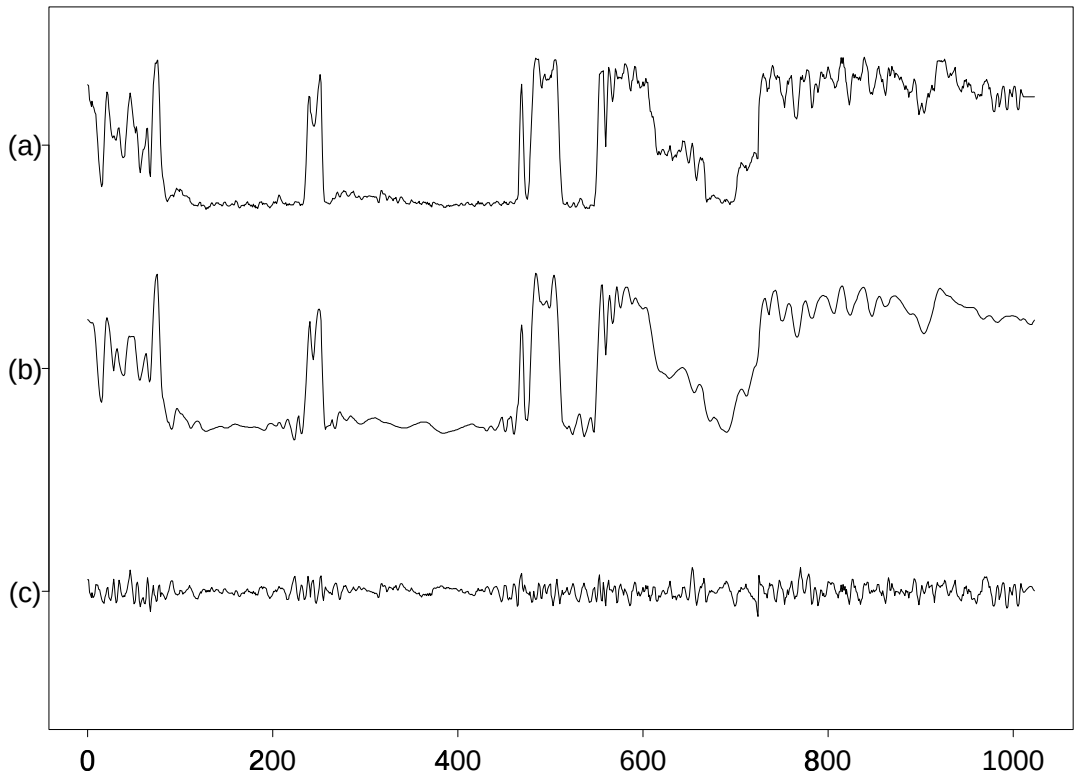


Figure 3.4: The estimate of the natural radioactivity profile of subsurface formations: (a) Original data which was measured in the borehole of an oil-producing well. (b) Estimation by the proposed method. (c) Residual error between (a) and (b).

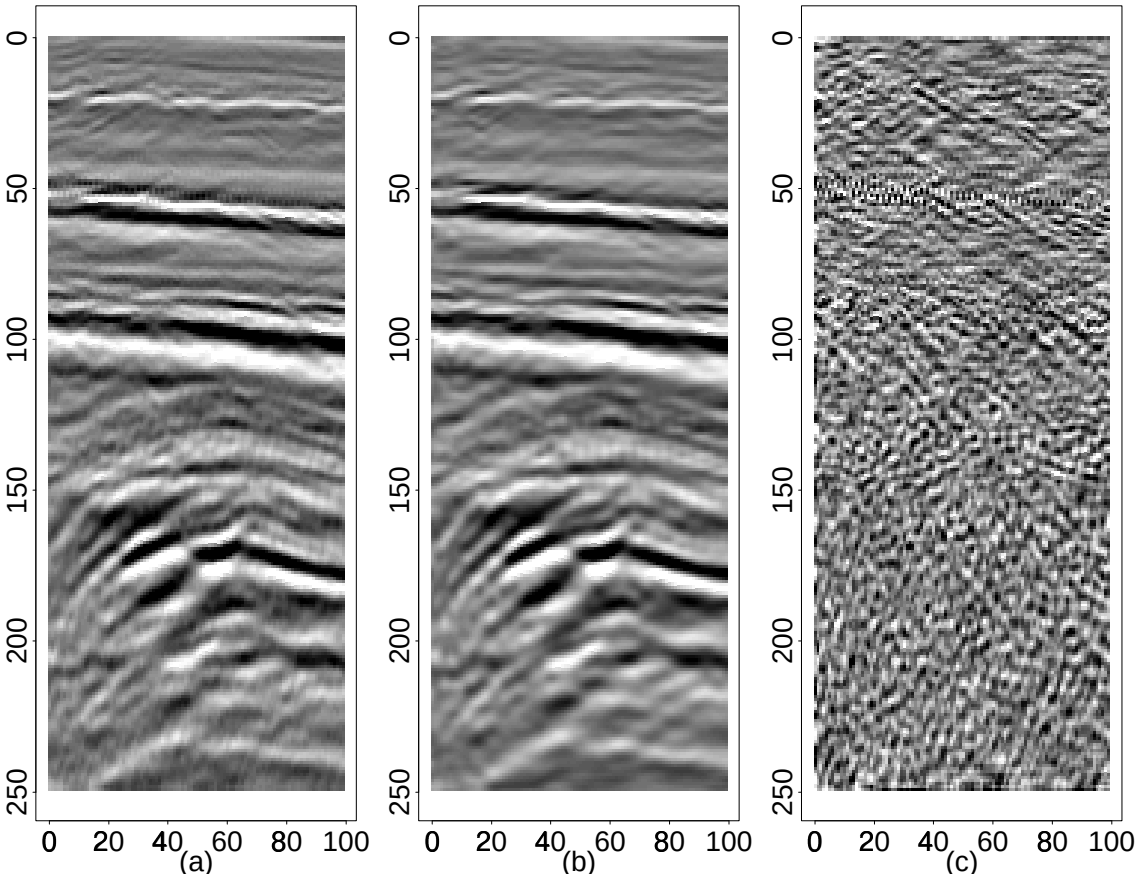


Figure 3.5: Results for the migrated seismic section: (a) Original seismic section with 128 traces and 256 time samples. (b) Estimation by the proposed method. (c) Residual error between (a) and (b). (Dynamic range of display (c) is different from those of (a) and (b).)

3.6 Discussion

Our algorithm is intimately connected to the "denoising" algorithm of Coifman and Majid [30], [36]. Their algorithm first picks the best basis from the collection of bases and sorts the best-basis coefficients in order of decreasing magnitude. Then they use the "theoretical compression rate" of the sorted best-basis coefficients $\{\alpha_i\}_{i=1}^n$ as a key criterion for separating a signal component from noise. The theoretical compression rate of a unit vector $\boldsymbol{u}$ is defined as $c(\boldsymbol{u}) = 2^{H(\boldsymbol{u})}/n(\boldsymbol{u})$, where $H(\boldsymbol{u})$ is the ℓ^2-entropy of $\boldsymbol{u}$, i.e., $H(\boldsymbol{u}) = -\sum_{i=1}^{n(\boldsymbol{u})} u_i^2 \log u_i^2$, and $n(\boldsymbol{u})$ is the length of $\boldsymbol{u}$. We note that $0 \leq c(\boldsymbol{u}) \leq 1$ for any real unit vector $\boldsymbol{u}$, and $c(\boldsymbol{u}) = 0$ implies $\boldsymbol{u} = \{\delta_{i,i_0}\}$ for some i_0 (the best possible compression), and $c(\boldsymbol{u}) = 1$ implies $\boldsymbol{u} = (1, \ldots, 1)/\sqrt{n(\boldsymbol{u})}$ (the worst compression). Then to decide how many coefficients to keep as a signal component, they compare $c(\{\alpha_i\}_{i=k+1}^n)$, the theoretical compression rate of the noise component (defined as the smallest $(n-k)$ coefficients), to the predetermined threshold τ. They search $k = 0, 1, \ldots$ which gives an unacceptably bad compression rate: $c(\{\alpha_i\}_{i=k+1}^n) \geq \tau$. Their algorithm critically depends on the choice of the threshold τ whereas our algorithm needs no threshold selection. On the other hand, their algorithm does not assume the WGN model we used in this chapter; rather, they *defined* the noise component as a vector reconstructed from the best-basis coefficients of small magnitude.

Recently, Coifman discovered a way to improve the denoising algorithms. This method utilizes the fact that the expansion coefficients in the library is not shift invariant. This is certainly an undesirable property of the library; however, the key observation is that the amount of variation on the coefficients caused by the shift is less emphasized in the signal component than in the noise component since the former is normally smoother than the latter. Based on this observation, he proposed to apply his denoising algorithm to several translated versions of the original signal then take the average of these denoised signals after undoing the translations. We applied this method with our MDL-based denoising algorithm to the signal of Example 3.11. Figure 3.6 shows the result of this "shift-denoise-

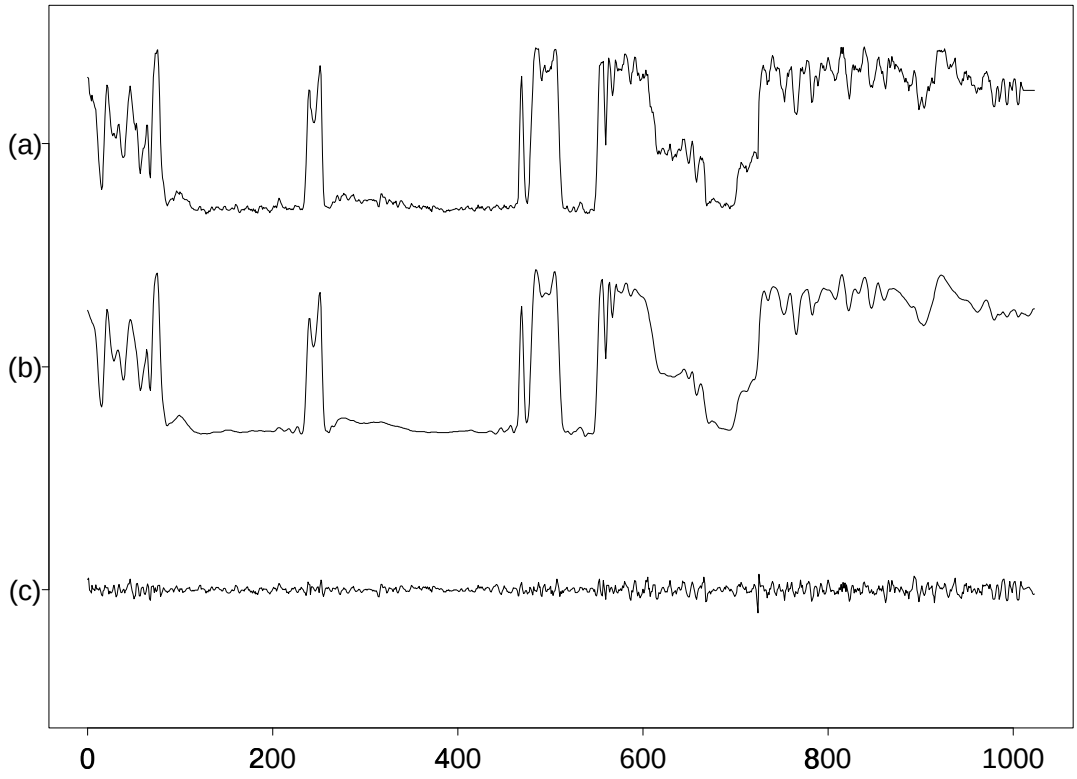


Figure 3.6: The result of the "shift-denoise-average" method using the signal of Example 3.11. (a) Original data. (b) Estimation by the "shift-denoise-average" method using the MDL-based denoising algorithm with 11 shifts. (c) Residual error between (a) and (b).

average" algorithm. Compare this with Figure 3.4. Eleven different versions with shifts $0, \pm 1, \ldots, \pm 5$ were used. We observe that the Gibbs-like phenomena around the edges are less emphasized in this method and the residual error becomes much closer to WGN than in the one step MDL-denoising result in Figure 3.4. An interesting issue here is to examine whether this "shift-denoise-average" process can be formulated in the MDL formalism and whether the optimal shift parameters can be selected automatically or not.

Our algorithm can also be viewed as a simple yet flexible and efficient realization of

the "complexity regularization" method for estimation of functions proposed by Barron [6]. He considered a general regression function estimation problem: given the data $(t_i, y_i)_{i=1}^n$, where $\{t_i \in \mathbb{R}^p\}$ is a sequence of the (p-dimensional) sampling coordinates (or explanatory variables) and $\{y_i \in \mathbb{R}\}$ is the observed data (or response variables), select a "best" regression function $\widehat{x}_n$ out of a list (library) $\mathfrak{L}_n$ of candidate functions (models). He did not impose any assumption on the noise distribution, but assumed that the number of models in the list $\mathfrak{L}_n$ depends on the number of observations n. Now the complexity regularization method of Barron is to find $\widehat{x}_n$ such that

$$R(\widehat{x}_n) = \min_{x \in \mathfrak{L}_n} \left(\frac{1}{n} \sum_{i=1}^{n} \delta(y_i, x(t_i)) + \frac{\lambda}{n} L(x) \right),$$

where $\delta(\cdot, \cdot)$ is a measure of distortion (such as the squared error), $\lambda > 0$ is a regularization constant, and $L(x)$ is a complexity of a function x (such as the $L(m) + L(\boldsymbol{\theta}_m \mid m)$ term in (3.7)). He showed that various asymptotic properties of the estimator $\widehat{x}_n$ as $n \to \infty$, such as bounds on the estimation error, the rate of convergence, etc. If we restrict our attention to the finite dimensional vector space, use the library of orthonormal bases described in Chapter 2, adopt the length of the Shannon code (3.8) as a distortion measure, assume the WGN model, and finally set $\lambda = 1$, then Barron's complexity regularization method reduces to our algorithm. Our approach, although restricted in the sense of Barron, provides a computationally efficient and yet flexible realization of the complexity regularization method, especially compared to the library consisting of polynomials, splines, trigonometric series discussed in [6].

Our algorithm also has a close relationship with the denoising algorithm via "wavelet shrinkage" developed by Donoho and Johnstone [53]. (A well-written summary on the wavelet shrinkage and its applications can be found in [52].) Their algorithm first transforms the observed discrete data into a wavelet basis (specified by the user), then applies a "soft threshold" $\tau = \sigma\sqrt{\ln n}$ to the coefficients, i.e., shrinks magnitudes of all the coefficients by the amount τ toward zero. Finally the denoised data is obtained by the inverse wavelet

transform. Donoho claimed informally in [52] that the reason why their method works is the ability of wavelets to compress the signal energy into a few coefficients. The main differences between our algorithm and that of Donoho and Johnstone are:

- Our method automatically selects the most suitable basis from a collection of bases whereas their method uses only a *fixed* basis specified by the user.
- Our method includes adaptive expansion by means of wavelet packets and local trigonometric bases whereas their method only uses a wavelet transform.
- Their method requires the user to set the coarsest scale parameter $J \leq n_0 = \log n$ and a good estimate of σ^2, and the resulting quality depends on these parameters. On the other hand, our method does not require any such parameter setting.
- Their approach is based on the minimax decision theory in statistics and addresses the risk of the estimation whereas our approach uses the information-theoretic idea and combines denoising and the data compression capability of wavelets explicitly.
- Their method thresholds the coefficients *softly* whereas our method can be said to threshold *sharply*. This might cause some Gibbs-like effects in the reconstruction using our method.

Future extensions of this research are to: 1) formulate the Coifman's "shift-denoise-average" method using the MDL principle, 2) incorporate noise models other than Gaussian noise, 3) extend the algorithm for highly nonstationary signals by segmenting them smoothly and adaptively, 4) investigate the effect of sharp thresholding, and 5) study more about the relation with the complexity regularization method of Barron as well as the wavelet shrinkage of Donoho-Johnstone.

3.7 Summary

We have described an algorithm for simultaneously suppressing the additive WGN component and compressing the signal component in data. One or more of the bases in a library of orthonormal bases can compress the signal component quite well whereas the WGN component cannot be compressed efficiently by any basis in the library. Based on this observation, we have derived an algorithm to estimate the signal component in the data by obtaining the "best" basis and the "best" number of terms to retain using the MDL criterion. Because of the use of the MDL criterion, this algorithm does not require the user to specify any parameter or threshold values. Both synthetic and real field data examples have shown the wide applicability and usefulness of this algorithm.

Chapter 4

Local Discriminant Bases and Their Applications

4.1 Introduction

Extracting relevant features from signals or images is an important process for data analysis, such as classifying signals into known categories (*classification*) or predicting a response of interest based on these signals (*regression*). In this chapter, we describe an extension to the "best-basis" method reviewed in Chapter 2 to construct an orthonormal basis suitable for classification problems rather than compression problems. In particular, we propose a fast algorithm to select an efficient basis (or coordinate system) from a library of orthonormal bases to enhance the performance of a few classification schemes. This algorithm reduces the dimensionality of these problems by using basis functions described in Chapter 2 (which are well-localized in the time-frequency plane) as feature extractors. Since this basis illuminates the differences among classes, it can be used to extract signal component from data consisting of signal and textured background.

The organization of this chapter is as follows. In Section 4.2, we formulate the

problem of feature extraction and classification. Then, in Section 4.3, we review some of the pattern classification schemes used in our study. In Section 4.4, we propose a fast algorithm for constructing such a local basis for classification problems. This is immediately followed by examples in Section 4.5. We then examine the effect of the denoising capability of the selected bases to the classification problems and discuss whether we should preprocess the input signals via the denoising algorithm described in Chapter 3. Finally, we discuss a method of signal / "background" separation as a further application of such a basis in Section 4.7.

4.2 Problem Formulation

Let us first define appropriate spaces of input signals (or patterns), extracted features, and outputs (or responses) and mapping functions among them. Let $\mathcal{X} \subset \mathbb{R}^n$ denote a *signal space* (or a *pattern space*) which is a subset of the standard n-dimensional vector space and which contains all signals (or samples, or patterns) under consideration. In this case, the *dimensionality* of the signal space or equivalently the length of each signal is n. Let $\mathcal{Y} = \{1, 2, \ldots, C\}$ be a set of the class or category names corresponding to the input signals. We call this space a *response space*. Signal classification can be considered as a mapping function (usually a many-to-one) $d : \mathcal{X} \to \mathcal{Y}$ between these two spaces. Direct manipulation of signals in the signal space for classification is prohibitive because: 1) the signal space normally has very high dimensionality (e.g., $n \approx 1000$ for a typical exploration seismic record per receiver, and for a typical CT scanner image, $n = 512 \times 512 = 262,144$), and 2) the existence of noise or undesired components (whether random or not) in signals makes classification difficult. On the other hand, the signal space is overly redundant compared to the response space. Therefore, it is extremely important to reduce the dimensionality of the problem, i.e., extract only relevant features for the problem at hand and discard all irrelevant information. If we succeed in doing this, we can greatly improve classification

performance both in its accuracy and efficiency. For this purpose, we set a *feature space* $\mathcal{F} \subset \mathbb{R}^k$ where $k \leq n$ between the signal space and the response space. A *feature extractor* is defined as a map $f : \mathcal{X} \to \mathcal{F}$, and a *predictor* (also called *classifier* for classification) as a map $g : \mathcal{F} \to \mathcal{Y}$. Let $\mathcal{T} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N \subset \mathcal{X} \times \mathcal{Y}$ be a *training* (or *learning*) dataset with N pairs of signals $\boldsymbol{x}_i$ and responses (class names) y_i This is the dataset used to construct a feature extractor f. Let N_c be the number of signals belonging to class c so that we have $N = N_1 + \cdots + N_C$. Also, let us denote a set of class c signals by $\{\boldsymbol{x}_i^{(c)}\}_{i=1}^{N_c} = \{\boldsymbol{x}_i\}_{i \in I_c}$ where $I_c \subset \{1, \ldots, N\}$ is a set of indices for class c signals in the training dataset with $|I_c| = N_c$.

Preferably, the performance of the whole process should be measured by the misclassification rate using a *test* dataset $\mathcal{T}' = \{(y_i', \boldsymbol{x}_i')\}_{i=1}^{N'}$ (which has not been used to construct the feature extractors and classifiers) as $(1/N') \sum_{i=1}^{N'} \delta(y_i' - d(\boldsymbol{x}_i'))$, where $\delta(r \neq 0) = 1$ and $\delta(0) = 0$. If we use the *resubstitution* error rates (i.e., the misclassification rates computed on the training dataset), we obviously have overly optimistic figures.

In this chapter, we focus on the feature extractors of the form $f = \Theta^{(k)} \circ \Psi$, where $\Theta^{(k)} : \mathcal{X} \to \mathcal{F}$ represents the selection rule (e.g., picking most important k coordinates from n coordinates), and $\Psi \in \mathrm{O}(n)$, i.e., an n-dimensional orthogonal matrix. In particular, we consider matrices representing the orthonormal bases in the library as candidates for Ψ. As a classifier g, we adopt Linear Discriminant Analysis (LDA) of R. A. Fisher [59] and Classification and Regression Trees (CART) [18].

4.3 A Review of Some Pattern Classifiers

In this section, we review the pattern classifiers used in our study, i.e., LDA and CART, although other classifiers such as k-nearest neighbor (k-NN) [39], or artificial neural networks (ANN) [125] are all possible to use in our algorithm. The reader interested in comparisons of different classifiers is referred to the excellent review article of Ripley [125]. The useful information on pattern classifiers in general can be found in the books [63], [103], [48], and

[152].

4.3.1 Linear Discriminant Analysis

LDA first tries to do its own feature extraction by a linear map $A^T : \mathcal{X} \to \mathcal{F}$ (in this case not necessarily orthogonal matrix). This map A simultaneously minimizes the scatter of sample vectors (signals) within each class and maximizes the scatter of mean vectors around the total mean vector. To be more precise, let $\boldsymbol{m}_c \stackrel{\Delta}{=} (1/N_c) \sum_{i=1}^{N_c} \boldsymbol{x}_i$ be a mean vector of class c signals[1]. Then the total mean vector $\boldsymbol{m}$ can be defined as:

$$\boldsymbol{m} \stackrel{\Delta}{=} \sum_{c=1}^{C} \pi_c \boldsymbol{m}_c,$$

where π_c is the prior probability of class c (which can be set to N_c/N without the knowledge on the true prior probability). The scatter of samples within each class can be measured by the *within-class covariance matrix*

$$\Sigma_w \stackrel{\Delta}{=} \sum_{c=1}^{C} \pi_c \Sigma_c,$$

where Σ_c is the *sample covariance matrix* of class c:

$$\Sigma_c \stackrel{\Delta}{=} \frac{1}{N_c} \sum_{i=1}^{N_c} (\boldsymbol{x}_i^{(c)} - \boldsymbol{m}_c)(\boldsymbol{x}_i^{(c)} - \boldsymbol{m}_c)^T.$$

The scatter of mean vectors around the total mean can be measured by the *between-class covariance matrix*

$$\Sigma_b \stackrel{\Delta}{=} \sum_{c=1}^{C} \pi_c (\boldsymbol{m}_c - \boldsymbol{m})(\boldsymbol{m}_c - \boldsymbol{m})^T.$$

Then, LDA maximizes a class separability index

$$J(A) \stackrel{\Delta}{=} \mathrm{tr}[(A^T \Sigma_b A)^{-1} (A^T \Sigma_w A)],$$

[1] The sample mean operation $(1/N_c) \sum_{i=1}^{N_c}$ in this subsection can be replaced by expectation E_c for general cases; however, in this thesis, we focus our attention on the cases of a finite number of samples, we stay with the sample mean operations.

which measures how much these classes are separated in the feature space. This requires solving the so-called generalized (or pencil-type) eigenvalue problem,

$$\Sigma_b A = \Sigma_w A \Lambda,$$

where Λ is a diagonal matrix containing the eigenvalues. Once the map A is obtained (normally $k = C - 1$ for LDA), then the feature vector $A^T \boldsymbol{x}_i$ is computed for each i, and finally it is assigned to the class which has the mean vector closest to this feature vector in the Euclidean distance in the feature space. This is equivalent to bisecting the feature space $\mathcal{F}$ by hyperplanes. In this chapter we regard LDA as a classifier although, as explained, it also includes its own feature extractor A.

LDA is the optimal strategy if all classes of signals obey multivariate normal distributions with different mean vectors and an equal covariance matrix [63], [103]. In reality, however, it is hard to assume this condition. Moreover, since it relies on solving the eigensystem, LDA can only extract global features (or squeezes all discriminant information into a few $[C - 1]$ basis vectors) so that the interpretation of the extracted features becomes difficult, it is sensitive to outliers and noise, and it requires $O(n^3)$ calculations.

4.3.2 Classification and Regression Trees

Another popular classification/regression scheme, CART [18] is a nonparametric method which recursively splits the input signal space *along* the coordinate axes and generates a partition of the input signal space into disjoint blocks so that the process can be conveniently described as a binary tree where nodes represent blocks. Such a tree for classification problems is called a *classification tree* (CT). At each node in a CT, a class label is assigned by the majority vote at that node. Then, candidate splits are evaluated by the "information gain" or the quantity called *deviance* and the most "informative" split is selected. The popular measure as the deviance for the classification is again entropy! This time the

entropy of a node is defined as

$$-\sum_{c=1}^{C} p_c \log p_c,$$

where p_c is the proportion of class c samples over the whole samples at that node. Thus, the best split amounts to maximally reducing the entropy of that node. Once the best split is determined, all the input signals belonging to that node is split into two groups (children nodes). Splitting is continued recursively until nodes become "pure", i.e., they contain only one class of signals, or become "sparse", i.e., they contain only a few signals.[2] An example of the CT is shown in Figure 4.1 (which, in fact, is the best tree for Example 4.7 we will study in Section 4.5). Finally, the pruning process to eliminate unimportant branches is usually applied after growing the initial tree to avoid the "overtraining." In Appendix A, we develop the pruning algorithm based on the MDL principle. Regression trees (RTs) of the CART methodology are intensively used in the next two chapters and are described there. We refer the reader to [18] for the details of splitting, stopping, and pruning rules.

CART requires searching and sorting all the coordinates of training signals for the best splits: it is computationally expensive for the problem of high dimensionality. This is more emphasized if we want to split the signal space "obliquely" by taking linear combinations of the coordinates to generate a tree.

4.4 Construction of Local Discriminant Basis

In order to fully utilize the classifiers including the ones reviewed in the previous section, we must supply them *good* features (preferably just a few) and throw out useless part of the data. This improves both accuracy and speed of these classifiers. In this section, we describe a fast algorithm to construct good feature extractors. In particular, we follow the "best-

[2] In the S-PLUS package [140] (the extended version of the statistical language S [10], [23]), which we intensively use to test our algorithms, by default, the split stops if either the number of samples belonging to that node becomes less than 10 or the deviance of that node becomes less than 1% of the deviance of the root node.

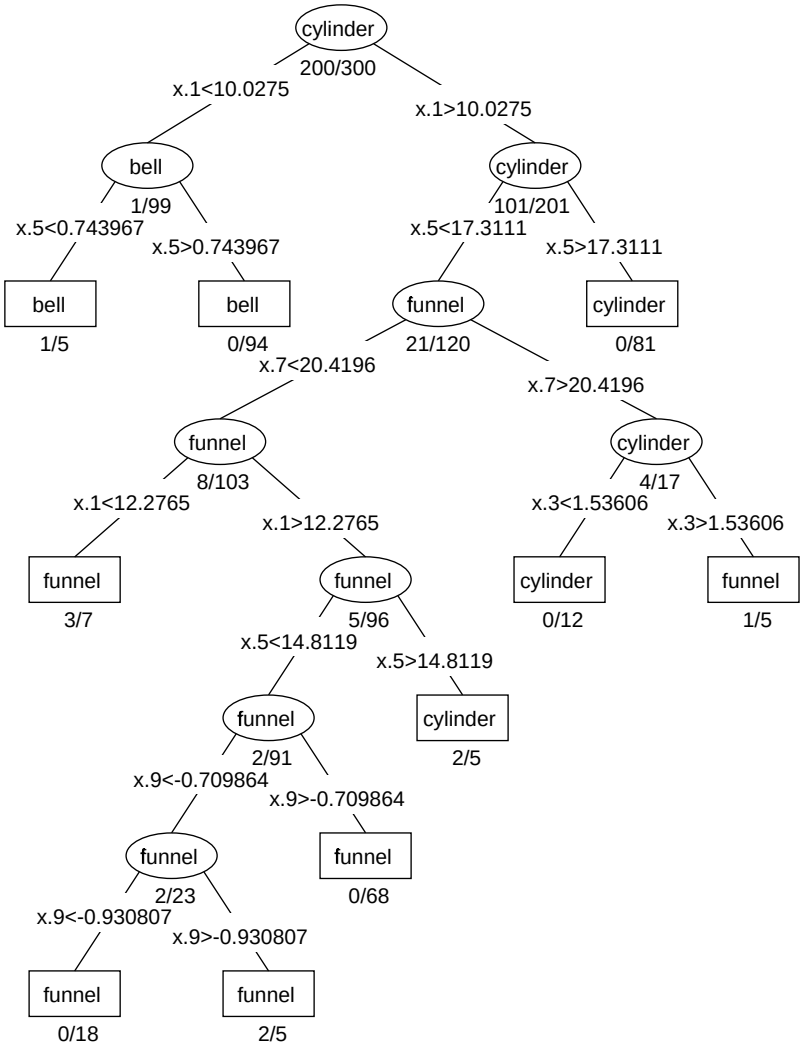


Figure 4.1: An example of a classification tree. Nodes are represented by ellipses (interior nodes) and rectangles (terminal nodes/leaves). The node labels are the predicted class names which are "cylinder", "bell", and "funnel" in this case. The ratio displayed under each node represent the misclassification rate of cases reached to that node. The splitting rules are displayed on the edges connecting nodes. The rule "x.1 $<$ 10.0275" implies "if the first coordinate value of the input signal is less than 10.0275, go to this branch." See Example 4.7 for the details.

basis" paradigm discussed in Chapters 1 and 2 which permits a rapid [e.g., $O(n \log n)$] search among a library of orthonormal bases for the problem at hand; we select basis functions which are well-localized in the time-frequency plane and which most discriminate given classes, and then the coordinates (expansion coefficients) of these basis functions are fed into LDA or CART.

4.4.1 Discriminant measures

Recall that the best-basis algorithm of Coifman and Wickerhauser [35] was developed mainly for signal compression as reviewed in Chapter 2. It selects a basis suitable for signal compression from a dictionary/library of orthonormal bases (i.e., a set of tree-structured subspaces which generates many orthonormal bases and which have different time-frequency localization characteristics) by measuring the efficiency of each subspace in the dictionary/library for representation/compression of signals. The Shannon entropy (2.1) is a natural choice as such a measure of efficiency, or information cost. This quantity measures the flatness of the energy distribution of the signal so that minimizing this leads to an efficient representation (or coordinate system) for the signal. For the classification problems, however, we need a measure to evaluate the power of discrimination of each subspace in the tree-structured subspaces rather than the efficiency in representation. Once the discriminant measure (or discriminant information function) is specified, we can compare the goodness of each node (subspace) for the classification problem to that of union of the two children nodes and can judge whether we should keep the children nodes or not, in the same manner as the best-basis search algorithm.

There are many choices for the discriminant measure (see e.g., [9]); all of them essentially measure "statistical distances" among classes. For simplicity, let us first consider the two-class case. Let $\boldsymbol{p} = \{p_i\}_{i=1}^n$, $\boldsymbol{q} = \{q_i\}_{i=1}^n$ be two nonnegative sequences with $\sum p_i = \sum q_i = 1$ (which can be viewed as normalized energy distributions of signals belonging

to class 1 and class 2 respectively in a coordinate system). The discriminant information function $\mathcal{D}(\boldsymbol{p}, \boldsymbol{q})$ between these two sequences should measure how differently $\boldsymbol{p}$ and $\boldsymbol{q}$ are distributed. One natural choice for $\mathcal{D}$ is the so-called *relative entropy* (also known as *cross entropy*, *Kullback-Leibler distance*, or *I-divergence*) [87]:

$$I(\boldsymbol{p}, \boldsymbol{q}) \stackrel{\Delta}{=} \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i}, \tag{4.1}$$

with the convention, $\log 0 = -\infty$, $\log(x/0) = +\infty$ for $x \gneqq 0$, $0 \cdot (\pm\infty) = 0$. It is clear that $I(\boldsymbol{p}, \boldsymbol{q}) \geq 0$ and equality holds iff $\boldsymbol{p} \equiv \boldsymbol{q}$. This quantity is not a metric since it is not symmetric and does not satisfy the triangle inequality. But it measures the discrepancy of $\boldsymbol{p}$ from $\boldsymbol{q}$. Note that if $q_i = 1/n$ for all i, i.e., q_i are distributed uniformly, then $I(\boldsymbol{p}, \boldsymbol{q}) = -H(\boldsymbol{p})$, the negative of the entropy of the sequence $\boldsymbol{p}$ itself.

The relative entropy (4.1) is asymmetric in $\boldsymbol{p}$ and $\boldsymbol{q}$. For certain applications the asymmetry is preferred (see e.g., Section 4.7). However, if a symmetric quantity is preferred, one should use the *J-divergence* between $\boldsymbol{p}$ and $\boldsymbol{q}$ [87]:

$$J(\boldsymbol{p}, \boldsymbol{q}) \stackrel{\Delta}{=} I(\boldsymbol{p}, \boldsymbol{q}) + I(\boldsymbol{q}, \boldsymbol{p}). \tag{4.2}$$

Another possibility of the measure $\mathcal{D}$ is a ℓ^2 analogue of $I(\boldsymbol{p}, \boldsymbol{q})$ [152]:

$$W(\boldsymbol{p}, \boldsymbol{q}) \stackrel{\Delta}{=} \|\boldsymbol{p} - \boldsymbol{q}\|^2 = \sum_{i=1}^{n} (p_i - q_i)^2. \tag{4.3}$$

Clearly, ℓ^p $(p \geq 1)$ versions of this measure are all possible.

To obtain a fast computational algorithm, the measure $\mathcal{D}$ should be *additive* similarly to $\mathcal{J}$:

Definition 4.1. The discriminant measure $\mathcal{D}(\boldsymbol{p}, \boldsymbol{q})$ is said to be *additive* if

$$\mathcal{D}(\{p_i\}_{i=1}^n, \{q_i\}_{i=1}^n) = \sum_{i=1}^{n} \mathcal{D}(p_i, q_i) \tag{4.4}$$

The measures (4.1) (subsequently (4.2) as well) and (4.3) are both additive.

For measuring discrepancies among C distributions, $\boldsymbol{p}^{(1)}, \ldots, \boldsymbol{p}^{(C)}$, one may take $\binom{C}{2}$ pairwise combinations of $\mathcal{D}$:

$$\mathcal{D}(\{\boldsymbol{p}^{(c)}\}_{c=1}^{C}) \triangleq \sum_{i=1}^{C-1} \sum_{j=i+1}^{C} \mathcal{D}(\boldsymbol{p}^{(i)}, \boldsymbol{p}^{(j)}). \tag{4.5}$$

4.4.2 The local discriminant basis algorithm

The fist step of our strategy for classification is to select a basis which most discriminates given classes from a library of orthonormal bases. Let us first consider the selection of such a basis from a dictionary of orthonormal bases in the library. Given an additive discriminant measure $\mathcal{D}$, what quantity should be supplied to $\mathcal{D}$ to evaluate the discrimination power of each subspace $\Omega_{j,k}$ in the binary-tree-structured subspaces in the dictionary? In order to fully utilize the time-frequency localization characteristics of our dictionary of bases, we compute the following *time-frequency energy map* for each class and supply them to $\mathcal{D}$:

Definition 4.2. Let $\{\boldsymbol{x}_i^{(c)}\}_{i=1}^{N_c}$ be a set of training signals belonging to class c. Then the *time-frequency energy map* of class c, denoted by Γ_c, is a table of real values specified by the triplet (j,k,l) as

$$\Gamma_c(j,k,l) \triangleq \sum_{i=1}^{N_c} \left(\boldsymbol{w}_{j,k,l}^T \boldsymbol{x}_i^{(c)} \right)^2 / \sum_{i=1}^{N_c} \|\boldsymbol{x}_i^{(c)}\|^2, \tag{4.6}$$

for $j = 0, \ldots, J,\ k = 0, \ldots, 2^j - 1,\ l = 0, \ldots, 2^{n_0 - j} - 1$.

In other words, Γ_c is computed by accumulating the squares of expansion coefficients of the signals at each position in the table followed by the normalization by the total energy of the signals belonging to class c. (This normalization may be important especially if there is significant differences in number of samples among classes.) In the following, we use the notation:

$$\mathcal{D}(\{\Gamma_c(j,k,\cdot)\}_{c=1}^{C}) = \sum_{l=0}^{2^{n_0-j}-1} \mathcal{D}(\Gamma_1(j,k,l), \ldots, \Gamma_C(j,k,l)).$$

Here is an algorithm to select an orthonormal basis (from the dictionary) which maximizes the discriminant measure on the time-frequency energy distributions of classes. We call this a *local discriminant basis* (LDB). Similarly to the best-basis algorithm, let $B_{j,k}$ denote a set of basis vectors at the subspace $\Omega_{j,k}$ as defined in (2.3). Let $A_{j,k}$ represent the LDB (which we are after) restricted to the span of $B_{j,k}$. Also, let $\Delta_{j,k}$ be a work array containing the discriminant measure of the subspace $\Omega_{j,k}$.

Algorithm 4.3 (The Local Discriminant Basis Selection Algorithm). *Given a training dataset $\mathcal{T}$ consisting of C classes of signals $\{\{\boldsymbol{x}_i^{(c)}\}_{i=1}^{N_c}\}_{c=1}^{C}$,*

Step 0: *Choose a dictionary of orthonormal bases $\mathfrak{D}$ (i.e., specify QMFs for a wavelet packet dictionary or decide to use either the local cosine dictionary or the local sine dictionary) and specify the maximum depth of decomposition J and an additive discriminant measure $\mathcal{D}$.*

Step 1: *Construct time-frequency energy maps Γ_c for $c = 1, \ldots, C$.*

Step 2: *Set $A_{J,k} = B_{J,k}$ and $\Delta_{J,k} = \mathcal{D}(\{\Gamma_c(J,k,\cdot)\}_{c=1}^{C})$ for $k = 0, \ldots, 2^J - 1$.*

Step 3: *Determine the best subspace $A_{j,k}$ for $j = J-1, \ldots, 0,\ k = 0, \ldots, 2^j - 1$ by the following rule:*

Set $\Delta_{j,k} = \mathcal{D}(\{\Gamma_c(j,k,\cdot)\}_{c=1}^{C})$.

If $\Delta_{j,k} \geq \Delta_{j+1,2k} + \Delta_{j+1,2k+1}$,

then $A_{j,k} = B_{j,k}$,

else $A_{j,k} = A_{j+1,2k} \oplus A_{j+1,2k+1}$ *and set* $\Delta_{j,k} = \Delta_{j+1,2k} + \Delta_{j+1,2k+1}$.

Step 4: *Order the basis functions by their power of discrimination (see below).*

Step 5: *Use $k\ (\leq n)$ most discriminant basis functions for constructing classifiers.*

The selection (or pruning) process in Step 3 is fast, i.e., $O(n)$ since the measure $\mathcal{D}$ is additive. After this step, we have a complete orthonormal basis LDB.

Proposition 4.4. *The basis obtained by Step 3 of Algorithm 4.3 maximizes the additive discriminant measure $\mathcal{D}$ on the time-frequency energy distributions among all the bases in the dictionary $\mathfrak{D}$ obtainable by the divide-and-conquer algorithm.*

Proof. Similarly to the proof of Proposition 2.4 described in [35], [157], we show this by induction on j (in the decreasing order, $J, J-1, \ldots, 0$). Let $\Omega_{j,k}$ be a span of $B_{j,k}$ as in Chapter 2. Let $A'_{j,k}$ be any basis of $\Omega_{j,k}$. Let $\Delta'_{j,k}$ be its discriminant measure on the time-frequency energy distributions. There is only one basis for $\Omega_{J,k}$ in $\mathfrak{D}$ (which is $B_{J,k}$) because J is the maximum depth of decomposition in $\mathfrak{D}$. Then, for $J-1$, let $A'_{J-1,k}$ be any basis for $\Omega_{J-1,k}$. Then either $A'_{J-1,k} = B_{J-1,k}$ or $A'_{J-1,k} = A'_{J,2k} \oplus A'_{J,2k+1}$. By the inductive hypothesis, $\Delta_{J,2k} \geq \Delta'_{J,2k}$ and $\Delta_{J,2k+1} \geq \Delta'_{J,2k+1}$. Thus, from the equations in Step 3, $\Delta_{J-1,k} \geq \Delta_{J,2k} + \Delta_{J,2k+1} \geq \Delta'_{J,2k} + \Delta'_{J,2k+1}$ for any $k \in \{0, 1, \ldots, 2^{J-1} - 1\}$. This implies that $\Delta_{0,0}$ becomes the largest possible discriminant value using this divide-and-conquer algorithm. □

Once the LDB is selected, we can use all expansion coefficients of signals in this basis as features; however, if we want to reduce the dimensionality of the problem, the two subsequent steps are still necessary. In Step 4, there are several choices as a measure of discriminant power of an individual basis function. For simplicity in notation, let $\lambda = (j, k, m) \in \mathbb{Z}^3$ be a triplet specifying one of the LDB functions selected in Step 3, and let $\alpha^{(c)}_{\lambda,i} = \boldsymbol{w}_\lambda^T \boldsymbol{x}_i^{(c)}$, i.e., an expansion coefficient of $\boldsymbol{x}_i^{(c)}$ in the basis vector $\boldsymbol{w}_\lambda$.

(a) the discriminant measure of a single basis function $\boldsymbol{w}_\lambda$:

$$\mathcal{D}(\Gamma_1(\lambda), \ldots, \Gamma_C(\lambda)). \tag{4.7}$$

(b) the Fisher's class separability of the expansion coefficients onto the basis function $\boldsymbol{w}_\lambda$:

$$\frac{\sum_{c=1}^{C} \pi_c (\mathrm{mean}_i(\alpha_{\lambda,i}^{(c)}) - \mathrm{mean}_c(\mathrm{mean}_i(\alpha_{\lambda,i}^{(c)})))^2}{\sum_{c=1}^{C} \pi_c \,\mathrm{var}_i(\alpha_{\lambda,i}^{(c)})}, \tag{4.8}$$

where $\mathrm{mean}_i(\cdot)$ and $\mathrm{var}_i(\cdot)$ are operations to take the sample mean and variance with respect to the samples indexed by i, respectively.

(c) the robust version of (b):

$$\frac{\sum_{c=1}^{C} \pi_c |\,\mathrm{med}_i(\alpha_{\lambda,i}^{(c)}) - \mathrm{med}_c(\mathrm{med}_i(\alpha_{\lambda,i}^{(c)}))|}{\sum_{c=1}^{C} \pi_c \,\mathrm{mad}_i(\alpha_{\lambda,i}^{(c)})}, \tag{4.9}$$

where $\mathrm{med}_i(\cdot)$ and $\mathrm{mad}_i(\cdot)$ are operations to take the sample median and median absolute deviation with respect to the samples indexed by i, respectively.

See [9], [77] for more examples. We note that this step can also be viewed as a restricted version of the projection pursuit algorithm [77].

Step 5 reduces the dimensionality of the problem from n to k without losing the discriminant information in terms of time-frequency energy distributions among classes. Thus many interesting statistical techniques which are usually computationally too expensive for n dimensional problems become feasible. How to select the best k is a tough interesting question. One possibility is to use model selection methods such as the minimum description length (MDL) criterion [128] (see also Chapter 3).

We can easily extend Algorithm 4.3 to a library of orthonormal bases. Let $\mathfrak{L} = \{\mathfrak{D}_1, \ldots, \mathfrak{D}_M\}$ denote a library. Let $\mathfrak{B}_m$ be the LDB selected from the dictionary $\mathfrak{D}_m$. Each LDB $\mathfrak{B}_m$ is associated with the maximum value of a discriminant measure on the time-frequency energy distributions as shown in Proposition 4.4. Let Δ_m^* denote this maximum value. Then we can simply pick the basis giving the maximum value among $\{\Delta_m^*\}$: the "best" of the LDBs $\mathfrak{B}_{m^*}$ is given by $\Delta_{m^*}^* = \max_{1 \le m \le M} \Delta_m^*$.

Remark 4.5. Our LDB method can be used for certain regression problems which are closely related to the classification problems. Let a training dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ consist of

C classes of samples $\{\{(\boldsymbol{x}_i, y_i)\}_{i \in I_c}\}_{c=1}^{C}$ as before. Let us assume that the response y_i for $i \in I_c$ is now a real number conditioned as $y_i \in R_c = [a_c, b_c]$ and that $\cap_{c=1}^{C} R_c \neq \emptyset$. Under this assumption, suppose one wants to estimate the response y_i for a given input signal $\boldsymbol{x}_i$ rather than its class label or assignment. This situation is not really special; we often encounter this type of regression problems in medical and geological sciences where the objects are made in the course of nature. In Chapter 6, we test and analyze the real dataset from the field of geophysical prospecting using the algorithms described in this chapter.

4.5 Examples

To demonstrate the capability of the LDB method, we conducted two classification experiments using synthetic signals. In both cases, we specified three classes of signals by analytic formulas. For each class, we generated 100 training signals and 1000 test signals. We first constructed LDA-based classifier and CT (with and without pruning) using the training signals represented in the original coordinate (i.e., standard Euclidean) system. We used the pruning algorithm based on the MDL principle described in Appendix A. Then we fed the test signals into these classifiers. Next we computed the LDB (using (4.5) as a discriminant measure and (4.7) for ordering the individual basis functions) on the training signals. Then we selected a small number of most discriminant basis functions, say about 10 % of the dimensionality of the signals, and used these coordinates to construct LDA-based classifier and CTs. Finally the test signals were projected onto these selected LDB functions and then fed into these classifiers. For each method, we computed the misclassification rates on the training dataset and the test dataset.

Example 4.6. *Triangular waveform classification.*

This is an example for classification originally examined in [18]. The dimensionality of the signal was extended from 21 in [18] to 32 for the dyadic dimensionality requirement of the bases under consideration. Three classes of signals were generated by the following

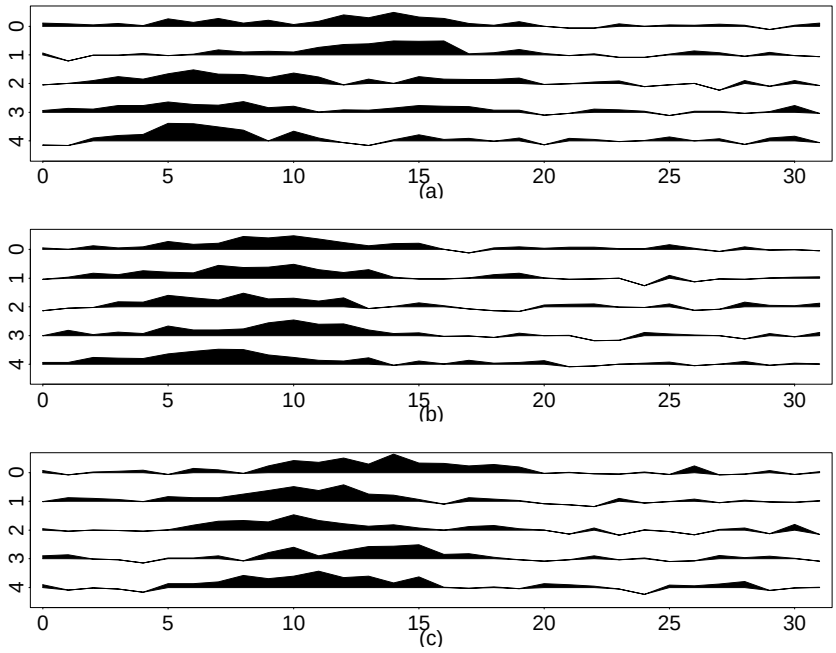


Figure 4.2: Five sample waveforms from (a) Class 1, (b) Class 2, and (c) Class3.

formulas:

$$x^{(1)}(i) = uh_1(i) + (1-u)h_2(i) + \epsilon(i) \quad \text{for Class 1,}$$

$$x^{(2)}(i) = uh_1(i) + (1-u)h_3(i) + \epsilon(i) \quad \text{for Class 2,}$$

$$x^{(3)}(i) = uh_2(i) + (1-u)h_3(i) + \epsilon(i) \quad \text{for Class 3,}$$

where $i = 1, \ldots, 32$, $h_1(i) = \max(6 - |i-7|, 0)$, $h_2(i) = h_1(i-8)$, $h_3(i) = h_1(i-4)$, u is a uniform random variable on the interval $(0, 1)$, and $\epsilon(i)$ are standard normal variates. Figure 4.2 shows five sample waveforms from each class. The LDB was computed from the wavelet packet coefficients with the 6-tap coiflet filter [43]. Then the five most discriminant coordinates were selected. In Figure 4.3, we compare the top five vectors from LDA and LDB. Only top two vectors were useful in LDA in this case. The top five LDB vectors look similar to the functions h_j or their derivatives whereas it is difficult to interpret the LDA vectors. The misclassification rates are given in Table 4.1. The best result so far

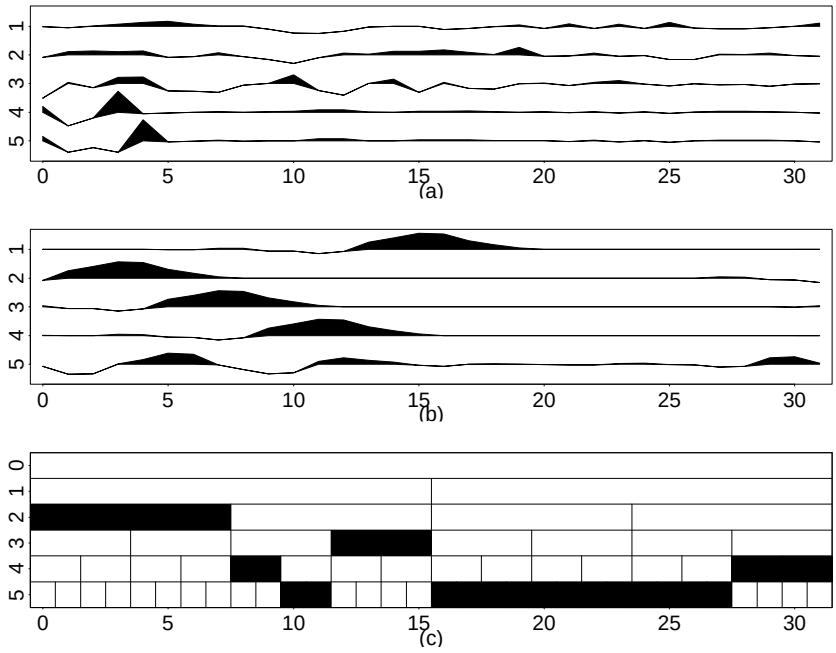


Figure 4.3: Plots from the analysis of Example 4.6: (a) Top five LDA vectors. (b) Top 5 LDB vectors. (c) The subspaces selected as the LDB.

was obtained by applying LDA to the top 5 LDB coordinates. We would like to note that according to Breiman et al. [18], the Bayes error of this example is about 14 %.

Example 4.7. *Signal shape classification.*

The second example is a signal shape classification problem. In this example, we try to classify synthetic noisy signals with various shapes, amplitudes, lengths, and positions into three possible classes. More precisely, sample signals of the three classes were generated by:

$$c(i) = (6+\eta)\cdot\chi_{[a,b]}(i) + \epsilon(i) \qquad \text{for "cylinder" class,}$$

$$b(i) = (6+\eta)\cdot\chi_{[a,b]}(i)\cdot(i-a)/(b-a) + \epsilon(i) \qquad \text{for "bell" class,}$$

$$f(i) = (6+\eta)\cdot\chi_{[a,b]}(i)\cdot(b-i)/(b-a) + \epsilon(i) \qquad \text{for "funnel" class,}$$

where $i = 1, \ldots, 128$, a is an integer-valued uniform random variable on the interval $[16, 32]$, $b-a$ also obeys an integer-valued uniform distribution on $[32, 96]$, η and $\epsilon(i)$ are standard

Method	Error rate (%)	
	Training	Test
LDA on STD	13.33	20.90
FCT on STD	6.33	29.87
PCT on STD	29.33	32.97
LDA on LDB5	14.33	**15.90**
FCT on LDB5	7.00	21.37
PCT on LDB5	17.00	25.10
FCT on LDB	7.33	23.60
PCT on LDB	17.00	25.10

Table 4.1: Misclassification rates of Example 4.6. In Method column, FCT and PCT denote the full and pruned classification trees, respectively. STD, LDB5, and LDB represent the standard Euclidean coordinates, the top 5 LDB coordinates, and all the LDB coordinates, respectively. We do not show the error rates of LDA on all the LDB coordinates since this is the same as the ones of LDA on STD theoretically. The smallest error on the test dataset is shown in bold font.

normal variates, and $\chi_{[a,b]}(i)$ is the characteristic function on the interval $[a, b]$. Figure 4.4 shows five sample waveforms from each class. If there is no noise, we can characterize the "cylinder" signals by two step edges and constant values around the center, the "bell" signals by one ramp and one step edge in this order and positive slopes around the center, and the "funnel" signals by one step edge and one ramp in this order and negative slopes around the center.

The 12-tap coiflet filter [43] was used for the LDB selection. Then the 10 most important coordinates were selected. In Figure 4.5, we compare the top 10 LDA and LDB vectors. Again, only the top two vectors were used for classification in LDA case. These LDA vectors are very noisy and it is difficult to interpret what information they captured. On the other hand, we can observe that the top 10 LDB vectors are located around the edges the centers of the signals. Also note that some of the vectors work as a smoother (low pass filter) and the others work as a edge detector (band pass filter), so that the resulting expansion coefficients carry the information on the edge positions and

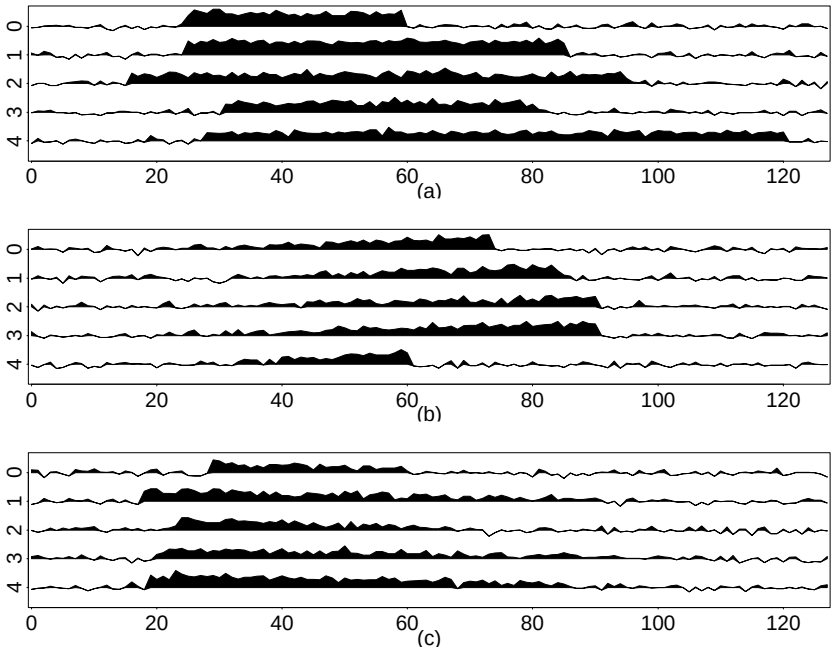


Figure 4.4: Five sample waveforms from (a) "cylinder" class, (b) "bell" class, and (c) "funnel" class.

types. The misclassification rates in this case are displayed in Table 4.2. As expected, LDA applied to the original coordinate system was almost perfect with respect to the training data, but it adapted too much to the features specific to the training data, and lost its generalization power; when applied to the new test dataset, it did not work well. The best result was obtained using the full CT on the top 10 LDB coordinates. In this case, the misclassification rates of the training data and test data are very close; that is, the algorithm really "learned" the structures of signals. In fact, this best tree was already shown in Figure 4.1 in Section 4.3. If the tree-based classification is combined with the coordinate system capturing local information in the time-frequency plane, the interpretation of the result becomes so explicit and easy: in Figure 4.1 we find that the LDB coordinate #1 is checked first. If this is less than 10.0275, it is immediately classified as "bell." From Figure 4.5 (b), we observe that the LDB function #1 is located around $i = 30$ which,

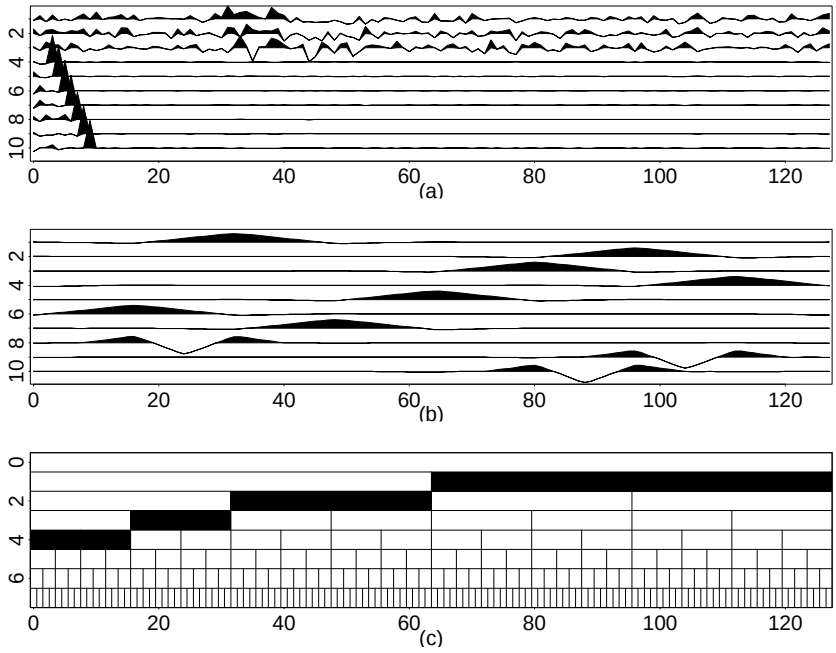


Figure 4.5: Plots from the analysis of Example 4.7: (a) Top 10 LDA vectors. (b) Top 10 LDB vectors. (c) The subspaces selected as the LDB.

in fact, coincides with the starting position (the parameter a in the formulas) of various signals. Around this region, both the cylinder and the funnel signals have sharp step edges. On the other hand, the bell signals start off linearly. Thus CART algorithm found that the LDB function #1 is the most important coordinate in this example. The separating the cylinder class from the funnel class turned out to be more difficult because of the large variability of the ending positions. This resulted in the more complicated structure of the right branch from the root node. But we can still obtain the intuitive interpretation: the first node in the right branch (with "cylinder" label) from the root node is split into either "funnel" or "cylinder" depending on the LDB coordinate #5 which is located around the middle of the axis ($i = 64$). The cylinder signals have roughly constant values around this area whereas the funnel signals roughly decrease in a linear fashion. One can continue the interpretation in a similar manner for all remaining nodes.

Method	Error rate (%)	
	Training	Test
LDA on STD	0.33	13.17
FCT on STD	3.00	13.37
PCT on STD	6.00	10.67
LDA on LDB10	3.67	6.20
FCT on LDB10	3.00	**3.83**
PCT on LDB10	4.33	6.40
FCT on LDB	3.33	5.97
PCT on LDB	4.00	6.13

Table 4.2: Misclassification rates of Example 4.7. In Method column, LDB10 represents the top 10 LDB coordinates, Other abbreviations are exactly the same as Table 4.1. The smallest error on the test dataset is shown in bold font.

In both examples, we see that the misclassification rates (of the test datasets) using the pruned CTs are worse than those using the full CTs. This tendency can also be found in some of the examples studied in [125]. We will investigate this issue further in our future research project.

From these examples, we can see that it is more important to select the good features than to select the best possible classifier without supplying the good features; each classifier has its advantages and disadvantages [125], i.e., the best classifier heavily depends on the problem (e.g., LDA was better than CART in Example 4.6 whereas the situation was opposite in Example 4.7.) By supplying a handful of good features, we can greatly enhance the performance of classifiers.

4.6 To Denoise or Not to Denoise?

The LDB-based classification developed so far worked quite well for the noisy synthetic datasets. An interesting question is whether this good performance is attributed solely to the denoising capability of the basis functions, or to the local features extracted by the

Method	Error rate (%) Training	Test
LDA on STD	15.33	24.13
FCT on STD	6.33	24.97
PCT on STD	18.00	24.37
LDA on LDB5	17.00	**21.37**
FCT on LDB5	7.67	22.40
PCT on LDB5	22.00	28.87
FCT on LDB	6.33	25.07
PCT to LDB	18.00	27.40

Table 4.3: Misclassification rates of Example 4.6 with the denoised input signals.

Method	Error rate (%) Training	Test
LDA on STD	2.00	18.83
FCT on STD	2.33	7.33
PCT on STD	5.00	8.13
LDA on LDB10	4.67	**6.73**
FCT on LDB10	3.67	8.10
PCT on LDB10	4.00	7.03
FCT on LDB	2.67	7.10
PCT on LDB	4.00	7.03

Table 4.4: Misclassification rates of Example 4.7 with the denoised input signals.

basis functions, or both. Thus, we applied the MDL-based denoising algorithm combined with shift-average method developed in Chapter 3 to the input signals. In this exercise, we fixed the QMF for the denoising, i.e., C06 for the signals of Example 4.6, and C12 for the signals of Example 4.7. Also we used the five-point shifts in that algorithm. After the denoising, exactly the same procedures were applied. The following two tables summarize these results. We can observe that the following tendency:

- Each error rate on the denoised signals represented in the LDB coordinates (whether taking top few coordinates or not) is consistently worse than the one without denoising.

- Error rates on the denoised signals represented in the standard Euclidean system depends on the methods; in both examples LDA resulted in the larger error rates on the denoised signals than on the original noisy signals. and the situation is opposite for the CTs.

- In both examples, the best performance is obtained using the signals without denoising.

Based on these observations, we may conclude that too much denoising should not be applied prior to the LDB analysis since it may lose some important information for classification. We will address how to choose the number of basis functions to retain for denoising without deteriorating the classification performance in our future project.

4.7 Signal/Background Separation by LDB

LDB vectors can also be used as a tool for extracting signal component from the data obscured by some unwanted noise or "background" (which may not be random). Let class 1 consist of a signal plus noise or a signal plus "background" and let class 2 consist of a pure noise or "background" Then, by selecting the LDB maximizing $\mathcal{D}$ between class 1 and class2, we can construct the best basis for denoising arbitrary noise or pulling a signal out of a textured background. In this application, the asymmetric relative entropy (4.1) makes more sense than the symmetric version (4.2).

We show one example here. As "background" (class 2), we generated 100 synthetic sinusoid with random phase as $b(k) = \sin(\pi(k/32+u))$, where $k = 1, \ldots, 128$, and u is a uniform random variable on $(0, 1)$. As class 1 samples, we again generated 100 "backgrounds", and added a small spike (as a "signal" component) for each sample vector randomly between $20 \leq k \leq 60$, i.e., $x(k) = \sin(\pi(k/32 + u)) + 0.01\delta_{k,r}$, where $\delta_{k,r}$ is the Kronecker delta and r is an integer-valued uniform random variable on the interval $[20, 60]$. Figure 4.6 shows

how these "backgrounds" were removed. Figure 4.6 (a) shows 10 sample vectors of class 1. We can hardly see the spikes. Then we transformed both class 1 and 2 samples by the discrete sine transform (DST) into "frequency" domain. Figure 4.6 (b) shows the transformed version of Figure 4.6 (a). Then these DST coefficients of both classes were supplied to the LDB algorithm of Section 4.4 using the local sine basis dictionary (which essentially does segmentation in frequency domain). After the LDB was found, the basis vectors were sorted by (4.7). The top 20 LDB vectors are displayed in Figure 4.6 (c). We can clearly see that the top eight basis vectors are concentrated around low frequency region and other vectors are located in higher frequency region. We regard the subspace spanned by these eight LDB vectors as "background" using the *a priori* knowledge that the "background" component consists of only low frequency component. The reason why these vectors have large values in (4.7) is that the "background" parts of class 1 samples are different from class 2 samples in phase, and the DST is not a shift-invariant transform. After removing the component belonging to this "background" subspace, we reconstructed the "signal" component of class 1 samples by inverse DST which are shown in Figure 4.6 (d). We can clearly see the spikes now. The LDB thus can improve the algorithm of extracting "coherent" component from the data by Coifman, Majid, and Wickerhauser [30], [36] if we know the statistics of the background *a priori* or have actual pure background signals.

A similar idea for multidimensional signals has been proposed by Harlan et al. [68]; they considered the problem of removing linear and hyperbolic structures from images (representing the geophysical acoustic signals such as Figure 3.5 in Chapter 3) using the Radon and the generalized Radon transforms. The key observation is that the structural components (e.g., lines and hyperbolas) in the images can be well-compressed or "focused" in the certain transformed domains (e.g., the Radon, the generalized Radon transformed domains). On the other hand, the unstructured components or backgrounds are "defocused" in these domains. Based on this observation, the thresholding operation in the transformed domain

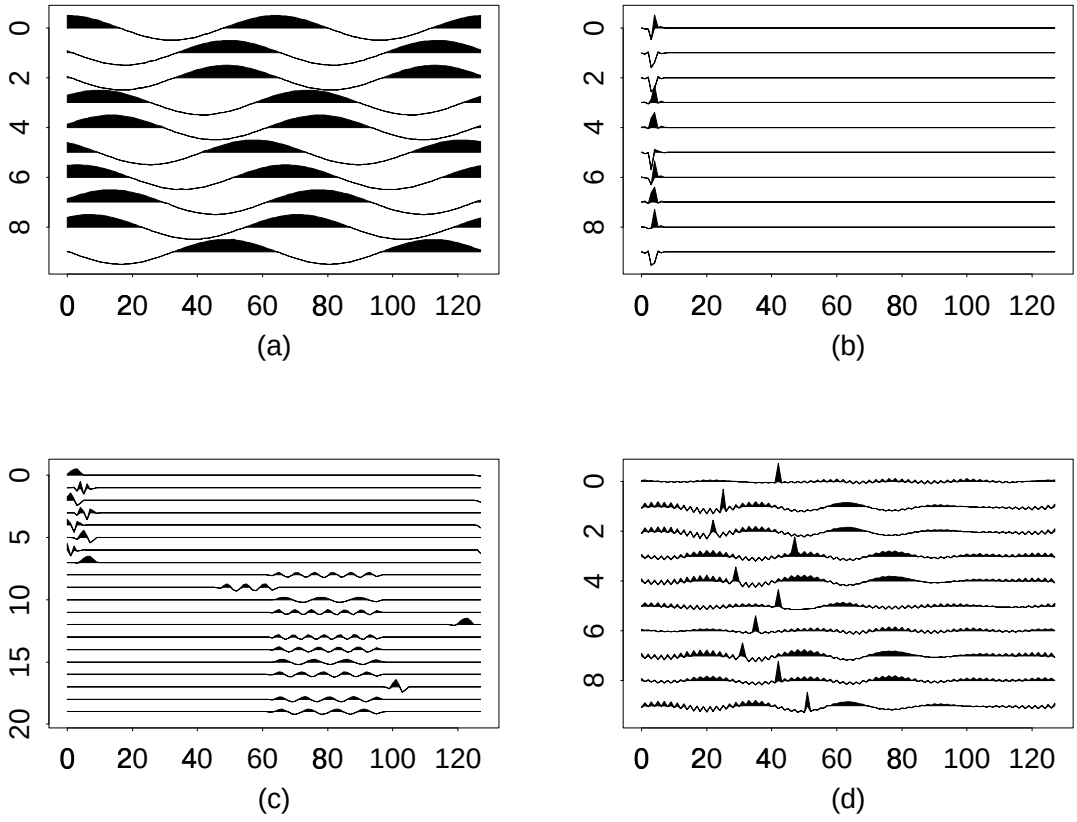


Figure 4.6: (a) Ten samples of Class 1 vectors, i.e., sinusoids plus spikes. (b) DST coefficients of vectors in (a). (c) Top 20 LDB vectors using the local sine dictionary on the frequency domain. (d) Reconstructed spikes after removing the "background".

is applied and only the "focused" objects in the transformed domain remain. Then the inverse transform only reconstructs the structural components and eliminates backgrounds. In this sense, the "structure" strongly depends on the transform under consideration. Our philosophy is to use the library of bases in Chapter 2; we have a large collection of transforms each of which can represent many different "structures" in signals. For images or multidimensional signals, it is not simple to determine which basis should be included in the library of bases because: 1) there are many possible two-dimensional bases both separa-

ble and nonseparable (see Section 2.7 and the references therein), and 2) the computational cost is much higher ($\approx O(n^2 \log_4 n^2)$ for an image of n rows and n columns). Here is the place to use *a priori* information carefully to restrict the number of bases or dictionaries in the library to achieve both the computational efficiency and the representation power of the bases. Classification and signal/noise separation for images are our important future project.

4.8 Summary

We have described an algorithm to construct an adaptive local orthonormal basis [*local discriminant basis* (LDB)] for classification problems by selecting a basis from a library of orthonormal bases using a discriminant measure (e.g., relative entropy). The basis functions generated by this algorithm can capture relevant local features (in both time and frequency) in data. LDB provides us with better insight and understanding of relationships between the essential features of the input signals and the corresponding outputs (class names), and enhances the performance of classifiers. We have demonstrated that LDB can also be used for pulling out signal component from the data consisting of signals plus "backgrounds."

Chapter 5

Local Regression Bases

5.1 Introduction

Basis functions selected by the "best-basis" paradigm from a library of orthonormal bases have been found useful for the classification problems in the previous chapter. They provide us with better insight and understanding of relationships between the local features of the input signals and the class assignments. A natural question is how to extend this paradigm for regression problems. Our definition of regression is simply any statistical method to construct a mapping function from the input signal space (generally high dimensional) to the response space (normally low dimensional). Estimation or prediction of some quantity from the input signals can be considered as a regression problem; e.g., classification problems are a subset of regression problems. In this chapter, we describe a method to select a complete orthonormal basis from a library of orthonormal bases suitable for regression problems.

The organization of this chapter is as follows. In Section 5.2, we formulate the problem of feature extraction and regression, and summarize the tree-based regression method in the CART methodology. In Section 5.3, we propose an algorithm for constructing such a local basis for regression problems. Then in Section 5.4, we show the result of applying our

method to the same examples used in Section 4.5 and compare their performances. Finally we discuss the related methods proposed by others and contrast them with our method in Section 5.5.

5.2 Problem Formulation

We formulate a regression problem in a similar manner as the classification problem in the previous chapter: let $\mathcal{T} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N \subset \mathcal{X} \times \mathcal{Y}$ be a training dataset with a signal space $\mathcal{X} \subset \mathbb{R}^n$, and a response space $\mathcal{Y} = \mathbb{R}$. We want to find a feature extractor $f : \mathcal{X} \to \mathcal{F} \subset \mathbb{R}^k$, $(k \leq n)$ for extracting relevant features and reducing the dimensionality of the problem without losing important information as much as possible. If we can succeed in constructing such a feature extractor, then the subsequent regression process can be improved in its accuracy and efficiency. We call this regression process $g : \mathcal{F} \to \mathcal{Y}$ a *predictor*. As in the previous section, $d = g \circ f : \mathcal{X} \to \mathcal{Y}$ denotes the overall regression process. We assess the performance of the whole process by the regression error (also called prediction error) using a *test* dataset $\mathcal{T}'$ as $\mathcal{T}' = \{(y_i', \boldsymbol{x}_i')\}_{i=1}^{N'}$ (which has not been used to construct the feature extractors and predictors) as $(1/N') \sum_{i=1}^{N'} \delta_p(y_i' - d(\boldsymbol{x}_i'))$, where $\delta_p(r) = r^p$ with $1 \leq p < \infty$ or the relative ℓ^p error, $\|\boldsymbol{y}' - \widehat{\boldsymbol{y}'}\|_p / \|\boldsymbol{y}'\|_p$, where $\boldsymbol{y}' = (y_i')_{i=1}^{N'}$ and $\widehat{\boldsymbol{y}'} = (d(\boldsymbol{x}_i'))_{i=1}^{N'}$. The resubstitution error (using training dataset), of course, gives overly optimistic figures.

In this chapter, we focus on the feature extractors of the form $f = \Theta^{(k)} \circ \Psi$, where $\Theta^{(k)} : \mathcal{X} \to \mathcal{F}$ represents the selection rule (e.g., picking most important k coordinates from n coordinates), and $\Psi \in \mathrm{O}(n)$, i.e., an n-dimensional orthogonal matrix. As a regression method g, we adopt the tree-based regression in the CART methodology [18] although other multivariate regression techniques such as ordinary linear regression [120], [54], projection pursuit regression [60], [77], or neural network regression [125], [24] are all possible to use.

Before proceeding to the description of a basis selection algorithm, we briefly review regression trees (RTs) in the CART methodology. Essentially, an RT constructs a piecewise

constant approximation to the response vector $\boldsymbol{y}$ by recursively partitioning the input signal space (along the coordinate axes) into a set of disjoint blocks and taking the average of the response values at each block. An actual prediction works as follows: each input signal $\boldsymbol{x}_i$ is dropped in the RT and reaches a certain terminal node; then the average value mentioned above is assigned as the predicted response value $\hat{y}_i$. An algorithm for constructing an RT is easily obtained by small modifications on the classification tree (CT) algorithm; as a node value, replace the class assignment by the average of the response values of the samples at that node, and as the deviance of a node, replace the entropy by the residual sum of squares (i.e., $\|\boldsymbol{y} - \widehat{\boldsymbol{y}}\|^2$); see [18, Chapter 8] for more details. The MDL-based pruning algorithm can also be modified easily for RTs; see Appendix A.

5.3 Construction of Local Regression Basis

We want to select a complete orthonormal basis from a library of orthonormal bases, i.e., a set of tree-structured subspaces. For classification, we have used relative entropy to measure the goodness of each subspace which leads to LDB. For the regression problem, instead of relative entropy, we use regression error computed from the expansion coefficients belonging to a subspace by invoking a specified regression method. As described in the previous section, let g denotes the final regression method (such as RT) after selecting the basis suitable for the regression. Let $g_j : \mathbb{R}^{2^{n_0-j}} \to \mathcal{Y}$ denote the regression method on the subspace $\Omega_{j,\cdot}$. The methods g_js are normally the same regression method as g except the dimensionality of the input space. Then, we may take the following relative ℓ^p error on the subspace to evaluate it:

$$\mathcal{R}_p(B_{j,k}X; g_j) \stackrel{\Delta}{=} \|\boldsymbol{y} - \widehat{\boldsymbol{y}}(B_{j,k}X; g_j)\|_p / \|\boldsymbol{y}\|_p, \tag{5.1}$$

where $B_{j,k}$ is a matrix of basis vectors belonging to the subspace $\Omega_{j,k}$ defined as (2.3), $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$, i.e., a matrix consisting of the training signals, and $\hat{y}_i(B_{j,k}X; g_j)$ is the

estimate of y_i by the regression method g_j on $B_{j,k}X$, i.e., all the expansion coefficients of the training signals belonging to the subspace $\Omega_{j,k}$. Alternatively, we may take a residual sum of p-th powers:

$$\mathcal{R}_p(B_{j,k}X; g_j) \stackrel{\Delta}{=} \sum_{i=1}^{N} |y_i - \hat{y}_i(B_{j,k}X; g_j)|^p. \tag{5.2}$$

We note that these measures are not additive in the sense of Definition 2.3. The following algorithm selects a basis suitable for the regression problem relative to g. We call this basis a *local regression basis* (LRB) relative to g. In contrast with the LDB algorithm of the previous chapter where the statistical method (classification) is used after the basis selection, the following LRB algorithm integrates the statistical method (regression) into the basis selection mechanism. The following is an algorithm for selecting such a basis from a dictionary of orthonormal bases.

Algorithm 5.1 (The Local Regression Basis Selection Algorithm). *Given a training dataset $\mathcal{T}$,*

Step 0: *Choose a dictionary of orthonormal bases $\mathfrak{D}$ (i.e., specify QMFs for a wavelet packet dictionary or decide to use either the local cosine dictionary or the local sine dictionary) and specify the maximum depth of decomposition J, a regression method g, and a measure of regression error $\mathcal{R}_p$.*

Step 1: *Expand each signal into the dictionary $\mathfrak{D}$.*

Step 2: *Set $A_{J,k} = B_{J,k}$ for $k = 0, \ldots, 2^J - 1$.*

Step 3: *Determine the best subspace $A_{j,k}$ for $j = J-1, \ldots, 0,\ k = 0, \ldots, 2^j - 1$ by*

$$A_{j,k} = \begin{cases} B_{j,k} & \text{if } \mathcal{R}_p(B_{j,k}X; g_j) \le \mathcal{R}_p(A_{j+1,2k}X \cup A_{j+1,2k+1}X; g_{j+1}), \\ A_{j+1,2k} \oplus A_{j+1,2k+1} & \text{otherwise.} \end{cases} \tag{5.3}$$

Step 4: *Supply $k\,(\leq n)$ most important coordinates to the final regression method g.*

Unlike the BB and LDB, this basis may not give the smallest prediction error (using g_js) in the set of all possible bases obtainable by the divide-and-conquer algorithm from the dictionary. This is not because the prediction error is non-additive but because the best prediction error of the union of the two individually-best subspaces may not be necessarily smaller than the best prediction error of the union of the two subspaces each of which is not individually-best by itself. This is a rather general problem in feature selection based on the prediction error or misclassification error. See [145], [38] for a few interesting examples; see also [63, Section 10.5], [103, Chapter 12] for more information. In this sense, the LRB is still a first step toward the general regression problem using the best-basis paradigm. We will study how to obtain a better selection scheme in our future project.

Step 4 is the so-called "selection-of-variables" problem. After Step 3, the regression error using g_j is assigned to each subspace. Hence one way to reduce the dimensionality of the problem is to only consider the coefficients generated by the projection onto the subspaces whose regression errors are smaller than some threshold. The other way is to let the final regression method g select the most important coordinates by supplying all the coefficients. We use the latter approach for the examples in the next section. In that case, each basis function is assigned its column index (ranging from 1 to n) of the table of tree-structured subspaces rather than an index of importance. In general, however, the MDL criterion [128] should be a good candidate for obtaining the optimal k, and our future research will address this problem.

As for an extension to a library of orthonormal bases, we can pick the LRB giving the smallest prediction error among the LRBs constructed from dictionaries in the library.

Remark 5.2. A useful variant of this LRB algorithm is to consider nonlinear operations (such as taking log of absolute values, squaring, or thresholding) of the expansion coefficients prior to the subspace evaluation (5.3). In particular, supplying the squares of the coefficients

corresponds to regressing the response function on the time-frequency energy distributions of the input signals. We call the LRB selected from the squares of the coefficients "LRB2."

5.4 Examples

It is possible to use the LRB algorithm for classification by replacing regression methods g and g_js by classification methods (e.g., CTs), and regression errors by misclassification rates. In this section we apply this LRB-based classification to the examples shown in the previous chapter and compare the results. A genuine regression example using a real dataset is described in the next chapter in depth.

We first applied the LRB algorithm to the dataset described in Example 4.6 and generated the following table of misclassification rates. The lowest misclassification rate was

Method	Error rate (%) Training	Test
FCT on LRBF	4.33	24.33
PCT on LRBF	17.00	25.10
FCT on LRBP	4.33	**22.13**
PCT on LRBP	16.67	25.00
FCT on LRB2F	5.00	23.00
PCT on LRB2F	21.67	27.50
FCT on LRB2P	4.00	25.30
PCT on LRB2P	17.00	25.10

Table 5.1: Misclassification rates of Example 4.6 using the LRB methods. The same QMF, C06, was used to generate the tree-structured expansion coefficients. In Method column, FCT and PCT denote the full and pruned classification trees, respectively. LRBF and LRBP represent the coordinates selected by the subspace evaluation using the FCTs and PCTs on the expansion coefficients, respectively. Thus, e.g., FCT on LRBP means that the full classification tree grown on all the coordinates selected by Algorithm 5.1 with pruned classification tree as g_j. Similarly LRB2F and LRB2P are the LRB coordinates selected by the subspace evaluation using the FCTs and PCTs on the squares of the expansion coefficients. The smallest error on the test dataset is shown in bold font.

obtained by the fully-grown RT on all the LRBP coordinates. Here LRBP means the LRB

Method	Error rate (%)	
	Training	Test
FCT on LRBF	2.00	5.73
PCT on LRBF	4.67	8.20
FCT on LRBP	2.00	6.47
PCT on LRBP	2.67	**5.40**
FCT on LRB2F	2.00	5.73
PCT on LRB2F	4.67	8.20
FCT on LRB2P	2.33	9.60
PCT on LRB2P	4.67	9.27

Table 5.2: Misclassification rates of Example 4.7. The same QMF, C12, was used to generate the tree-structured expansion coefficients. The abbreviations are exactly the same as Table 5.1. The smallest error on the test dataset is shown in bold font.

selected by Algorithm 5.1 with pruned CT as g_js for subspace evaluation. Figure 5.1 shows this best tree. This tree selected 11 LRBP coordinates for the classification out of 32 possible coordinates. Figure 5.2 shows these selected coordinates as well as the subspace pattern of the LRB. Comparing with the corresponding table and figures of the LDB methods in the previous chapter, we observe the following:

- The misclassification rates except the one by the LDA-based classification in Table 4.1 are comparable.
- Seven functions out of 11 selected LRB functions have larger scale features (from the subspaces $\Omega_{4,0}, \Omega_{4,1}, \Omega_{3,1}, \Omega_{3,2}$) than the top 5 LDB functions shown in Figure 4.3 (b) (from $\Omega_{2,0}$). In fact the LRB functions try to combine the elementary triangular waves h_1, h_2, h_3 of Example 4.6, e.g., the LRB function #6 has two major positive peaks around the functions h_1 and h_2 and a major negative peak around h_3.

Next we applied the same procedures to the dataset of Example 4.7. The misclassification errors are summarized in Table 5.2. In this example, the pruned CT on the LRBP coordinates gives the lowest misclassification rate. Figure 5.3 shows this best tree. This

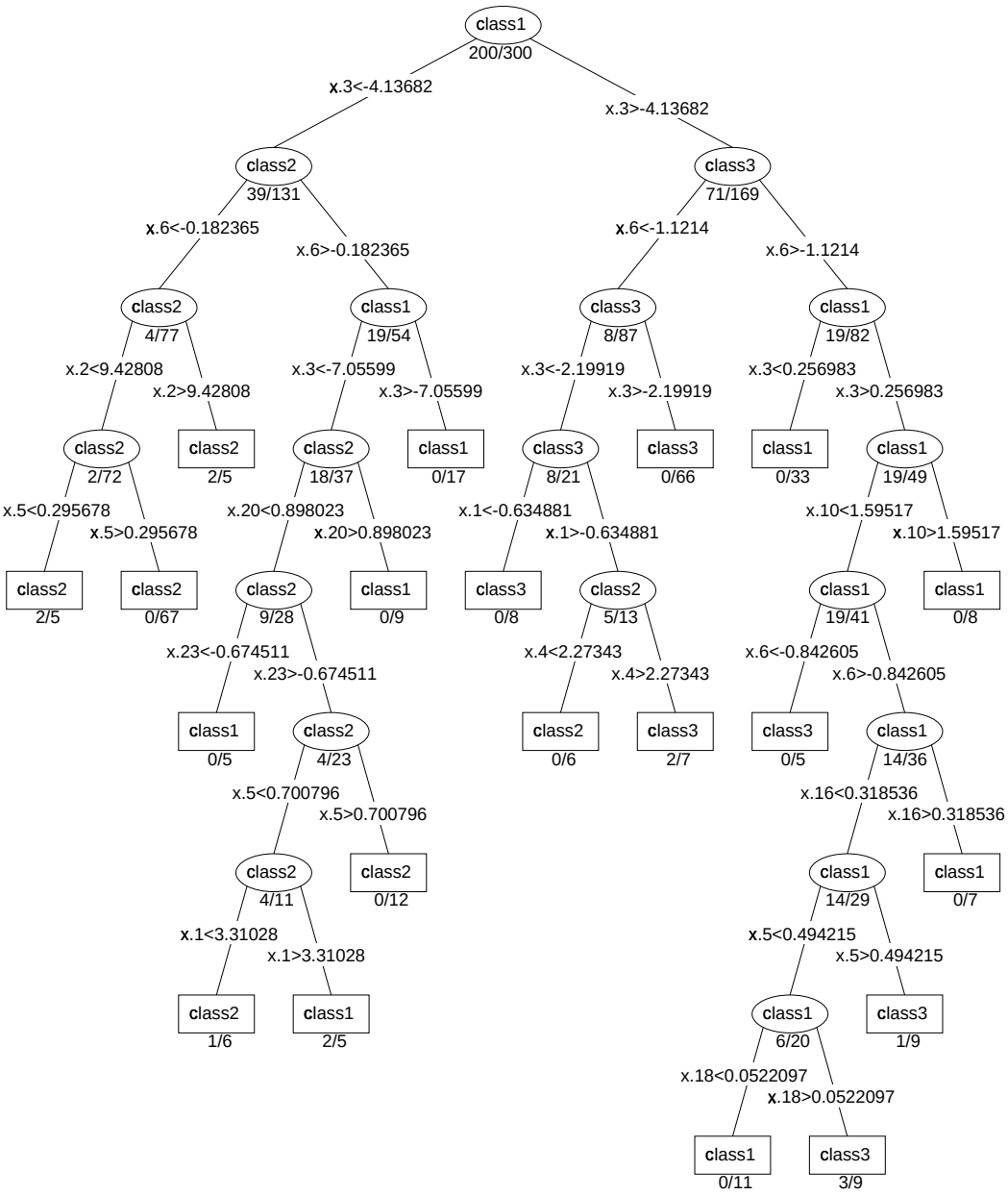


Figure 5.1: The full CT giving the lowest misclassification rate using the LRB methods on the dataset of Example 4.6. This tree is grown on the signals represented in the LRBP coordinates.

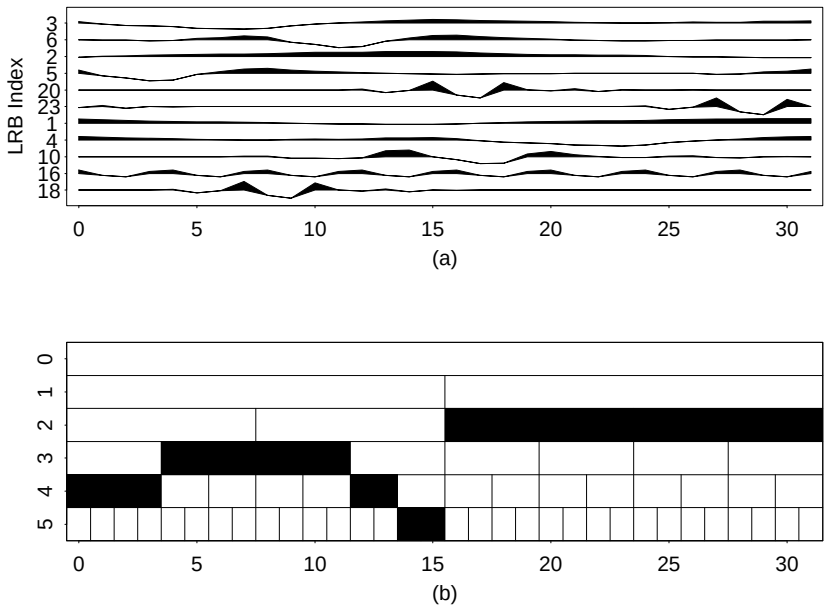


Figure 5.2: (a) The LRB functions used in the CT shown in Figure 5.1. (b) The selected subspaces as the LRB.

best tree selected 4 LRBP coordinates for the classification out of 128 possible coordinates. Figure 5.4 shows these selected coordinates as well as the subspace pattern of the LRB. Comparing with the corresponding table and figures of the LDB methods in the previous chapter, we observe the following:

- Both misclassification rates are comparable.
- Two basis functions from the subspace $\Omega_{4,0}$ were selected by both methods.

From these two classification examples, it is difficult to judge which method is superior. For the genuine regression problems where the data does not have a natural association with classes or categories, the LDB methods cannot be used. We study this type of regression problem in the next chapter.

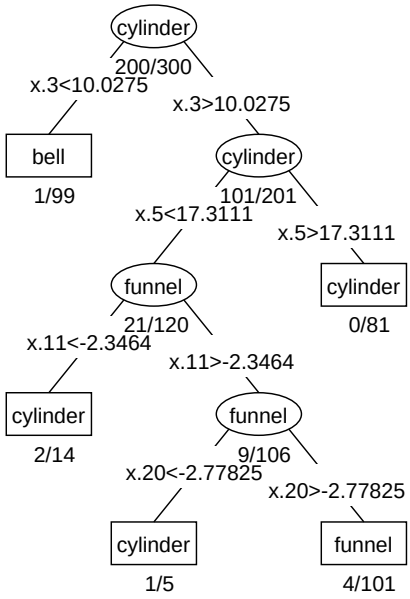


Figure 5.3: The pruned CT giving the lowest misclassification rate using the LRB methods on the dataset of Example 4.7. This tree was initially grown on the signals represented in the LRBP coordinates and then pruned by the MDL-based pruning algorithm.

5.5 Discussion

In this section, we discuss some of the related methods proposed by others and a possible extension of our methods.

In [67] Guo and Gelfand proposed an idea to improve the CT by using a small neural network at each node in the tree. By the use of neural networks, their method allows one to split the input signal space *nonlinearly* to gain more class separability than the CTs using coordinate-wise splits and the linear splits. The fundamental difference between our approach and their approach is that they do not use the local information in the time-frequency plane; they claim that they can extract the local features but in our opinion, local only in the sense of features mappable from the original coordinates by the neural networks they use. For example, it is essentially impossible to use the local frequency information by

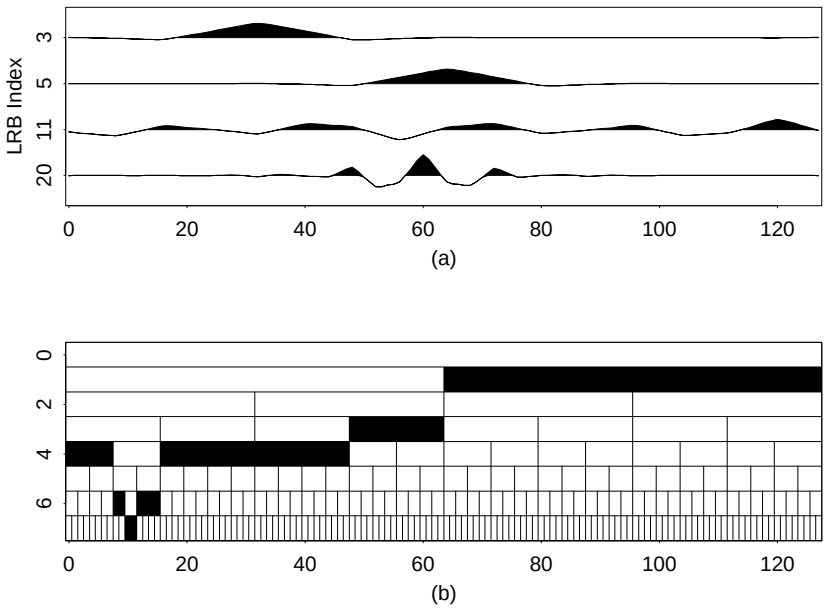


Figure 5.4: (a) The LRB functions used in the CT shown in Figure 5.3. (b) The selected subspaces as the LRB.

their algorithm unless one supplies the input signals represented in the Fourier basis. They applied their algorithm to the same example as we used, i.e., Example 4.6 and achieved the misclassification rate about 18% using 250 training samples and 5000 test samples. This is comparable with our results. In fact, their result is worse than the best LDB method and is better than the best LRB method. The key advantage of the use of neural networks is their ability to approximate nonlinear relationships between inputs and outputs. It is an interesting exercise to compare their result with the performance of the LRB method using a neural network as a regression scheme g which more directly address the extraction of the local features using neural networks. We expect that our LRB method will improve significantly if nonlinear interactions among the local basis coordinates at each subspace are allowed via neural networks. Similar approaches to Guo and Gelfand's method have been reported in the field of speech recognition [141], [119].

Regarding the use of the wavelet basis functions in the neural networks, there exist several articles [160], [4], [113], [15], [142]. The works described in [160], [4], [113], [15] are essentially the same: they suggest replacing the activation functions (three popular choices are 1) the step function, 2) the sigmoid function, and 3) the Gaussian function) by wavelet basis functions for their multiresolution capabilities. Their approach address the neural networks themselves and incorporate wavelet bases for improving the classification/regression ability of the neural networks. Our approach is fundamentally different; we address selecting the best possible basis functions localized in the time-frequency plane as feature extractors from a library of bases using the regression scheme which gives the best performance for the problem at hand. A neural network is simply one of the regression schemes from our viewpoint: if there is a better regression scheme, we use it instead. We do not attach any inherent importance to the neural network although we appreciate their flexibility and nonlinear capability.

The work of Szu et al. [142] is slightly different from the above-mentioned works; they input the wavelet coefficients of the training signals to the neural network having the sigmoidal activation function. More precisely, let $\psi(t)$ be a wavelet mother function and $\psi_{(a_k,b_k)}(t) = \psi((t-b_k)/a_k)$, i.e., a translated and dilated version of ψ. Then they compute the inner products of the input signals $\{\boldsymbol{x}_i\}$ with $\boldsymbol{\psi}_{(a_k,b_k)} = (\psi_{(a_k,b_k)}(1), \ldots, \psi_{(a_k,b_k)}(n))$ and supply them to the sigmoidal nonlinearity $\sigma(\cdot)$ with some weights $\{w_k\}$, i.e., the output from the network can be written as

$$v_i = \sigma\left(\sum_{k=1}^{K} w_k \boldsymbol{x}_i^T \boldsymbol{\psi}_{(a_k,b_k)}\right),$$

for $i = 1, \ldots, N$. The whole exercise is to compute (w_k, a_k, b_k) so as to minimize the classification error measured by $\mathcal{R} = (1/2)\sum_{i=1}^{N}(y_i - v_i)^2$ where y_i is the desired classifier output, i.e., $y_i = 1$ if $\boldsymbol{x}_i$ belongs to class 1 and $y_i = 0$ if $\boldsymbol{x}_i$ belongs to class 2 (they considered the binary classification case there). In particular, they use the Morlet mother wavelet, $\psi(t) = \cos(\omega t)\exp(-t^2/2)$ and compute the parameters (w_k, a_k, b_k) using the conjugate

gradient method. We can see a clear difference from our philosophy. They try to optimize the translation-dilation parameters for the fixed wavelet mother function whereas our approach uses many different wavelet bases (including wavelet packets/local trig. bases) for a set of predefined translation-dilation parameters (dyadic dilations and associated circular translations). The potential problems of their approach are: 1) how to select the mother wavelet function in the first place, and 2) the conjugate gradient method can get stuck in the local minima of the objective function $\mathcal{R}$.

As for an extension of the LRB method, instead of restricting only one regression family g_j to evaluate subspaces at level j, we may consider many different regression methods $g_{1,j}, \ldots, g_{m,j}$, and take a method which gives the smallest regression error for each subspace. Thus, if we register the best regression method as well as the smallest regression error at each subspace, then we obtain the LRB with a list of the regression methods. This strategy makes sense since each regression method has pros and cons; e.g., RTs are relatively good for nonlinear relationships between input signals and responses, but are less accurate for the problems with linear relationships where the linear regression works much better, etc. We will address the use of multiple regression methods in our future project.

Finally, we give our thoughts on the LRB method versus the LDB method. As we can easily see, the LRB method is more flexible and general than the LDB method. But it is more computationally intensive than the LDB method; a regression method g_j has to be invoked at each subspace in the tree-structured subspaces. Which method to be used really depends on the problem at hand. For the general regression problem, the choice is definitely the LRB method. For classification problems and certain regression problems mentioned in Remark 4.5, the LDB method may be a first choice because of its computational efficiency.

5.6 Summary

In this chapter, we have proposed a method to select a complete orthonormal basis [*local regression basis*(LRB)] from a library of orthonormal bases which is suitable for regression problems. This method uses prediction error (computed by a specified regression scheme) as a measure of the goodness of each subspace so that the regression scheme is integrated into the basis selection mechanism. We have shown that the LRB method can also be used for the classification problems and have examined its performance using Examples 4.6 and 4.7 of the previous chapter. The results are comparable with those of the LDB method. The LRB method is more flexible and general than the LDB method; however, it is more computationally intensive than the LDB method.

Chapter 6

Extraction of Geological Information from Acoustic Well-Logging Waveforms Using LDB and LRB Methods

6.1 Introduction

In this chapter, we apply the LDB and LRB methods developed in the previous chapters to a real geophysical regression problem. The problem we consider here is to infer some geological properties of subsurface formations from measured acoustic waveforms which propagated through these formations. The acoustic measurements have been used in geophysical well logging for a long time to infer petrophysical properties of subsurface formations [144]. These measurements consist of the following procedure. First, an acoustic pulse is generated at the transmitter of a measurement tool lowered down in a borehole. Then, this pulse propagates through the surrounding formations. Finally, the pressure field is recorded at the receiver of

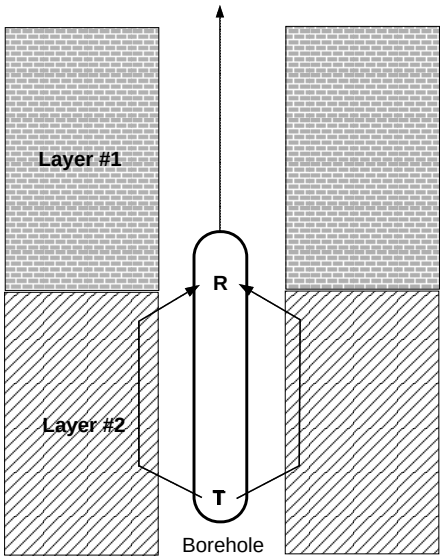


Figure 6.1: An illustration of a simple sonic tool. The tool is represented by the ellipse. The symbols T and R denote a transmitter and a receiver equipped in the tool, respectively. A typical distance between the transmitter and the receiver is 9 feet. An actual tool normally has two or eight receivers to compensate the borehole effects. The arrows connecting the transmitter and the receiver simply illustrate raypaths of P or S wave components. The recorded waveform data is digitized and sent to the surface processing unit through the cable attached to the tool.

the same measurement tool. This process is repeated until the tool is drawn up to a certain depth level. See the illustration of this type of tool in Figure 6.1. The main purpose of this measurement is to:

- Calibrate the reflection seismic imaging algorithms.
- Deduce the lithology information from the acoustic/elastic properties of subsurface formations.
- Assess the mechanical property of formations including fracture detection.

A typical recorded waveform, as shown in Figure 6.2, consists of three types of localized wave components: a refracted compressional (or longitudinal or primary) wave called P wave, a refracted shear (or transverse or secondary) wave called S wave, and a guided

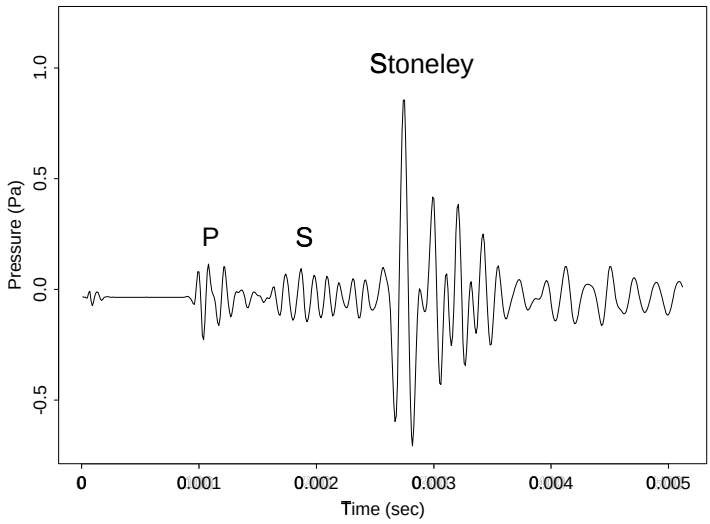


Figure 6.2: A typical acoustic waveform recorded in the downhole. The surrounding subsurface formation consists of shale in this case. The Stoneley wave component normally has a dominant energy.

surface wave called the Stoneley wave. The P and S waves follow paths that minimize the traveltimes between the transmitter and the receiver. The Stoneley wave, which is guided by the fluid-rock interface, travels more slowly than the two refracted waves and is the dominant event at later times in the waveform. Traditionally, velocities of these three wave components with or without amplitudes of these components have been used to infer geological information of the formations. These quantities are related to the formation properties such as porosity, mineralogy, grain contacts, and fluid saturation etc.; see e.g., [154], [109], [158] and references therein for more details.

The velocity and amplitude information of a particular wave component is just a part of the information contained in the entire waveform shape since the velocity can be computed from its arrival time (the starting time position) and the amplitude is simply the maximum value of the wave component. It is an extremely difficult task to validate the entire shape information by the exact mathematical modelling and computer simulation

because of the complexity of: 1) the material, i.e., the subsurface formations of various mineralogy with varying pore spaces containing different types of fluids, 2) the geometry, i.e., varying diameters of the borehole and the rugosity of the borehole wall, and 3) the physics, i.e., acoustic/elastic wave propagation phenomena in such formations. These are the reasons why there have been only a few attempts to fully utilize the waveform shape information [71], [75], although it has been long recognized the relationships between the shapes of the waveforms and the types of rocks.

The first systematic method to use the waveform shape information is due to Hoard [71].His method estimates lithologic information or attributes (i.e., porosity, volume percentages of various rocks such as sandstone, shale, and limestone, etc.) from the full acoustic waveforms combining with several other geophysical measurements such as the natural radioactivity and resistivity of the formations. In his study, the lithologic information was obtained by careful study of all available data including drill cuttings, core samples, and other geophysical measurements. After selecting the training dataset, the clusters in the input signal space are identified using the graph-theoretic clustering algorithm [63, pp.539–541]. Any clustering method requires one to specify the similarity measure among input vectors, and the standard Euclidean norm was used in his approach. This is, however, a global measure of similarity: a slight time shift in the waveforms creates a large distance in this norm, and the large amplitude portions "mask" small features. Because of this problem, the envelopes of the waveforms are computed by their Hilbert transforms, and then the log values of the envelopes are used as inputs to the clustering algorithm. After identifying the clusters, a mean vector and mean lithologic attributes are computed for each cluster. Finally, for each vector in the test dataset, the distances between that vector and the mean vectors of the clusters are compared and the lithologic attributes of the closest mean vector is taken as the test vector's lithologic attributes. Although he claims that it works well as long as the test dataset comes from the similar geological environment such as

the wells near from the training well, there are several problems with this approach. Since this simply uses the entire envelope information with the Euclidean norm as a similarity measure, it is very difficult to interpret the results: which wave component is responsible for what lithologic attributes? Also, the training process is computationally extremely expensive since this approach does not reduce the dimensionality of the problem. Finally, it is not too clear why the clustering technique (i.e., unsupervised learning) is used rather than classification/regression techniques (i.e., supervised learning methods).

A few years later, Hsu recognized the importance of the relationships between individual wave components and lithologic information, and proposed a different approach [75]. His approach first extracts the wave components in the dataset separately by aligning the arrival times of each component[1] throughout the dataset and segmenting each component with an appropriate time window. This process generates three sets of vectors corresponding to P, S, and Stoneley wave components. Then for each wave component set, the Karhunen-Loève transform (KLT) is applied and a few largest eigenvectors are obtained. Because of the alignment, the first and second eigenvectors account for major part (in his example, more than 90 %) of the total energy of the set. Then all vectors in the set are projected in this coordinate system (spanned by these two eigenvectors) and the structures of the point clouds in this coordinate system are examined. In his example, the projected data points tended to have clusters depending on the formation rock types (i.e., sandstone, limestone etc.) where they propagated and this tendency was more pronounced in the Stoneley wave set than in the P wave set. (He could not use the S wave components because the S waves did not exist for certain depth levels due to the soft formation conditions.) Although his findings are interesting and his method is easier to interpret than that of Hoard, it is still computationally very expensive due to the use of the KLT as we mentioned in Chapter 2.

[1] This alignment is normally done in a semi-automatic way: the user first defines an appropriate time window, then the computer tries to track the first zero-crossing of the wave component within the time window. Since the positions of these zero-crossing vary (sometimes wildly) from trace to trace, the manual editing is sometimes necessary.

(He proposed a computationally faster method to simulate the first and second eigenvectors. Such a method, however, may be only applicable to the specific example he used. See [75] for more details.) Moreover, the features extracted may not carry subtle discriminatory information since a few largest KL basis vectors only capture the major and common features in the dataset. His method is again considered as a clustering or an unsupervised learning technique rather than classification/regression.

Although our purpose here is similar to the above studies, our approach is fundamentally different from theirs. We use the regression techniques developed in the previous chapters which use the *local* information in the time-frequency plane. Similarly to the genuine classification problems, the key is how to extract useful features from input signals and reduce the dimensionality of the problem at hand since this again enhances the conventional regression methods in both efficiency and accuracy.

6.2 Data Description and Problem Setting

In this exercise, we use the acoustic waveforms recorded at a certain well. We have 3012 waveforms recorded at every 0.5 foot depth interval. Each waveform consists of 512 time samples with sampling rate of 10^{-5} second. Along with the waveforms, we also have lithologic information around each receiver location which was computed from various geophysical measurements using the volumetric analysis methods described in [118], [22]. In this study, we use the volume fractions of quartz (main constituent of sandstone), illite (a type of clay which is a main constituent of shale in this area), and gas as the lithologic information. We note that no acoustic/elastic information was used to generate the lithologic information in this case. The region where the well is located mainly consists of sandstone-shale sequences, i.e., this is a relatively simple geologic setting. Most sandstone layers contain either gas or water. Figure 6.3 shows the dataset under study. From the volume fraction curves in Figure 6.3, we observe that

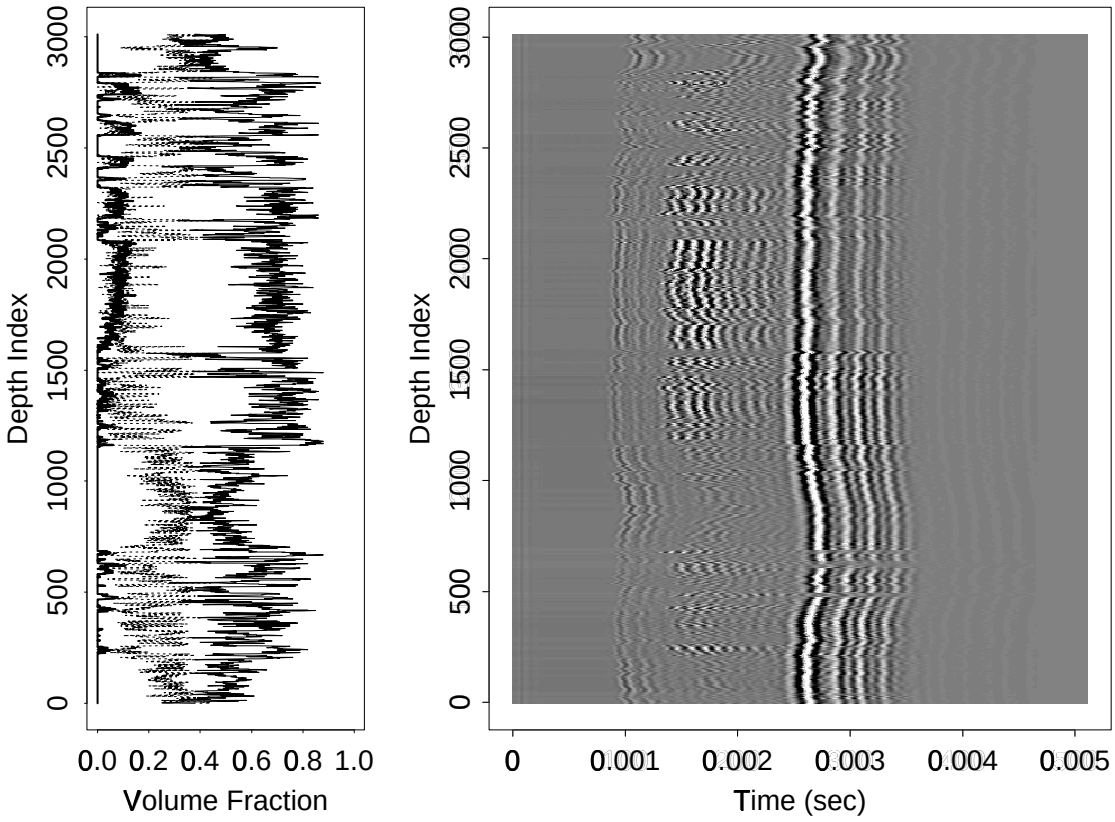


Figure 6.3: These figures show the whole dataset used in this study. The left figure shows three curves representing volume fractions of quartz (solid line), illite (dotted line), and gas (thick solid line). The right figure shows the acoustic waveforms recorded at the corresponding depth levels as a gray scale image. The depth index 0 corresponds to the deepest level.

1. There is a thick sandstone layer containing gas around the depth index ranging from 1600 to 2100.

2. There is a shale layer around the depth index ranging from 700 to 1100.

3. There are alternating sandstone-shale sequences above the thick sandstone layer and below the shale layer described above.

Let us call the waveforms propagated through sandstone layers "sand waveforms" and those propagated through shale layers "shale waveforms" for short. We observe the following waveform features from Figure 6.3:

1. The S wave components in the sand waveforms have much stronger energy and faster speed than those in the shale waveforms.

2. Velocities of the P wave components in the sand waveforms are higher than those in the shale waveforms.

3. Velocities of the Stoneley wave components in the shale waveforms are lower than those in the sand waveforms except those in the bottom 200 levels.

The physics of wave propagation suggests that in fact the P and S velocities are sensitive to the fluid content and the mineralogy, and the Stoneley wave velocity is sensitive to the permeability of the formations as well as the borehole conditions such as rugosity and diameters of the borehole [154], [109], [158]. The exceptionally high velocities of the Stoneley wave components in the bottom region mentioned above may be due to the borehole conditions. Because of the sensitivity to the borehole conditions, we smoothly taper off the Stoneley wave component from each waveform and only consider the earlier part of the waveforms.

As a training dataset, we selected the data from the most representative regions, i.e., 201 contiguous depth levels from the main shale layer and 201 depth levels from three different sandstone layers as shown in Figure 6.4.

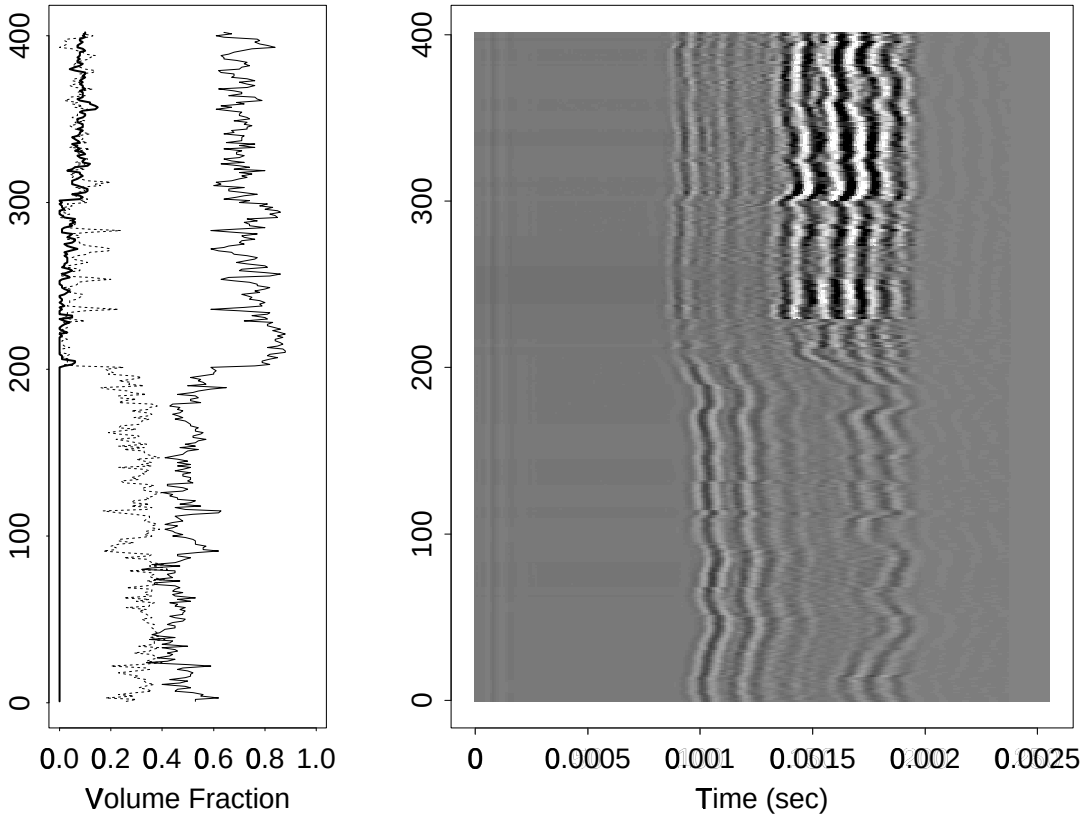


Figure 6.4: These figures show the training dataset selected for this study. Bottom 201 recordings correspond to the shale dominant region (depth index ranging from 800 to 1000 in Figure 6.3). Top 201 recordings correspond to the sandstone regions (depth indices ranging from 1160 to 1188 [water sand], from 1340 to 1410 [water sand], and from 1780 to 1880 [gas sand]). Three curves in the left figure again correspond to the volume fractions of quartz (solid line), illite (dotted line), and gas (thick solid line). The acoustic waveforms shown in the right figure have been smoothly tapered off to eliminate the Stoneley wave components.

The purpose of this exercise is to examine: 1) how accurately we can predict (in an automatic manner) the volume fractions of quartz, illite, gas at each depth level from the acoustic waveform propagated around that level without assuming the detailed physical models, and 2) what features in the waveform are important for estimating these volume fractions. Using this training dataset, we proceed to the regression analysis using LDB and LRB with CART as a basic regression tool.

6.3 Results

Since the velocity information (i.e., the locations of the wave components in the time domain) is important in this study, a natural choice of the time-frequency decomposition is the local trigonometric transforms rather than the wavelet packets. Hence, we use the local sine transform (LST) in this study. In the following, the test dataset means the whole dataset excluding the training dataset, i.e., the test dataset consists of 2610 waveforms and the corresponding lithologic attributes. Also, we simply say "volume" instead of volume fraction for short.

6.3.1 Analysis by LDB

First we computed the LDB assuming that the training dataset consists of two classes, i.e., shale class and sandstone class. In reality, there exist layers of "shaly sand", a mixture of sandstone and shale, in this region, which are difficult to classify into either sandstone or shale in a clear manner. This assumption, however, is a good starting point to examine what features in the waveforms carry the discriminatory information between sandstone and shale. In this study, the symmetric relative entropy (4.2) was used as the discriminant measure. The order of importance of individual LDB functions was computed by (4.7). Top 50 basis functions and selected subspaces are displayed together in Figure 6.5. We can observe that the most discriminant basis functions are the localized wiggles around the P

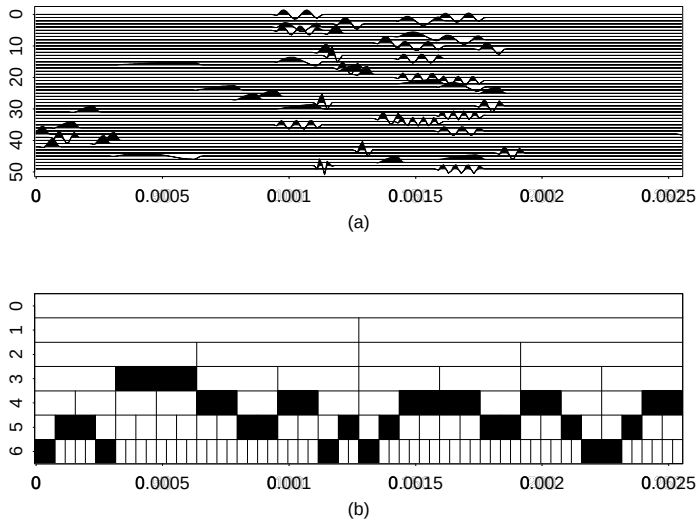


Figure 6.5: (a) Top 50 LDB functions using LST. (b) Selected subspaces as the LDB.

and S wave components (around $t = 0.001$ and $t = 0.0015$, respectively). Then, for each lithologic attribute, we applied the CART procedure to the training waveforms represented in the standard Euclidean coordinates and then those represented in the LDB coordinates. In each case, the full tree was grown first, and the regression (or equivalently prediction) errors for the training dataset and the test dataset were obtained. Finally, the full tree was pruned by the MDL-based algorithm described in Appendix A and the regression error was computed. We adopt the relative ℓ^2 error, i.e., $(\sum(y_i - d(\boldsymbol{x}_i))^2 / \sum y_i^2)^{1/2}$, as a measure of regression error since this is used as the deviance in the regression tree (RT) procedure in S [10], [23] and S-PLUS [140] which we use for all the experiments in this chapter. These regression errors are summarized in Table 6.1.

For the quartz volume, the best result (in this table) was obtained by using the pruned tree regression on all the LDB coordinates. This pruned tree is plotted in Figure 6.6. Only four LDB coordinates out of 256 are used in this tree. These LDB functions are displayed in Figure 6.7. Three LDB functions with indices $(127, 27, 85)$ are located around

Method	Quartz		Illite		Gas	
	Training	Test	Training	Test	Training	Test
FRT on STD	0.06985	0.2641	0.1630	0.6770	0.2132	0.8616
PRT on STD	0.09597	0.2602	0.2083	0.6648	0.2390	0.8586
FRT on LDB50	0.07275	0.2499	0.1689	0.6196	0.2180	0.8688
PRT on LDB50	0.09799	0.2532	0.2314	0.6186	0.2825	**0.8349**
FRT on LDB	0.06988	0.2574	0.1629	0.5948	0.1986	0.8818
PRT on LDB	0.09697	**0.2423**	0.2117	**0.5843**	0.2437	0.8859

Table 6.1: The prediction errors on the lithologic attributes using the tree-based regression with the waveform data represented in the standard Euclidean coordinates and the LDB coordinates. In Method column, FRT and PRT denote full regression tree and pruned regression tree, respectively. STD, LDB50, LDB denote the waveforms represented in the standard Euclidean coordinates, the top 50 LDB coordinates, and all the LDB coordinates, respectively. The smallest errors in the test data columns are displayed in bold font.

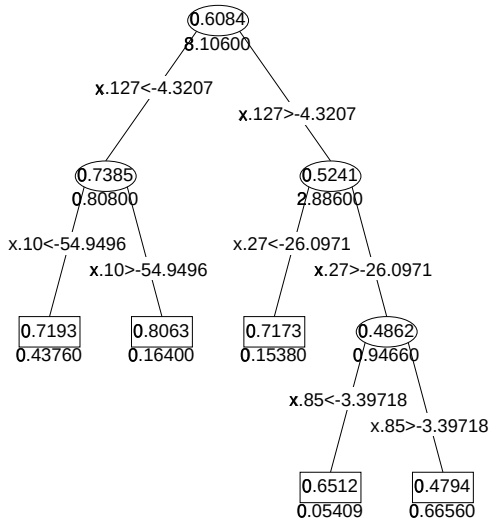


Figure 6.6: The pruned regression tree for the quartz volume. This tree was initially grown on the complete waveforms represented in the LDB coordinates. Nodes are represented by ellipses (interior nodes) and rectangles (terminal nodes/leaves). The node labels are the predicted values of the quartz volume. The numbers displayed under each node represent the residual sum of squares within that node. The splitting rules are displayed on the edges connecting nodes and x.127 means the basis coordinate of index 127.

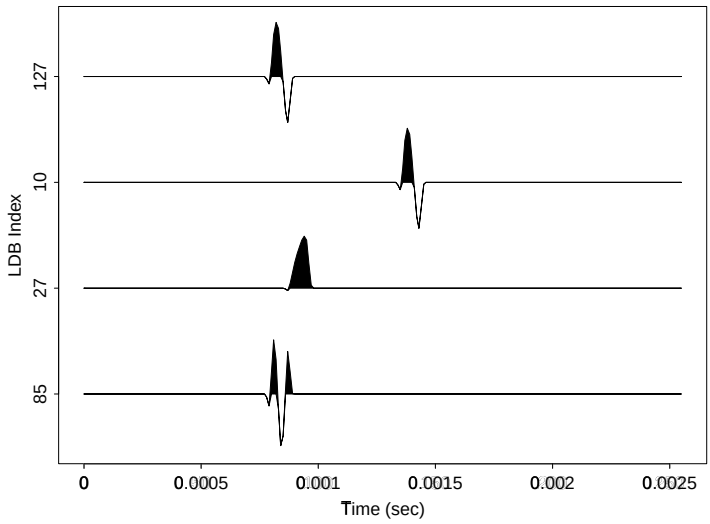


Figure 6.7: The LDB functions used in the pruned regression tree for the quartz rate shown in Figure 6.6. These are displayed in the depth-first search manner in the tree.

the P wave components of the sand waveforms and one LDB function #10 is located around the S wave components of the sand waveforms. (We use $\#k$ to denote index k for short.) Examining the tree in Figure 6.6 reveals that the combination of the LDB functions #127 and #10 is responsible for high quartz volume: the tree says, "If the LDB coordinate #127 is less than -4.3207, then check the coordinate #10. If that is less than -54.9496, then assign 0.7193, otherwise assign 0.8063 as the quartz rate." This observation agrees with the physics of wave propagation described in [154], [109], and [158].

Remark 6.1. The indices $(127, 10, 27, 85)$ used in the best tree are in fact the order of importance in terms of (4.7). These indices change when a different ordering scheme is used; e.g., with the Fisher index (4.8), they are $(1, 21, 112, 3)$. This would suggest, at least in this example, letting the regression method select the best LDB coordinates by supplying all the LDB coordinates rather than worrying about the order of importance and selection of the coordinates prior to applying the regression method. (This is only applicable if the regression method has a built-in capability of selecting the best coordinates. CART is one of

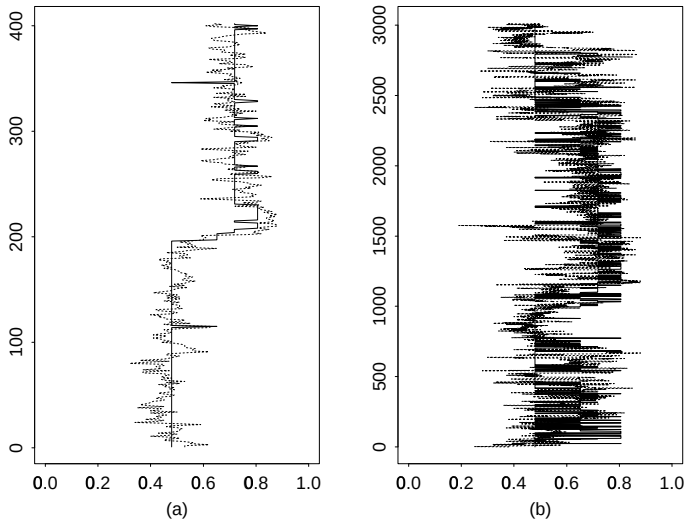


Figure 6.8: The prediction of the quartz volume by the pruned regression tree shown in Figure 6.6. (a) The training dataset. (b) The whole dataset. In both cases, the solid line and the dotted line correspond to the predicted quartz volume and the original quartz volume, respectively.

such regression methods.) The LRB method completely avoids this problem at the expense of the computational time.

Figure 6.8 compares the original and the predicted quartz volume for the training dataset and the whole dataset (including the training dataset) by this pruned tree regression.

For the illite volume, the same procedure, i.e., the pruned tree on all the LDB coordinates gives the best result in this table. This tree in Figure 6.9. Again only four LDB coordinates are used in this tree, and three out of which are exactly the same as the quartz case. This may be explained by the "duality" of the response curves of quartz and illite volumes in Figures 6.3 and 6.4. The illite volume curve is roughly a flipped version (around volume fraction 0.4) of the quartz volume curve. The corresponding basis functions are displayed in Figure 6.10. The basis function #222, located rather late in time, seems

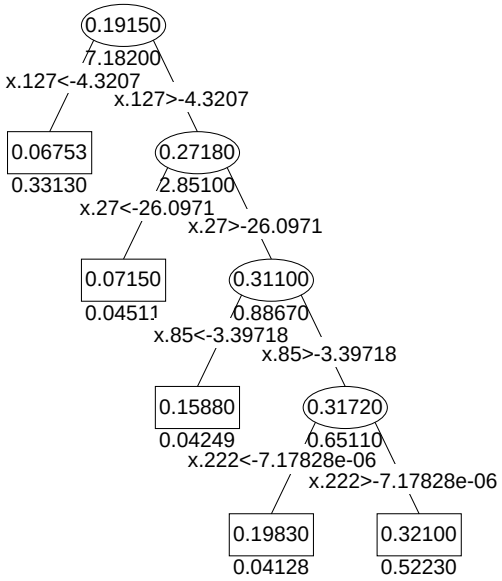


Figure 6.9: The pruned regression tree for the illite volume. This tree was initially grown on the complete waveforms represented in the LDB coordinates. Notice that three coordinates are the same as the ones in the tree for the quartz volume.

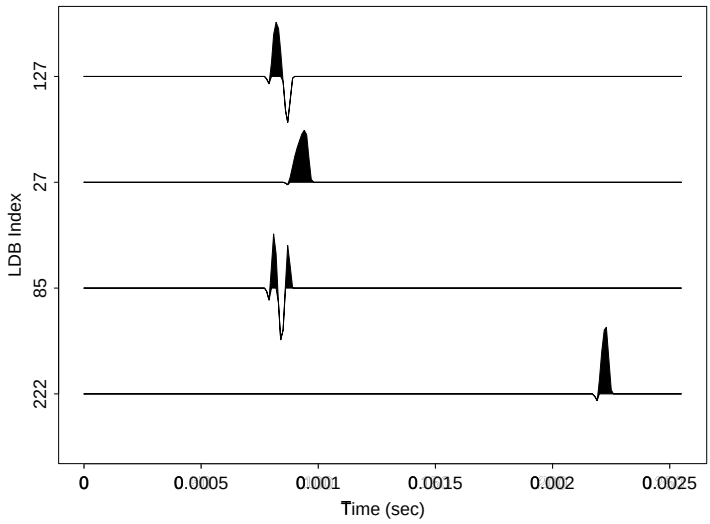


Figure 6.10: The LDB functions used in the pruned regression tree for the illite rate shown in Figure 6.9. These are displayed in the depth-first search manner in the tree.

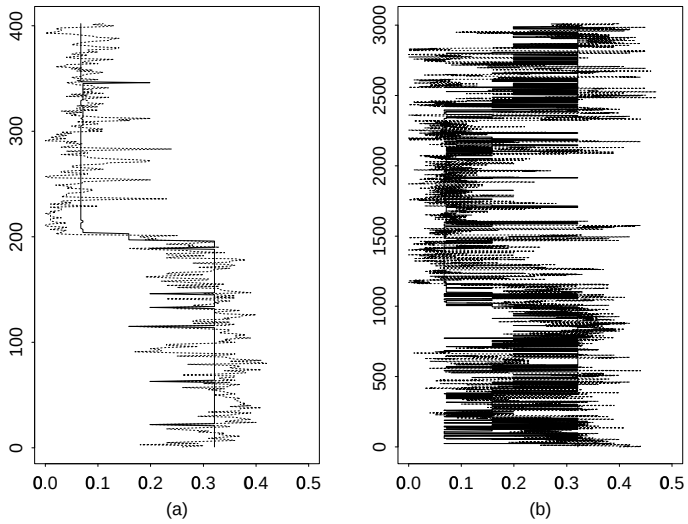


Figure 6.11: The prediction of the illite volume by the pruned regression tree shown in Figure 6.9. (a) The training dataset. (b) The whole dataset.

responsible for high illite volume (see Figure 6.9); however, it is not at all clear whether that is the case since the threshold used for this coordinate is very small (-7.17828×10^{-6}), and the function is located around the tapered region. On the other hand, removing this coordinate from the tree does not reduce the regression error either (the error on the test dataset becomes 0.5885 instead of 0.5843). Figure 6.11 shows the prediction of the illite rate for the training dataset and the whole dataset by this pruned tree regression.

Finally for the gas volume, the pruned regression tree on the top 50 LDB coordinates gives lower regression error than the one on all the LDB coordinates does. This tree is plotted in Figure 6.12. The corresponding basis functions are displayed in Figure 6.13. The majority of the selected LDB functions are located around the S wave components. Examining the tree in Figure 6.12 carefully, we observe that the nodes in the right branch from the root node has higher gas volume than those in the left branch. In particular, the LDB functions of indices $(23, 17, 11)$ are responsible for high gas volume. Out of these three basis functions, the functions $(23, 11)$ are located around $t = 0.0015$ where in fact the S

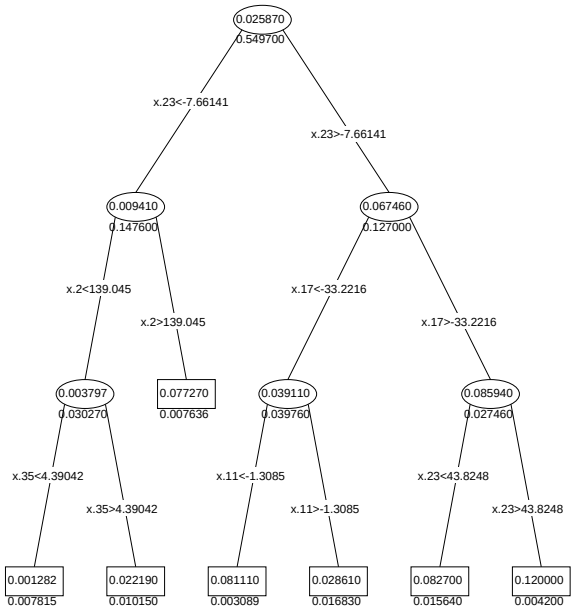


Figure 6.12: The pruned regression tree for the gas volume. This tree was initially grown on the top 50 LDB coefficients of the waveforms.

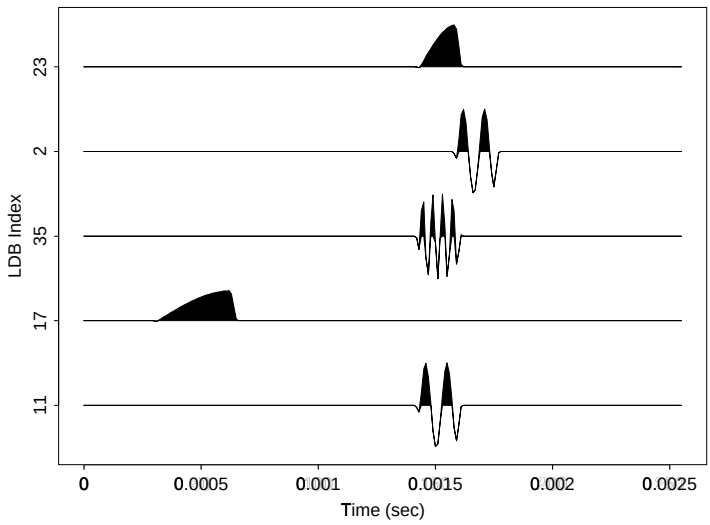


Figure 6.13: The LDB functions used in the pruned regression tree for the gas volume shown in Figure 6.12. These are displayed in the depth-first search manner in the tree.

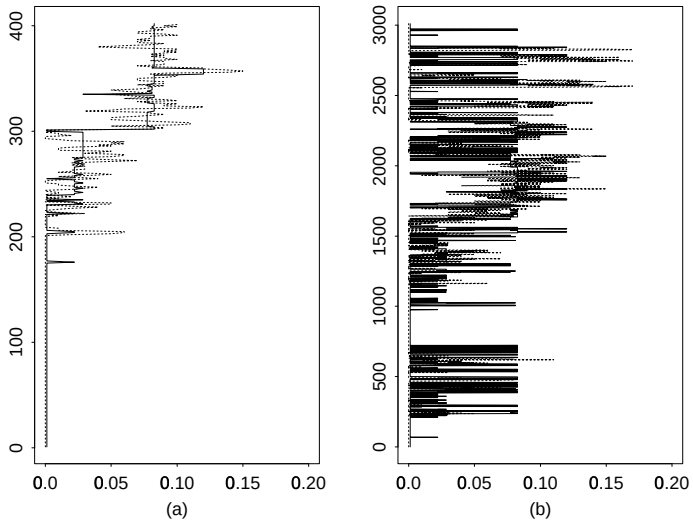


Figure 6.14: The prediction of the gas volume by the pruned regression tree shown in Figure 6.12. (a) The training dataset. (b) The whole dataset.

wave components with high amplitudes and relatively low frequency bands exist in the sand waveforms, especially the "gas sand" waveforms as we can see Figure 6.4. Note that the basis function #35 in Figure 6.13 also located around $t = 0.0015$; however, its frequency band is higher than functions $(23, 11)$. In fact the function #35 is used in the left branch of the tree in Figure 6.12 and is responsible for low gas volume if combined with the low value of the LDB coordinate #2. These observations again agrees with the explanation from the physics [154], [109], [158]. Figure 6.14 shows the prediction of the gas rate for the training dataset and the whole dataset by this pruned tree regression.

6.3.2 Analysis by LRB

Now we describe the results using the LRB methods. Without assuming class assignments, four different LRBs were computed using the combinations of the regression method for the subspace evaluation (Full RT or Pruned RT) and the coefficients mapping (original expansion coefficients or the squares of them). For each LRB, all the basis coordinates were

Method	Quartz		Illite		Gas	
	Training	Test	Training	Test	Training	Test
FRT on LRBF	0.07166	0.2744	0.1593	0.6144	0.1890	0.8614
PRT on LRBF	0.09635	0.2517	0.2073	**0.5949**	0.2174	0.8936
FRT on LRBP	0.07423	0.2481	0.1611	0.6097	0.2075	0.8233
PRT on LRBP	0.09825	**0.2356**	0.2078	0.5957	0.2442	**0.8108**
FRT on LRB2F	0.06931	0.2534	0.1699	0.6158	0.1899	0.8702
PRT on LRB2F	0.09518	0.2523	0.2177	0.6061	0.1956	0.8727
FRT on LRB2P	0.06581	0.2550	0.1584	0.6158	0.1911	0.8628
PRT on LRB2P	0.09198	0.2475	0.2044	0.6063	0.2037	0.8675

Table 6.2: The prediction errors on the lithologic attributes using the tree-based regression with the waveform data represented in the LRB coordinates. The smallest errors in the test data columns are displayed in bold font. In Method column, LRBF and LRBP denote the bases selected by invoking the full and pruned RTs at each subspace, respectively. LRB2F and LRB2P denote the bases selected by invoking the full and pruned RTs on the square of the expansion coefficients at each subspace.

supplied to the CART program and the full and pruned RTs were obtained. Unlike the LDB method, we decided not to order the individual basis functions in terms of their importance; see also Remark 6.1 and Section 5.3. The regression errors are shown in Table 6.2. These were also measured by the relative ℓ^2 error. Overall, errors using the LRB coordinates are comparable with those using the LDB coordinates. For the quartz and gas volumes, the best results by the LRB method beat those by the LDB method.

For the quartz volume, the best result so far was obtained by using the pruned tree regression on the LRBP coordinates. As explained in Chapter 5, LRBP denotes the basis selected by the LRB algorithm using the subspace evaluation based on the pruned tree regression errors. This tree is plotted in Figure 6.15. The corresponding basis functions and the selected basis pattern (subspaces) are displayed in Figure 6.16. A close examination of Figures 6.15 and 6.16 suggests that the most important LRB coordinate is #153, i.e., the localized wiggle around the S wave components of the sand waveforms. This basis function works as a detector: if the expansion coefficient of each waveform onto this function is

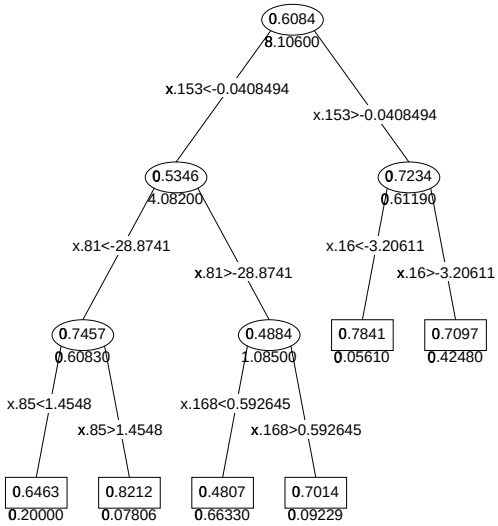


Figure 6.15: The pruned regression tree for the quartz volume using the waveforms represented in the LRBP coordinates.

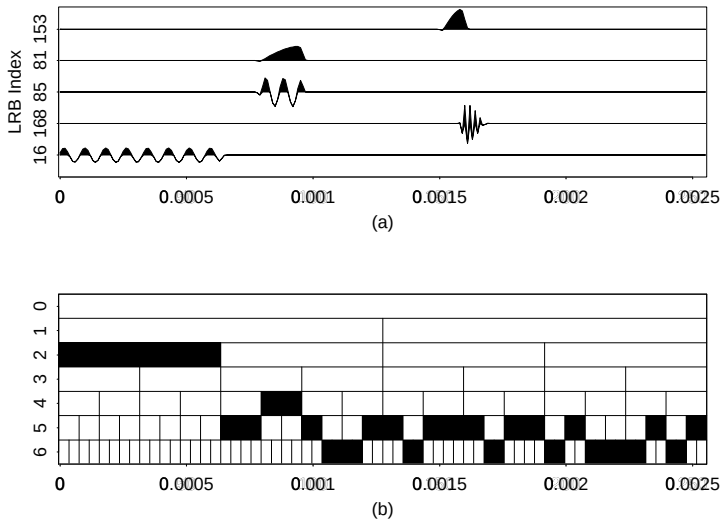


Figure 6.16: (a) The LRB functions used in the pruned regression tree for the quartz volume shown in Figure 6.15. These are displayed in the depth-first search manner in the tree. (b) The selected subspaces as the LRB.

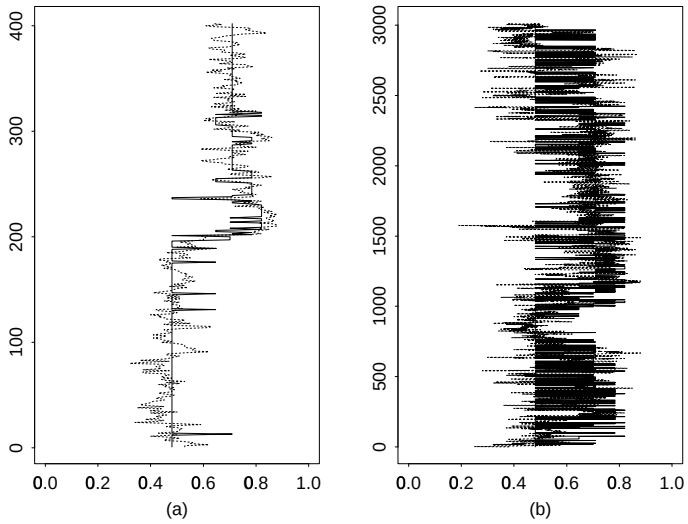


Figure 6.17: The prediction of the quartz volume by the pruned regression tree shown in Figure 6.15. (a) The training dataset. (b) The whole dataset.

below a certain threshold (-0.0408494), that waveform is considered as "sand waveform." Also, we can observe that the combination of the LRB functions #81 #85 (both located around the P wave components) works also as a detector for the sandstone layer containing water: if the coordinate #81 is smaller than -28.8741 and the coordinate #85 is larger than 1.4548, then the highest quartz volume (0.8212) is assigned. When we see the prediction curve of the quartz volume for the training dataset in Figure 6.17, we find that in fact this highest quartz volume is assigned around the indices ranging 207 to 230 which corresponds to the depth indices $1165 \sim 1188$ in the whole dataset, i.e., the water sand region. In Figure 6.4, we notice that the waveforms of the water sand region under discussion have rather different characteristics components than the other water/gas sand regions. The LRB functions #81 and #85 "saved" these waveforms: if we had cut these coordinates, the quartz volume estimate of this region would have been much lower (in fact, 0.5346 as we can see in Figure 6.15). These subtle arguments could not have been done without extracting the local features in the time-frequency plane.

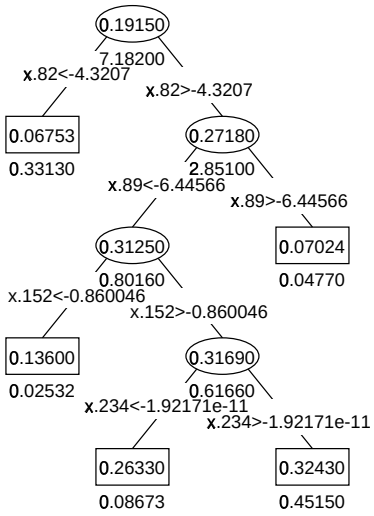


Figure 6.18: The pruned regression tree for the illite volume using the waveforms represented in the LRBF coordinates.

For the illite volume, the pruned tree regression on all the LRBF coordinates gives the best result in this table. The best tree, the LRB functions and the selected subspaces and the prediction results are plotted in Figures 6.18, 6.19, and 6.20, respectively. From these figures, we observe that the selected LRB functions are very similar to the LDB functions shown in Figure 6.10 except that the LRB has a basis function #152 located around the S wave component of the sand waveforms. The sand waveforms have considerable energy in the LRB coordinates $(82, 89, 152)$ whereas the shale waveforms have very small energy in these coordinates. The LRB algorithm decides to use this information to infer the illite volume.

For the gas volume, the pruned regression tree on the LRBP coordinates gives the lowest regression error among all the methods we have tried so far. The best tree, the LRB functions and the selected subspaces and the prediction results are plotted in Figures 6.21, 6.22, and 6.23, respectively. From these figures, we observe that the time axis was first split into half. In the later half (which includes the S wave components), the LRB method

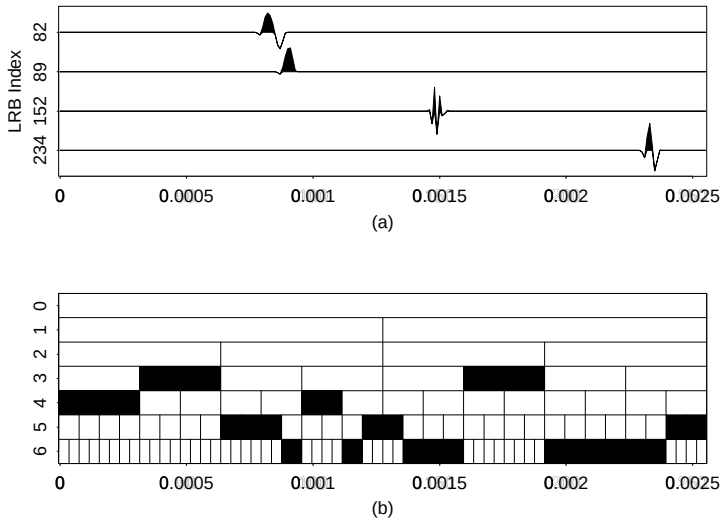


Figure 6.19: (a) The LRB functions used in the pruned regression tree for the illite volume shown in Figure 6.18. These are displayed in the depth-first search manner in the tree. (b) The selected subspaces as the LRB.

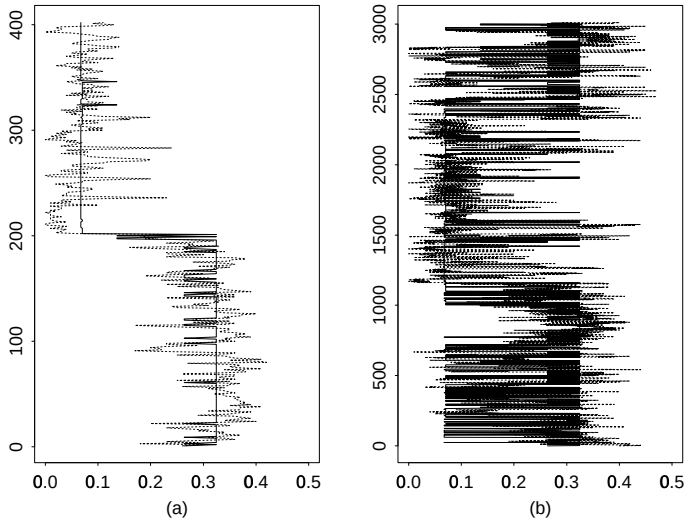


Figure 6.20: The prediction of the illite volume by the pruned regression tree shown in Figure 6.18. (a) The training dataset. (b) The whole dataset.

decided to do the "frequency analysis." The earlier half of the time axis (which includes the P wave components) was further segmented into finer time windows. The LRB functions $(8, 9, 143, 34, 122, 126)$ are responsible for high gas volume, and in particular, $(143, 122)$. The function #143 is a sine wave whose frequency content agrees well with the S wave components of the gas sand waveforms. The function #122 is located between $t \approx 0.001$ and $t = 0.00125$ and has rather high frequency content. From Figure 6.4, we observe that the gas sand waveforms have such waveform features in that interval whereas the water sand waveforms have smaller energy and the shale waveforms have much lower frequency content in that interval.

6.4 Discussion

In the previous section, we showed the results of the regression analysis using the LDB and LRB methods. The selected basis functions (as the useful features for predicting the lithologic attributes) were interpreted more easily in the light of the physics of wave propagation than the previously proposed methods such as [71] and [75]. But there still remain some questions to our approaches.

6.4.1 On the choice of the training dataset

In our study, we have selected the most representative regions of water sand, gas sand, and shale from the whole dataset. In other words, we have used our *a priori* knowledge on the dataset and have selected the training dataset in a nonrandom fashion. If we have such knowledge, we should actively use it for the training. To examine the dependence of the performance of our method on the selection of a training dataset, we conduct the following experiment: the same number of the depth levels (402 levels) is chosen randomly from the whole dataset (3012 depth levels) and is used as a new training dataset. In this case, it is extremely difficult to apply the LDB method since the classes of randomly sampled levels

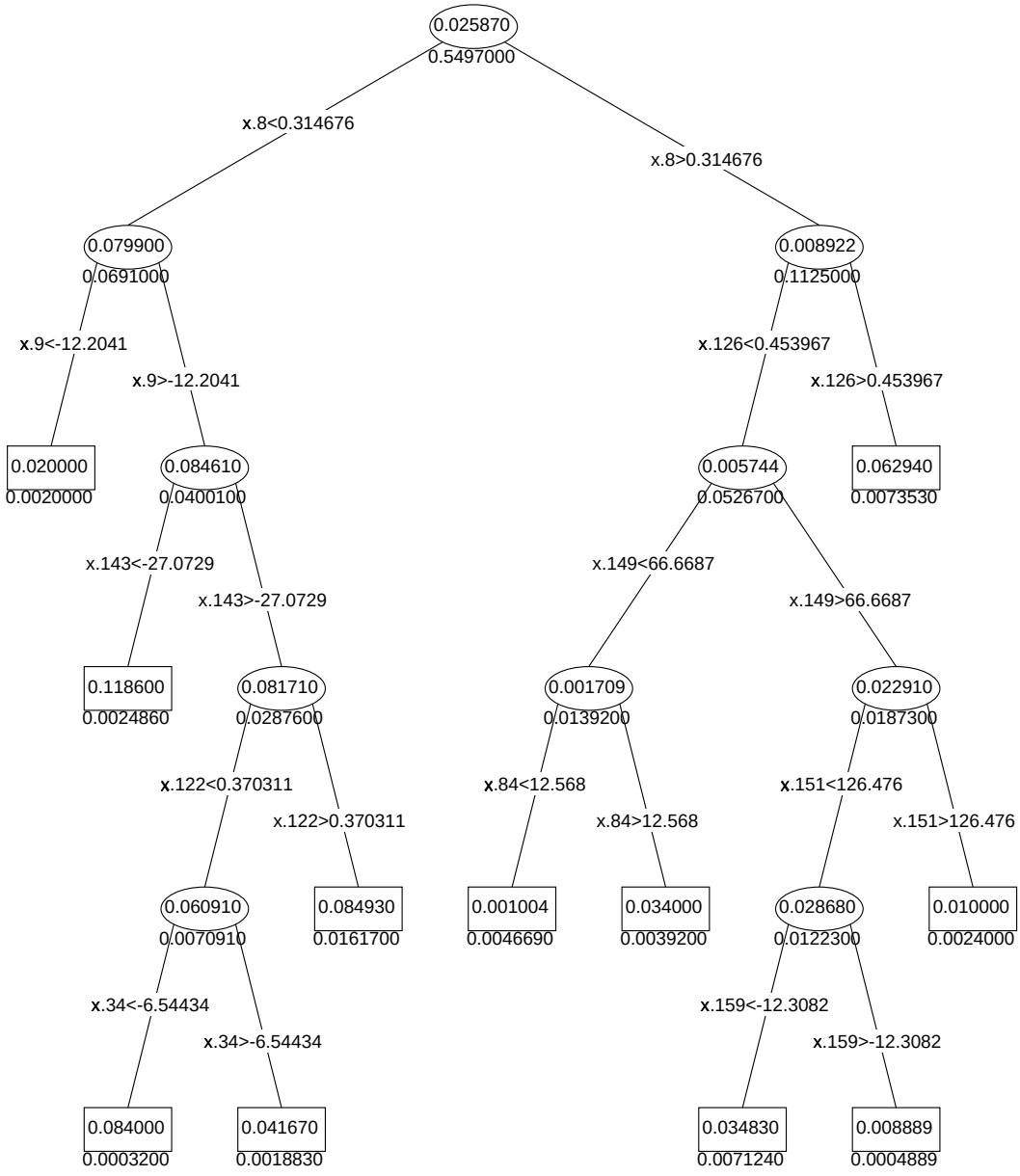


Figure 6.21: The pruned regression tree for the gas volume using the waveforms represented in the LRBP coordinates.

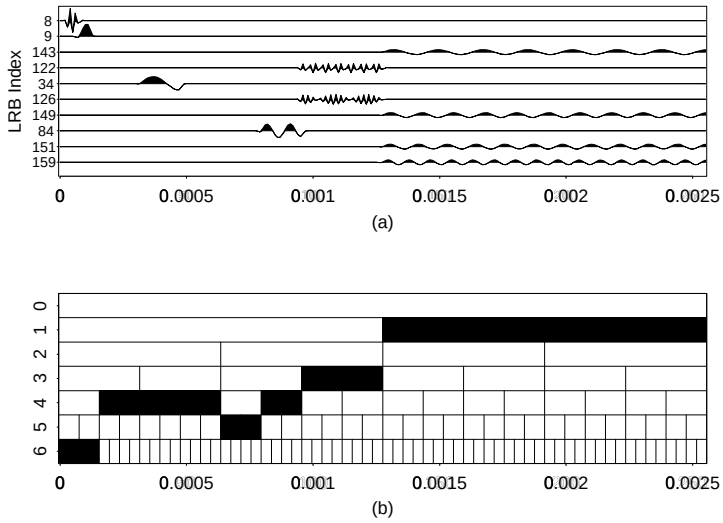


Figure 6.22: (a) The LRB functions used in the full regression tree for the gas volume shown in Figure 6.21. These are displayed in the depth-first search manner in the tree. (b) The selected subspaces as the LRB.

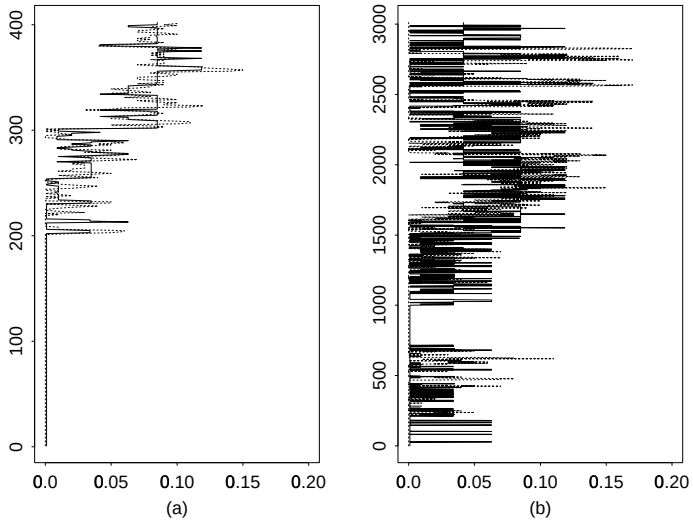


Figure 6.23: The prediction of the gas volume by the full regression tree shown in Figure 6.21. (a) The training dataset. (b) The whole dataset.

Method	Quartz		Illite		Gas	
	Training	Test	Training	Test	Training	Test
FRT on STD	0.08916	0.2203	0.2296	0.5328	0.3426	0.8068
PRT on STD	0.1623	0.1888	0.4146	0.4589	0.5423	0.7329
FRT on LRBF	0.08652	0.2355	0.2247	0.5493	0.2784	0.7769
PRT on LRBF	0.1666	**0.1869**	0.3963	**0.4579**	0.4583	**0.7127**
FRT on LRBP	0.09237	0.2186	0.2402	0.5164	0.2888	0.7947
PRT on LRBP	0.1666	**0.1869**	0.3859	0.4595	0.4131	0.7364
FRT on LRB2F	0.09201	0.2425	0.2247	0.5493	0.3295	0.7894
PRT on LRB2F	0.1785	0.1950	0.3963	**0.4579**	0.4715	0.7372
FRT on LRB2P	0.09041	0.2104	0.2323	0.5357	0.3127	0.7287
PRT on LRB2P	0.1686	0.1909	0.4073	0.4688	0.4916	0.7218

Table 6.3: The prediction errors on the lithologic attributes using the LRB methods applied to the randomly-sampled training dataset. The smallest errors in the test data columns are displayed in bold font.

are not well-defined: the training dataset includes the data from the "shaly-sand" region as mentioned in the beginning of the previous section. Therefore, we conducted the tree regression analysis on the LRB coordinates (and on the standard Euclidean coordinates also). The results are summarized in Table 6.3. For each lithologic attribute, the smallest regression error on the test dataset in the table overcomes the best of all the results using the nonrandom training dataset of the previous section. On the other hand, the resubstitution errors in this table are consistently larger than those using the nonrandom training dataset. These two observations suggest that the nonrandom training dataset, even though we thought we had selected the good representative regions, does not cover the actual response space well (e.g., the data from the "shaly-sand" region). In our future project, we will address more elaborate random sampling schemes such as the bootstrap [58] to increase the prediction accuracy. An interesting exercise would be to construct a regression rule using the whole dataset from this well and then use that rule to predict the lithologic attributes of the neighboring wells using the waveforms recorded at those wells. We also note that the pruned tree regression on the LRBF coordinates consistently gives the best

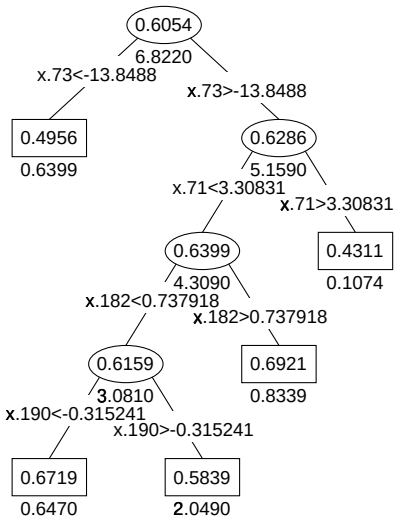


Figure 6.24: The pruned regression tree for the quartz volume using the randomly-sampled training dataset. The tree was initially grown on the waveforms represented in the LRBF coordinates. The same tree was also obtained on the LRBP coordinates.

result for each lithologic attribute although there are some ties.

For the quartz volume, the pruned tree regression on the LRBF coordinates tie the one on the LRBP coordinates. Although the subspace patterns of these two bases are different, the selected basis functions by the RTs are exactly the same ones. Figures 6.24, 6.25, 6.26 show the best tree, the selected LRB functions, and the predicted curves for the quartz volume, respectively. Although the functions located around the P and S wave components are selected again, there are notable differences from those of the nonrandom training dataset shown in Figure 6.16: all the selected LRB functions using the randomly-sampled training dataset have longer support than those using the nonrandomly-sampled training dataset. This implies that the LRB using the randomly-sampled training dataset is more tolerable in the variability in the velocities of the P and S waves. This may be one of the advantages of using the randomly-sampled training dataset.

For the illite volume, we have again a tie as shown in Table 6.3. In both cases, our

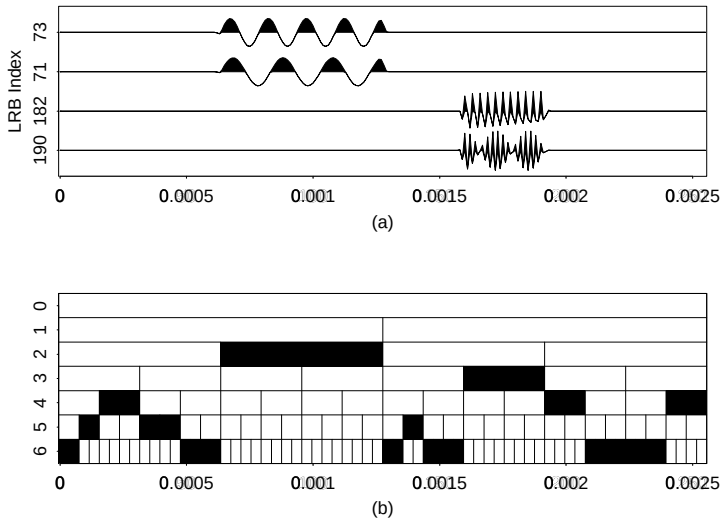


Figure 6.25: (a) The LRB functions used in the pruned regression tree for the quartz volume shown in Figure 6.24. These are displayed in the depth-first search manner in the tree. (b) The selected subspaces as the LRB.

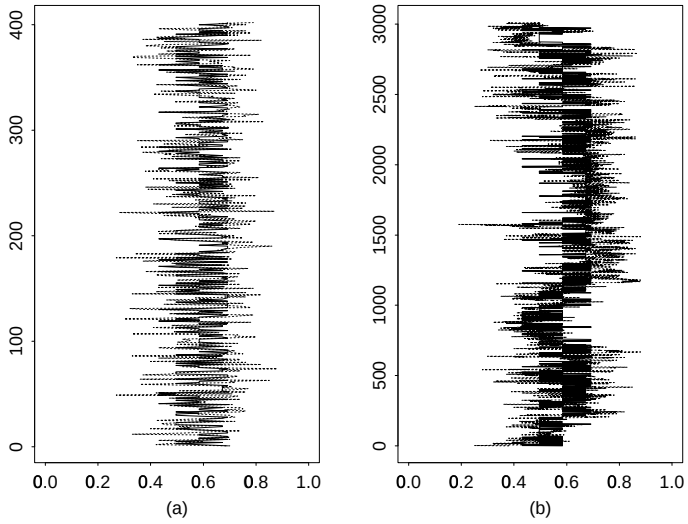


Figure 6.26: The prediction of the quartz volume by the pruned regression tree shown in Figure 6.24. (a) The training dataset. (b) The whole dataset.

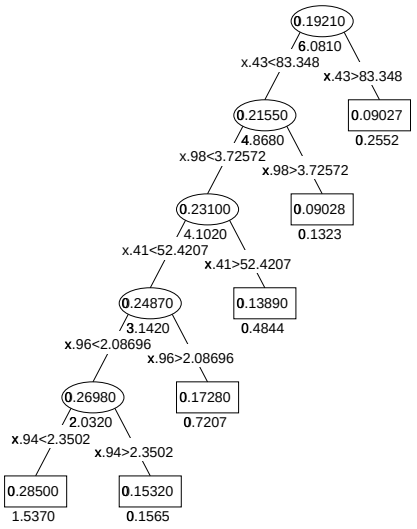


Figure 6.27: The pruned regression tree for the illite volume using the randomly-sampled training dataset. The tree was initially grown on the waveforms represented in the discrete sine basis.

algorithm selected the discrete sine basis as the best LRB and the smallest regression error was obtained by the pruned RT on this coordinate system. The best tree, the selected basis functions, and the predicted curves for the illite volume, are shown in Figures 6.27, 6.28, 6.29, respectively. It is interesting to observe that the highest illite volume (0.285) is assigned for the waveforms whose projections onto each discrete sine basis functions shown in Figure 6.28 is less than a certain positive threshold. In other words, the shale waveforms have either very small energy in these frequencies or negative correlation with these sine basis. The pruned RT on the LRB2P coordinates also gives the same result. This implies that in fact the shale waveforms have rather small energy in these frequencies.

Finally for the gas volume, the smallest regression error was obtained by the pruned tree regression on the LRBF coordinates. Figures 6.30, 6.31, 6.32 show the best tree, the selected LRB functions, and the predicted curves for the gas volume, respectively. From these figures, we observe that the time axis was first split into half. In the earlier half

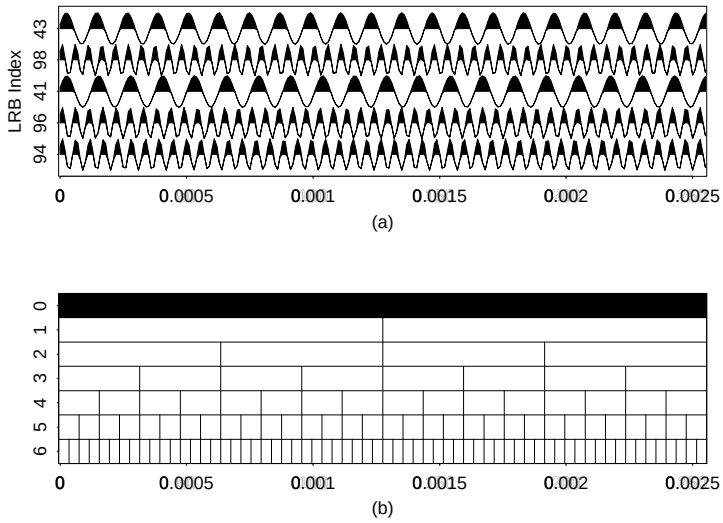


Figure 6.28: The basis functions used in the pruned regression tree for the illite volume shown in Figure 6.27 turn out to be the discrete sine basis. These are displayed in the depth-first search manner in the tree.

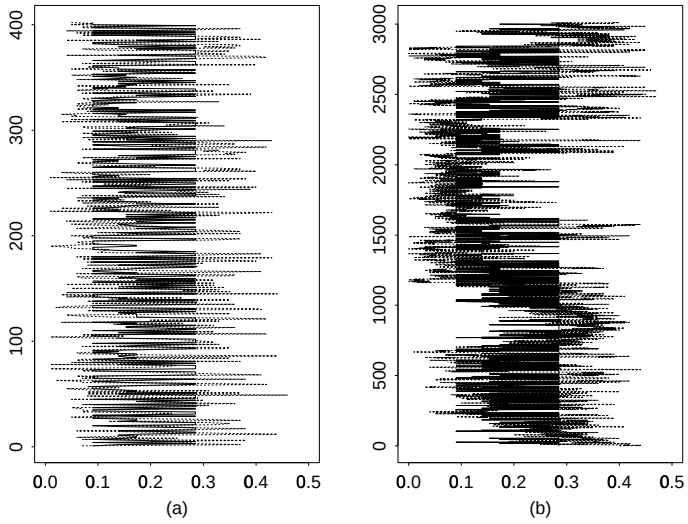


Figure 6.29: The prediction of the illite volume by the pruned regression tree shown in Figure 6.27. (a) The training dataset. (b) The whole dataset.

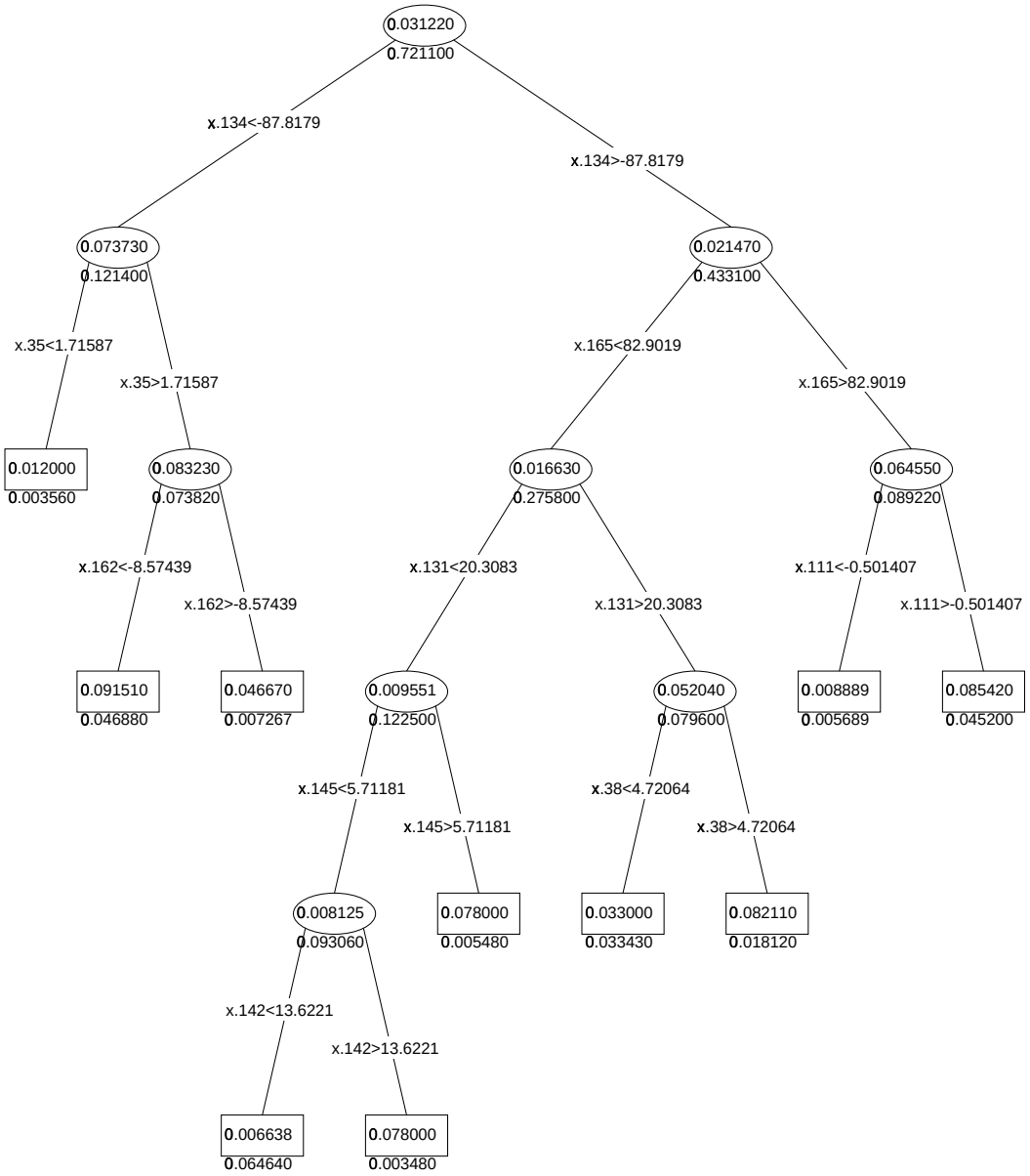


Figure 6.30: The pruned regression tree for the gas volume using the randomly-sampled training dataset. The tree was initially grown on the waveforms represented in the LRBF coordinates.

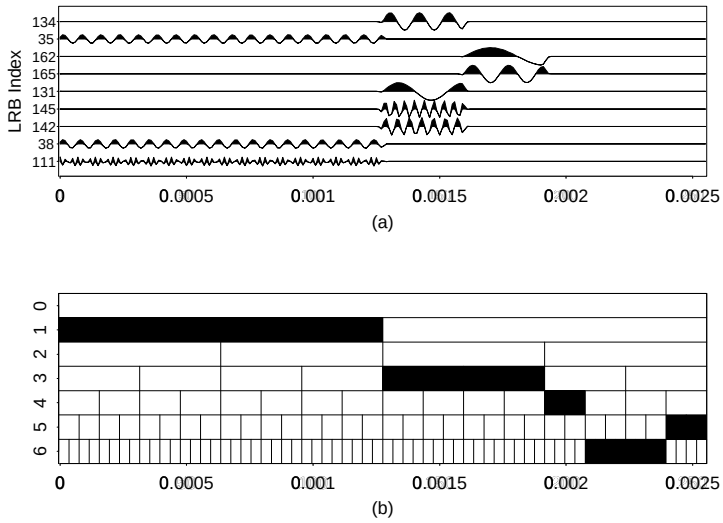


Figure 6.31: (a) The LRB functions used in the pruned regression tree for the gas volume shown in Figure 6.30. These are displayed in the depth-first search manner in the tree. (b) The selected subspaces as the LRB.

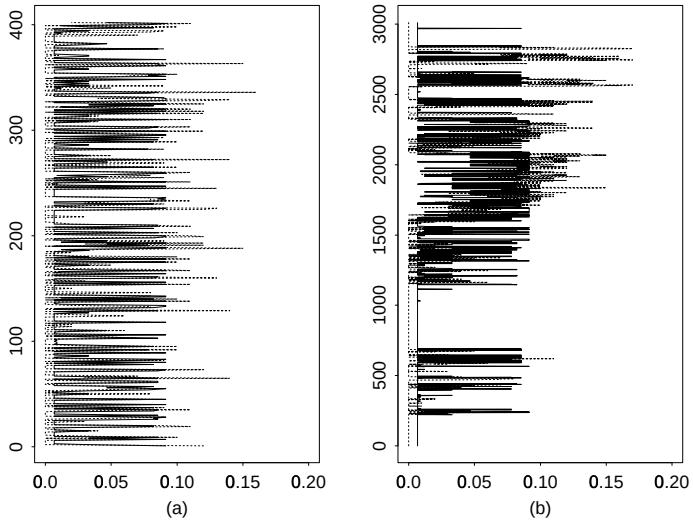


Figure 6.32: The prediction of the gas volume by the pruned regression tree shown in Figure 6.30. (a) The training dataset. (b) The whole dataset.

(which includes the P wave components), the LRB method decided to do the "frequency analysis." The later half of the time axis (which includes the S wave components) was further segmented into finer time windows. This segmentation is completely opposite to the one in the nonrandom training data; see Figure 6.22. Interpreting this tree is rather difficult. Figure 6.32 shows that many depth levels are assigned the constant lowest value (i.e., 0.006638). From the tree in Figure 6.30, we can trace back the LRB functions giving this value; the functions $(134, 165, 131, 145, 142)$ are responsible for this low value, and in particular, $(145, 142)$. The functions $(145, 142)$ work as detectors; if the projected values onto these basis functions are smaller than certain thresholds, the algorithm assigns the low gas volume; otherwise assign rather high gas volume.

6.4.2 Using the physically-derived quantity

In our study so far, we have not really used any physics-based insight to select the features except that we have chosen the good training dataset in Section 6.3 and adopted the LST as the time-frequency decomposition method. Everything else have been automatic. Comparing with the performances of these automatic procedures with those of the regression analysis using the quantities directly derived from the physics has its own interest. The physics of wave propagation suggests that the key parameter for determining the lithology is the ratio of P and S wave velocities V_p/V_s. Although this quantity is still affected by the other factors such as the crack, pore, and geometry, this is related to Poisson's ratio, a characteristic property of elastic solids [109], [143]. In our dataset, the velocities of the P and S wave components at each depth level were computed using the modified version of the Radon transform [81]. This technique was applied to the waveforms recorded at eight receivers[2] above the specific depth level, which were generated by the same acoustic pulse from transmitter below. Based on this available velocity information, we compute the ratios

[2]The tool actually used has eight receivers; in the previous sections we have used only the waveforms recorded at the receiver nearest to the transmitter.

Method	Quartz		Illite		Gas	
	Training	Test	Training	Test	Training	Test
FRT for NTR	0.1065	0.2435	0.2393	0.5854	0.4427	0.7804
PRT for NTR	0.1252	0.2218	0.2786	0.5652	0.5346	0.7338
FRT for RTR	0.1540	0.2086	0.3833	0.5124	0.5267	0.7881
PRT for RTR	0.1880	**0.1821**	0.4667	**0.4522**	0.6425	**0.7147**

Table 6.4: The table of regression errors using the V_p/V_s values of the nonrandomly-sampled training dataset and the randomly-sampled training datasets (which are denoted by NTR and RTR, respectively). The smallest errors in the test data columns are displayed in bold font.

of the P wave speed to the S wave speed at each depth and use these quantities as input signals (now the input signal space is one-dimensional). The regression errors by the full and pruned trees on the nonrandomly-sampled training dataset and the randomly-sampled training dataset are summarized in Table 6.4. For each lithologic attribute, the best result in this table is consistently obtained by the pruned tree regression using the V_p/V_s values of the randomly-sampled training dataset. We also note that the best estimates for the quartz and illite rates in this table are the "best" ones in all the experiments we have done so far. They are even better than the corresponding resubstitution estimates (i.e., the results on the training dataset).

The following figures show the best trees and the best prediction curves of the quartz, the illite, and the gas volumes using the V_p/V_s values.

From Figures 6.33 and 6.35, we observe that the structures and split rules of the pruned RTs for the quartz and the illite volumes are exactly the same. This may be explained again by the "duality" of the quartz and the illite rates in this region. The prediction of the gas volume is slightly worse than the best LRB method.

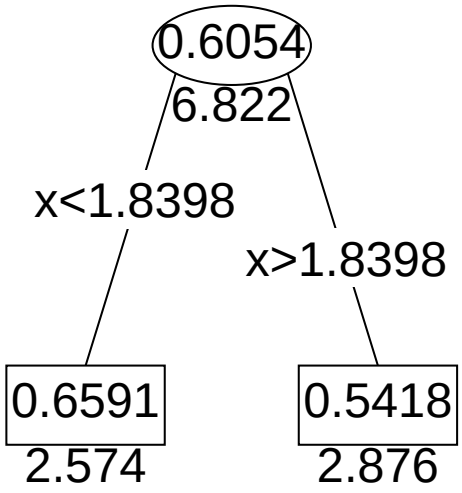


Figure 6.33: The pruned regression tree for the quartz volume using the V_p/V_s values of the randomly-sampled training dataset. This tree has only two terminal nodes: the prediction values have only two possibilities.

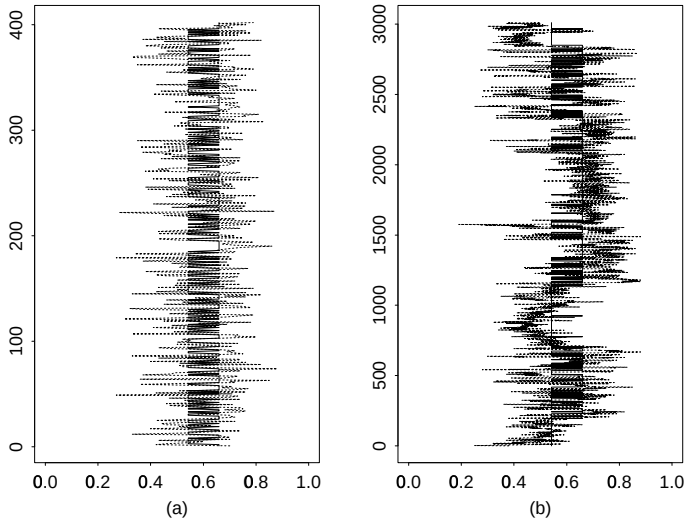


Figure 6.34: The prediction of the quartz volume by the pruned regression tree shown in Figure 6.33. (a) The training dataset. (b) The whole dataset.

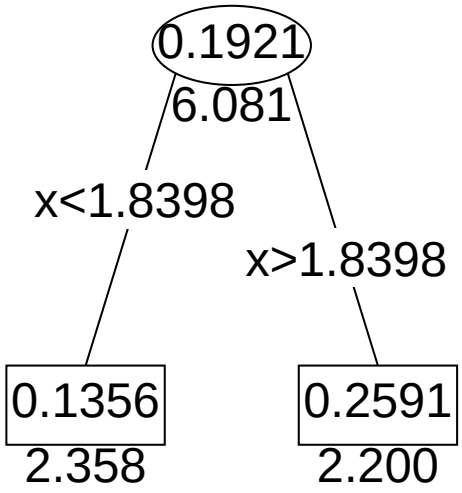


Figure 6.35: The pruned regression tree for the illite volume using the V_p/V_s values of the randomly-sampled training dataset. This tree's structure is exactly the same as the one shown in Figure 6.35.

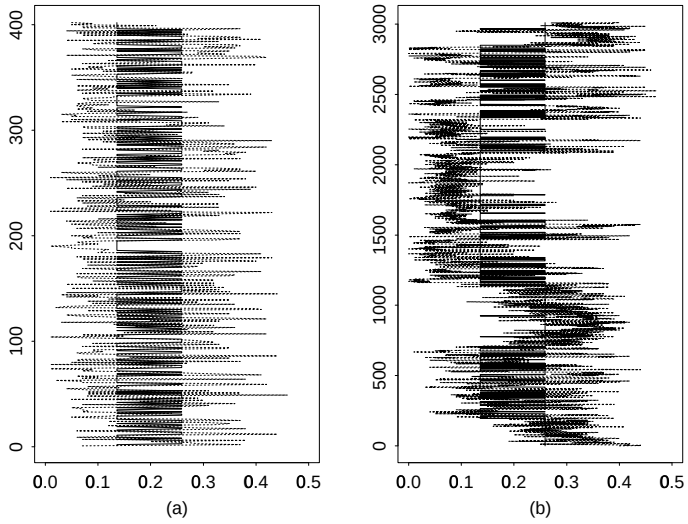


Figure 6.36: The prediction of the illite volume by the pruned regression tree shown in Figure 6.35. (a) The training dataset. (b) The whole dataset.

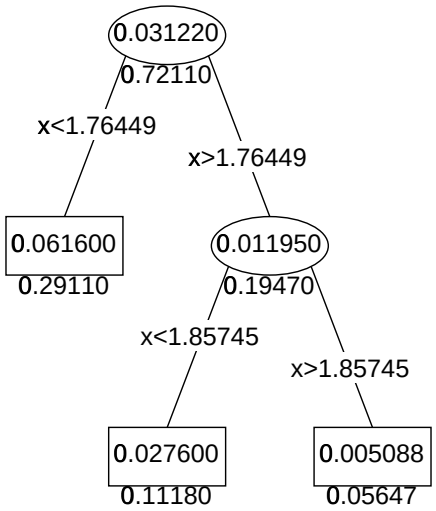


Figure 6.37: The pruned regression tree for the gas volume using the V_p/V_s values of the randomly-sampled training dataset.

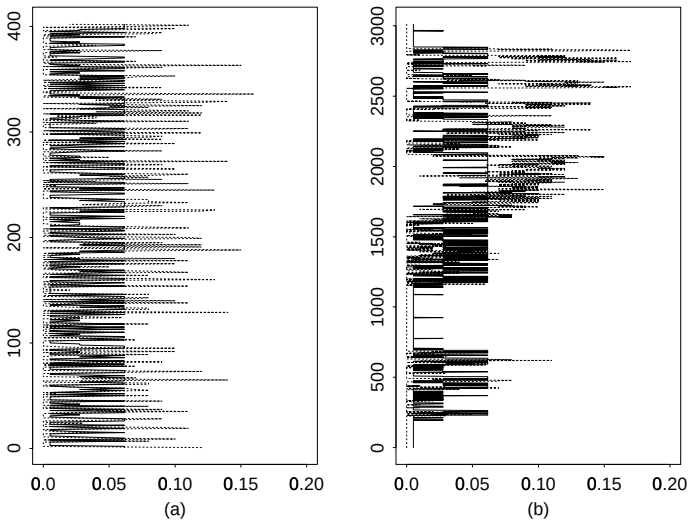


Figure 6.38: The prediction of the gas volume by the pruned regression tree shown in Figure 6.37. (a) The training dataset. (b) The whole dataset.

6.4.3 On the measure of regression errors

Is the ℓ^2 norm an appropriate measure of regression error? This measure has been used in all the splitting rules in the RTs as well as in the tables of the prediction errors in this chapter. Since the function we want to approximate/predict is not smooth at all (they are discretized versions of the functions in the space of functions of bounded variation), the error measure based on the ℓ^1 norm would be more appropriate. It is true that the quartz and illite volume predictions derived from the V_p/V_s values shown in Figures 6.34 and 6.36 give the least ℓ^2 error among all the experiments we have done; however, they have only two possible values both of which are close to the total mean values. At least, visually, these are not satisfactory compared to the predictions by the LRB methods shown in, e.g., Figures 6.26 and 6.29. This problem is often encountered in the image processing field such as image compression and reconstruction: the small ℓ^2 error does not necessary imply the satisfactory image estimate; see [49] for the details on the error criterion for image compression. Unfortunately, the software we rely on, i.e., S-PLUS [140], does not allow us to change the measure of regression error; the ℓ^2 norm-based error is hard-wired in the current version of S-PLUS. Breiman et al. describe the RT based on the ℓ^1 norm (the so-called "Least Absolute Deviation" [LAD] regression) in [18, Section 8.11]. We hope that the RT based on the LAD regression error will be available soon in the S-PLUS.

Besides what we mentioned earlier, our future projects also include to: 1) examine the importance of the frequency contents/shape information, the amplitude information, and the velocity information, *independently*, on predicting the lithologic information, 2) incorporate the user interactions to pick the basis functions flexibly for the regression analysis, and 3) examine the capability of other nonlinear regression methods such as neural networks.

6.5 Summary

In this chapter, we have applied the LDB and LRB methods developed in Chapters 4 and 5 to a geophysical problem of predicting the lithologic information from the acoustic well-logging waveforms. Using these methods, we could successfully extract the useful features for predicting this information. The results, in general, agree with the explanations from the physics of wave propagation, although our use of the physics in constructing the regression rules is minimal. The best results using our methods are found to be comparable to the predictions using the physically-derived quantities such as the V_p/V_s values.

Chapter 7

Multiresolution Representations Using the Autocorrelation Functions of Wavelets and Their Applications

7.1 Introduction

So far we have concentrated on the library of orthonormal bases and their applications. The orthogonality plays an important role over there mainly for the computational efficiency and the simplicity in implementation of the numerical algorithms. In this chapter, we consider a library of "non-orthogonal" bases, namely, the autocorrelation functions of wavelets. We trade the orthogonality with the convenient properties for explicitly characterizing edges or singularities of signals. It is certainly possible to estimate the local behavior of signals by analyzing the growth or decay from scale to scale of the coefficients of the orthonormal wavelet expansions; however, the coefficients of the orthonormal wavelet expansions are not

shift-invariant. Thus, redundant representations (without subsampling at each scale, e.g., [19], [139], [123], [131], or the continuous wavelet transforms [85]) are being used in order to simplify the analysis of coefficients from scale to scale. In particular, the orthonormal wavelet expansion of a vector of length N without subsampling is not only shift-invariant but also contains all the wavelet coefficients to represent N circularly-shifted versions of the original signal [139], [12], [131].

The asymmetric shape of the orthonormal compactly supported wavelets presents another difficulty for the analysis of signals. The symmetric basis functions are preferred since, for example, their use simplifies finding zero-crossings (or extrema) corresponding to the locations of edges in images at later stages of processing. There are several approaches for dealing with this problem. The first approach consists in constructing approximately symmetric orthonormal wavelets and gives rise to approximate quadrature mirror filters [94]. The second consists in using biorthogonal bases [27], [147], so that the basis functions may be chosen to be exactly symmetric.

Alternatively, a redundant (shift-invariant) representation using dilations and translations of the *autocorrelation functions* of wavelets may be used for signal analysis instead of the wavelets per se. The exact filters for the decomposition process are the autocorrelations of the quadrature mirror filter coefficients of the compactly supported wavelets and, therefore, are exactly symmetric. The recursive definition of the autocorrelation functions of wavelets leads to fast iterative algorithms to generate a shift-invariant multiresolution representation which we call the *autocorrelation shell representation.* A remarkable feature of this representation is a natural interpolation algorithm associated with it. This interpolation algorithm, the so-called *symmetric iterative interpolation*, is due to Dubuc [55] and Deslauriers and Dubuc [47]. The coefficients of the interpolation scheme of [55] and [47] generated from the Lagrange polynomials are the autocorrelation coefficients of the quadrature mirror filters associated with the compactly supported wavelets of [42]. This connection

was also noticed by Shensa in [139]. This interpolation scheme of Dubuc is also related to the "algorithme à trous" in [72], [56]. Another interesting feature of this representation is its convertibility to the redundant expansion (without subsampling) by the corresponding orthonormal wavelets on each scale, independently of other scales.

As an application of the proposed representation, we consider the reconstruction of a signal from its multiscale edge representation. Here, the multiscale edge representation means the pairs of: 1) locations of zero-crossings which indicate positions of edges, and 2) slopes at these zero-crossings, in the multiple scale representation of the signal, in particular, in the autocorrelation shell representation. This problem has intrigued many researchers in human and machine vision community, mainly due to David Marr's conjecture [101]: "The edge information of a signal in multiple scales is sufficient to recover the original signal itself." The autocorrelation functions of wavelets give two advantages toward this problem: 1) the symmetric iterative interpolation scheme allows us to detect zero-crossings and to compute the slopes at these points in a simple and efficient manner, and 2) the reconstruction of the original signal from its zero-crossing information is posed as solving a system of linear algebraic equations so that the relationship between the zero-crossings representation and the original signal becomes quite explicit. Such a representation should be useful for nonlinear manipulation of signals, for example, edge-preserving smoothing and interpolation; see Mallat and Zhong [98], Mallat and Hwang [96]. For other nonlinear edge-preserving smoothing algorithms, see e.g., [115], [111].

Our results can also be viewed as a way to obtain the continuous-like multiresolution analysis starting from the discrete multiresolution analysis. Another approach to make the connection between continuous and discrete multiresolution analyses is developed by Duval-Destin et al. [57], where the starting point is the continuous version of the multiresolution analysis. The interested reader in this connection is referred to the recent work of Beylkin and Torrésani [14]; see also Donoho's "interpolating" wavelet transforms [50] which compute

the (non-orthogonal) wavelet coefficients of continuous functions not by integration but by sampling based on the Deslauriers-Dubuc interpolation scheme.

The organization of this chapter is as follows. In Section 7.2 we introduce the notion of the *orthonormal shell* expansion of signals which generates a shift-invariant representation using the orthonormal wavelets. In Section 7.3 we consider expansion of signals into the autocorrelation shell. A new interpretation of Dubuc's iterative interpolation scheme is discussed in Section 7.4. In Section 7.5 we formulate our approach to the reconstruction of a signal from its multiscale zero-crossings representation and give examples.

7.2 Orthonormal Shell: A Shift-Invariant Representation Using Orthonormal Wavelets

In certain pattern recognition applications such as pattern matching, shift invariance of a signal representation is of critical importance. Although coefficients of orthonormal wavelet expansions are not shift invariant, the wavelet coefficients of all N circulant shifts of a vector of size $N = 2^n$ may be computed in $O(N \log N)$ operations [12]. In this section, we propose another way to compute such set of coefficients. Once all wavelet coefficients of N circulant shifts of the vector are computed, we may use them for a variety of applications where the shift invariance is essential.

But, first, let us briefly review the properties of the compactly supported wavelets (for details we refer to [42], [43]). The orthonormal basis of compactly supported wavelets of $L^2(\mathbb{R})$ is formed by the dilation and translation of a single function $\psi(x)$,

$$\psi_{j,k}(x) = 2^{-j/2}\psi(2^{-j}x - k), \tag{7.1}$$

where $j, k \in \mathbb{Z}$. The function $\psi(x)$ has a companion, the scaling function $\varphi(x)$.

The wavelet basis induces a multiresolution analysis on $L^2(\mathbb{R})$ [104], [93], i.e., the

decomposition of the Hilbert space $L^2(\mathbb{R})$ into a chain of closed subspaces

$$\cdots \subset \mathcal{V}_2 \subset \mathcal{V}_1 \subset \mathcal{V}_0 \subset \mathcal{V}_{-1} \subset \mathcal{V}_{-2} \subset \cdots \tag{7.2}$$

such that

$$\bigcap_{j\in\mathbb{Z}} \mathcal{V}_j = \{0\}, \quad \overline{\bigcup_{j\in\mathbb{Z}} \mathcal{V}_j} = L^2(\mathbb{R}). \tag{7.3}$$

By defining $\mathcal{W}_j$ as an orthogonal complement of $\mathcal{V}_j$ in $\mathcal{V}_{j-1}$,

$$\mathcal{V}_{j-1} = \mathcal{V}_j \oplus \mathcal{W}_j, \tag{7.4}$$

the space $L^2(\mathbb{R})$ is represented as a direct sum

$$L^2(\mathbb{R}) = \bigoplus_{j\in\mathbb{Z}} \mathcal{W}_j. \tag{7.5}$$

On each fixed scale j, the wavelets $\{\psi_{j,k}(x)\}_{k\in\mathbb{Z}}$ form an orthonormal basis of $\mathcal{W}_j$ and the functions $\{\varphi_{j,k}(x) = 2^{-j/2}\varphi(2^{-j}x - k)\}_{k\in\mathbb{Z}}$ form an orthonormal basis of $\mathcal{V}_j$. In addition, the function ψ has M vanishing moments

$$\int_{-\infty}^{+\infty} \psi(x)x^m\,dx = 0, \qquad m = 0, \ldots, M-1. \tag{7.6}$$

Due to the fact that $\mathcal{V}_j, \mathcal{W}_j \subset \mathcal{V}_{j-1}$, the functions φ and ψ satisfy the following relations:

$$\varphi(x) = \sqrt{2}\sum_{k=0}^{L-1} h_k\varphi(2x-k), \tag{7.7}$$

$$\psi(x) = \sqrt{2}\sum_{k=0}^{L-1} g_k\varphi(2x-k), \tag{7.8}$$

where

$$g_k = (-1)^k h_{L-k-1}, \qquad k = 0, \ldots, L-1. \tag{7.9}$$

The number of coefficients L in (7.7) and (7.8) is related to the number of vanishing moments M and for the wavelets in [42] $L = 2M$. If additional conditions (such as less asymmetry and regularity) are imposed, then the relation might be different [43]; e.g., the coiflet, which is a less asymmetric version of the Daubechies wavelet, has $L = 3M$. But L is always even.

The coefficients $H = \{h_k\}_{0 \le k \le L-1}$ and $G = \{g_k\}_{0 \le k \le L-1}$ in (7.7) and (7.8) are quadrature mirror filters since they satisfy the equation

$$|m_0(\xi)|^2 + |m_1(\xi)|^2 = 1, \tag{7.10}$$

where the 2π-periodic functions m_0 and m_1 are defined as

$$m_0(\xi) = \frac{1}{\sqrt{2}} \sum_{k=0}^{L-1} h_k \mathrm{e}^{\mathrm{i}k\xi}, \tag{7.11}$$

$$m_1(\xi) = \frac{1}{\sqrt{2}} \sum_{k=0}^{L-1} g_k \mathrm{e}^{\mathrm{i}k\xi} = \mathrm{e}^{\mathrm{i}(\xi+\pi)} \overline{m_0(\xi+\pi)}. \tag{7.12}$$

7.2.1 The orthonormal shell

In practical applications there is always the finest scale of interest and, therefore, it is sufficient to consider only shifts by multiples of some fixed unit. Throughout this chapter we will assume that the number of scales is finite and that there exist a finest and a coarsest scale of interest. Without loss of generality, we will assume that the finest scale is described by the $N(= 2^n)$ dimensional subspace $\mathcal{V}_0 \subset L^2(\mathbb{R})$ and consider only circulant shifts on $\mathcal{V}_0$. In this case, the multiresolution decomposition of the space $\mathcal{V}_0$ may be written as

$$\mathcal{V}_0 = \left(\bigoplus_{j=1}^{J} \mathcal{W}_j \right) \oplus \mathcal{V}_J, \tag{7.13}$$

where $1 \le J \le n$ and the subspace $\mathcal{V}_J$ describes the coarsest scale.

Since the functions

$$\{\psi_{j,k}(x)\}_{1 \le j \le J,\ 0 \le k \le 2^{n-j}-1} \quad \text{and} \quad \{\varphi_{J,k}(x)\}_{0 \le k \le 2^{n-J}-1}, \tag{7.14}$$

form an orthonormal basis of $\mathcal{V}_0$, for any vector $f \in \mathcal{V}_0$ represented by

$$f(x) = \sum_{k=0}^{N-1} s_k^0 \varphi_{0,k}(x), \tag{7.15}$$

we have the following relation:

$$\|f\|^2 = \sum_{j=1}^{J} \sum_{k=0}^{2^{n-j}-1} (d_k^j)^2 + \sum_{k=0}^{2^{n-J}-1} (s_k^J)^2. \tag{7.16}$$

The coefficients s_k^j and d_k^j in (7.16) are defined as

$$s_k^j = \int f(x)\varphi_{j,k}(x)\,dx, \tag{7.17}$$

$$d_k^j = \int f(x)\psi_{j,k}(x)\,dx, \tag{7.18}$$

for $j = 1, 2, \ldots, J$ and $k = 0, 1, \ldots, 2^{n-j} - 1$, and the norm is defined as

$$\|f\| = \left(\sum_{k=0}^{N-1} (s_k^0)^2 \right)^{1/2}. \tag{7.19}$$

We refer to the set of coefficients $\{d_k^j\}_{1 \le j \le J,\ 0 \le k \le 2^{n-j}-1}$ and $\{s_k^J\}_{0 \le k \le 2^{n-J}-1}$ as orthonormal wavelet coefficients.

We now consider a family of functions

$$\left\{ \widetilde{\psi}_{j,k}(x) \right\}_{1 \le j \le J,\ 0 \le k \le N-1} \quad \text{and} \quad \{\widetilde{\varphi}_{J,k}(x)\}_{0 \le k \le N-1}, \tag{7.20}$$

where

$$\widetilde{\psi}_{j,k}(x) = 2^{-j/2}\psi(2^{-j}(x-k)), \tag{7.21}$$

$$\widetilde{\varphi}_{J,k}(x) = 2^{-J/2}\varphi(2^{-J}(x-k)). \tag{7.22}$$

We call this family a *shell* of the orthonormal wavelets for shifts in $\mathcal{V}_0$. Henceforth, we call this family an *orthonormal shell* for short.

Let us define the following norm,

$$\|f\|_{\mathcal{S}}^2 = \sum_{j=1}^{J} 2^{-j} \sum_{k=0}^{N-1} (d_k^j)^2 + 2^{-J} \sum_{k=0}^{N-1} (s_k^J)^2, \tag{7.23}$$

where the coefficients s_k^j and d_k^j are defined as

$$s_k^j = \int f(x) \widetilde{\varphi}_{j,k}(x)\, dx, \tag{7.24}$$

$$d_k^j = \int f(x) \widetilde{\psi}_{j,k}(x)\, dx, \tag{7.25}$$

We refer to the set of coefficients $\{d_k^j\}_{1 \le j \le J,\ 0 \le k \le N-1}$ and $\{s_k^J\}_{0 \le k \le N-1}$ as the *orthonormal shell coefficients*. The factor 2^{-j} in (7.23) is used to offset the redundancy of this representation. In other words, at the jth scale this representation is 2^j times more redundant than the orthonormal wavelet representation. It is clear that

$$\|f\|^2 = \|f\|_{\mathcal{S}}^2. \tag{7.26}$$

7.2.2 A fast algorithm for expanding into the orthonormal shell

Let us assume that the orthonormal wavelet coefficients of the finest scale $\{s_k^0\}_{0 \le k \le N-1}$ are given as an original signal and let us consider the function $f = \sum_{k=0}^{N-1} s_k^0 \varphi_{0,k} \in \mathcal{V}_0$. To obtain the orthonormal shell coefficients of this function f, we use the quadrature mirror filters $H = \{h_l\}_{0 \le l \le L-1}$ and $G = \{g_l\}_{0 \le l \le L-1}$ (associated with the orthonormal basis of compactly supported wavelets) and compute

$$s_k^j = \sum_{l=0}^{L-1} h_l s_{k+2^{j-1}l}^{j-1}, \tag{7.27}$$

$$d_k^j = \sum_{l=0}^{L-1} g_l s_{k+2^{j-1}l}^{j-1}, \tag{7.28}$$

for $j = 1, \ldots, J,\ k = 0, \ldots, N-1$. Clearly, computations via (7.27) and (7.28) require $2NJ \le 2N \log_2 N$ operations. The diagram for computing these coefficients via (7.27)

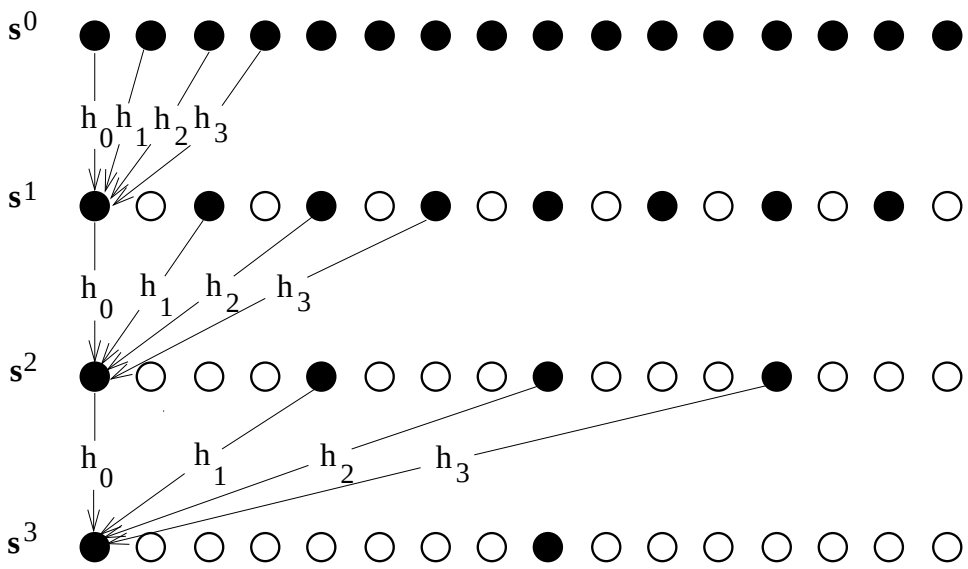


Figure 7.1: A scheme illustrating the algorithm for expanding into the orthonormal shell. Using the quadrature mirror filter $H = \{h_0, h_1, h_2, h_3\}$, all points are computed for the orthonormal shell, whereas only points marked by $\bullet$ are computed for the orthonormal wavelet expansion.

and (7.28) is illustrated in Figure 7.1. We note that the computational diagram of this algorithm is essentially identical to the Hierarchical Discrete Correlation scheme (HDC) [19] of P. Burt, which was designed for efficient correlation of images at multiple scales. We also note that the HDC scheme was proposed prior to the Laplacian pyramid scheme [21] which, in turn, stimulated the development of the multiresolution analysis [93] and of the orthonormal bases of the compactly supported wavelets [42].

Using (7.11) and (7.12), we rewrite (7.27) and (7.28) in the Fourier domain,

$$\hat{s}^j(\xi) = \sqrt{2}\ \overline{m_0(2^{j-1}\xi)}\ \hat{s}^{j-1}(\xi), \tag{7.29}$$

$$\hat{d}^j(\xi) = \sqrt{2}\ \overline{m_1(2^{j-1}\xi)}\ \hat{s}^{j-1}(\xi). \tag{7.30}$$

Let us show that we have computed the orthonormal wavelet coefficients of all circulant shifts of the function f. Since the algorithmic structures for computing the coefficients

$\{s_k^j\}$ and $\{d_k^j\}$ are exactly the same, we consider only $\{d_k^j\}$. At the first scale, we have

$$d_k^1 = \sum_{l=0}^{L-1} g_l s_{k+l}^0. \tag{7.31}$$

We rewrite (7.31) as

$$d_{2k}^1 = \sum_{l=0}^{L-1} g_l s_{2k+l}^0, \tag{7.32}$$

$$d_{2k+1}^1 = \sum_{l=0}^{L-1} g_l s_{2k+1+l}^0, \tag{7.33}$$

for $k = 0, \ldots, N/2 - 1$. The right hand side of (7.32) coincides with the computation of the orthonormal wavelet coefficients orthonormal wavelet coefficients of the *shifted* signal $f(x+1)$. It is clear that the sequence $\{d_{2k}^1\}$ contains all the orthonormal wavelet coefficients that appear if $f(x)$ is circularly shifted by $2, 4, \ldots$, and the sequence $\{d_{2k+1}^1\}$ contains all the orthonormal wavelet coefficients for odd shifts $(1, 3, \ldots)$.

Similarly, at the jth scale, we may rewrite d_k^j as

$$d_{2^j k}^j = \sum_{l=0}^{L-1} g_l s_{2^{j-1}(2k+l)}^{j-1}, \tag{7.34}$$

$$d_{2^j k+1}^j = \sum_{l=0}^{L-1} g_l s_{2^{j-1}(2k+l)+1}^{j-1}, \tag{7.35}$$

$$\vdots$$

$$d_{2^j k+2^j-1}^j = \sum_{l=0}^{L-1} g_l s_{2^{j-1}(2k+l)+2^j-1}^{j-1}, \tag{7.36}$$

for $k = 0, \ldots, 2^{n-j}$. Now the sequences $\{d_{2^j k}^j\}$, $\{d_{2^j k+1}^j\}$, $\ldots$, $\{d_{2^j k+2^j-1}^j\}$ contain the orthonormal wavelet coefficients of the jth scale of the signal shifted by $0, 1, \ldots, 2^j - 1$, respectively. Therefore, the set $\{d_k^j\}_{1 \le j \le J,\ 0 \le k \le N-1}$ and $\{s_k^J\}_{0 \le k \le N-1}$ contains all the coefficients of the orthonormal wavelet expansion of $f(x), f(x+1), \ldots, f(x+N-1)$.

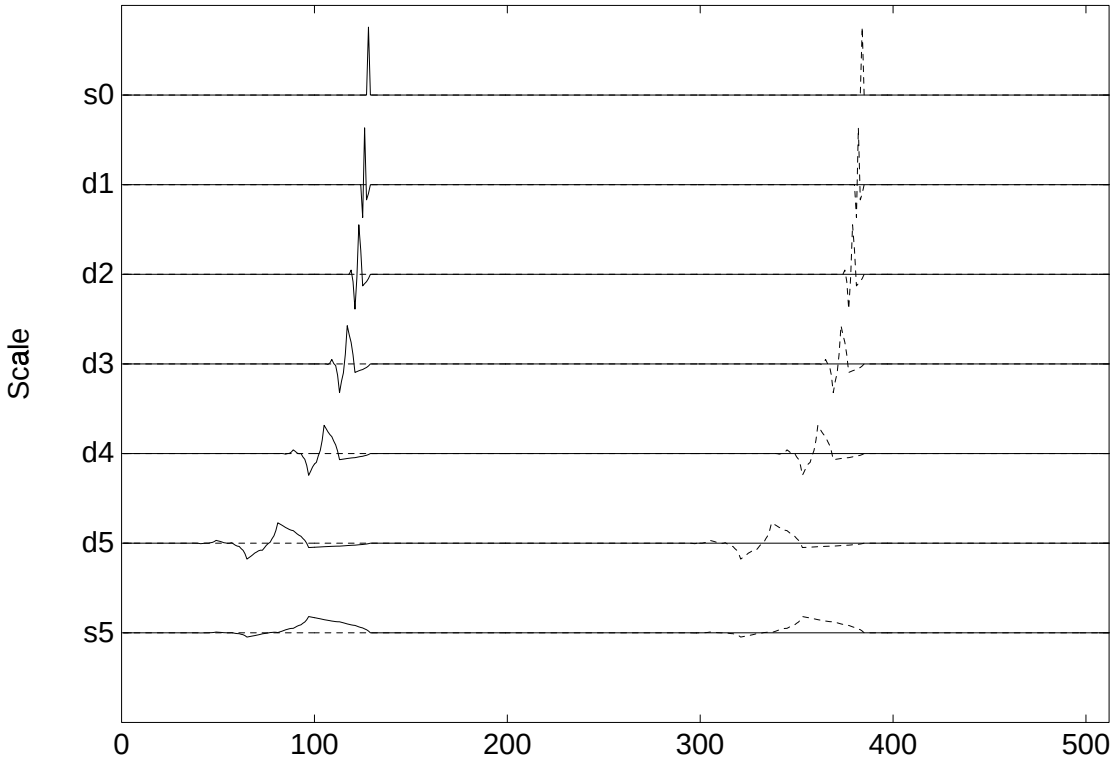


Figure 7.2: The expansion of two unit impulses into the orthonormal shell using the Daubechies wavelet with two vanishing moments and $L = 4$.

Figure 7.2 illustrates the shift invariance of the representation. In Figure 7.2 we use the quadrature mirror filters with two vanishing moments and of length $L = 4$. We also set depth of expansions $J = 5$. The top row shows the original impulses. Dotted lines highlight the expansion of the shifted impulse. Clearly, the shift in the original signal is preserved at each scale. Note that in Figure 7.2 we see the mirror images of wavelets with details appropriate at the corresponding scales. It is clear that due to "rough" shapes of wavelets there might be "too many" zero-crossings. Also, the positions of peaks are shifted across the scales due to the asymmetry of the Daubechies wavelets.

The following proposition shows the relationship between the original signal $\{s_k^0\}$ and the coefficients of the orthonormal shell expansion of this signal.

Proposition 7.1. *For any function* $f \in \mathcal{V}_0$, $f(x) = \sum_{k=0}^{N-1} s_k^0 \varphi(x-k)$, *the coefficients* $\{s_k^j\}$ *and* $\{d_k^j\}$ *defined in* (7.24) *and* (7.25) *satisfy the following identities*

$$\sum_{k=0}^{N-1} s_k^j \widetilde{\varphi}_{0,k}^{\triangleleft} = \sum_{k=0}^{N-1} s_k^0 \widetilde{\varphi}_{j,k}^{\triangleleft}, \tag{7.37}$$

$$\sum_{k=0}^{N-1} d_k^j \widetilde{\varphi}_{0,k}^{\triangleleft} = \sum_{k=0}^{N-1} s_k^0 \widetilde{\psi}_{j,k}^{\triangleleft}, \tag{7.38}$$

where $\widetilde{\varphi}_{0,k}(x) = \varphi_{0,k}(x)$, $\widetilde{\varphi}^{\triangleleft}(x) = \widetilde{\varphi}(-x)$, *and* $\widetilde{\varphi}_{j,k}$, $\widetilde{\psi}_{j,k}$ *are defined in* (7.22) *and* (7.21).

Proof. Recursively applying (7.29) and (7.30), the relationship between $\hat{s}^0(\xi)$ and $\hat{s}^j(\xi)$, $\hat{d}^j(\xi)$ may be written as

$$\hat{s}^j(\xi) = \hat{s}^0(\xi)\ 2^{j/2} \prod_{l=1}^{j} \overline{m_0(2^{l-1}\xi)}, \tag{7.39}$$

$$\hat{d}^j(\xi) = \hat{s}^0(\xi)\ 2^{j/2}\ \overline{m_1(2^{j-1}\xi)} \prod_{l=1}^{j-1} \overline{m_0(2^{l-1}\xi)}, \tag{7.40}$$

for $k = 0, \ldots, N-1$. By multiplying (7.39) by $\overline{\hat{\varphi}(\xi)}$, we have

$$\hat{s}^j(\xi)\overline{\hat{\varphi}(\xi)} = \hat{s}^0(\xi)\ 2^{j/2} \prod_{l=1}^{j} \overline{m_0(2^{l-1}\xi)}\ \overline{\hat{\varphi}(\xi)} = \hat{s}^0(\xi)\ 2^{j/2}\overline{\hat{\varphi}(2^j\xi)}, \tag{7.41}$$

where we have used the identity

$$\hat{\varphi}(\xi) = \prod_{l=1}^{\infty} m_0(2^{-l}\xi). \tag{7.42}$$

The inverse Fourier transform of (7.41) yields (7.37). The relation (7.38) may be derived similarly. □

Figure 7.2 illustrates the proposition. By applying the proposition to the sequence $\{s_k^0 = \delta_{k_0,k}\}$, we have an expansion $\{2^{-j/2}\varphi(2^{-j}(k_0 - x))\}_{1\le j\le J}$ and $2^{-J/2}\varphi(2^{-J}(k_0 - x))$. Therefore, we see the mirror images of the basis functions themselves.

7.2.3 A fast reconstruction algorithm

Given the coefficients $\{d_k^j\}_{1\le j\le J,\ 0\le k\le N-1}$ and $\{s_k^J\}_{0\le k\le N-1}$, we adopt the most natural reconstruction algorithm to recover the original vector $\{s_k^0\}_{0\le k\le N-1}$. Let us use the expressions in the Fourier domain (7.29) and (7.30). Multiplying (7.29) by the filter $m_0(2^{j-1}\xi)$, (7.30) by the filter $m_1(2^{j-1}\xi)$, adding the results, and using (7.10), we obtain

$$m_0(2^{j-1}\xi)\ \hat{s}^j(\xi) + m_1(2^{j-1}\xi)\ \hat{d}^j(\xi) = \sqrt{2}\ \hat{s}^{j-1}(\xi). \tag{7.43}$$

This expression is equivalent to

$$s_k^{j-1} = \frac{1}{2}\sum_{l=0}^{L-1}(h_l s^j_{k-2^{j-1}l} + g_l d^j_{k-2^{j-1}l}), \tag{7.44}$$

for $j = 1, \ldots, J,\ k = 0, \ldots, N-1$. As shown in the previous subsection, the orthonormal shell coefficients on the jth scale are twice as redundant as on the $(j-1)$st scale, and the factor $\frac{1}{2}$ in (7.44) accounts for this.

7.3 Autocorrelation Shell: A Symmetric Shift-Invariant Multiresolution Representation

The representation of a signal in the orthonormal shell is shift invariant. This representation, however, has several drawbacks for certain applications, such as detecting and characterizing edges (or singularities) in the signal, where the scale-to-scale analysis of the coefficients is necessary [66], [95], [98]. This representation lacks symmetry because of the asymmetric shapes of compactly supported wavelets. It results in a rather complicated representation even in the case of the unit impulse sequence, as may be seen in Figure 7.2. Also, because of the "rough" shape of compactly supported wavelets, there might be "too many" zero-crossings in this representation.

In this section we introduce the notion of an *autocorrelation shell* of compactly supported wavelets, that is, a shell formed by dilations and translations of the autocorrelation functions of compactly supported wavelets. The decomposition filters associated with these autocorrelation functions naturally have symmetric shapes which simplifies the scale-to-scale analysis of the coefficients. One of the interesting features of this representation is its convertibility to the orthonormal shell of the corresponding compactly supported wavelets on each scale independently of other scales. The algorithm for such conversion is discussed in detail in this section. We also investigate the subsampled version of the autocorrelation shell. It turns out to be possible to reconstruct the original signal, if we store a single additional number at each scale: the Nyquist frequency component of the signal on that scale.

7.3.1 Properties of the autocorrelation functions of compactly supported wavelets

Let us first summarize the properties of the autocorrelation functions of a compactly supported scaling function $\varphi(x)$ and the corresponding wavelet $\psi(x)$. By definition of the autocorrelation function, we have

$$\Phi(x) = \int_{-\infty}^{+\infty} \varphi(y)\varphi(y-x)\,dy, \tag{7.45}$$

$$\Psi(x) = \int_{-\infty}^{+\infty} \psi(y)\psi(y-x)\,dy. \tag{7.46}$$

Given the fact that $\{\varphi(x-k)\}_{0\le k\le N-1}$ and $\{\psi(x-k)\}_{0\le k\le N-1}$ form orthonormal bases on $\mathcal{V}_0$ and $\mathcal{W}_0$ respectively, we immediately obtain that at integer points

$$\Phi(k) = \delta_{0k}, \tag{7.47}$$

$$\Psi(k) = \delta_{0k}, \tag{7.48}$$

where δ_{0k} denotes the Kronecker delta.

The Fourier transforms of the autocorrelation functions in (7.45) and (7.46) are as follows:

$$\hat{\Phi}(\xi) = |\hat{\varphi}(\xi)|^2, \tag{7.49}$$

$$\hat{\Psi}(\xi) = |\hat{\psi}(\xi)|^2. \tag{7.50}$$

By taking the Fourier transforms of (7.7) and (7.8), we have

$$\hat{\varphi}(\xi) = m_0(\xi/2)\hat{\varphi}(\xi/2), \tag{7.51}$$

$$\hat{\psi}(\xi) = m_1(\xi/2)\hat{\varphi}(\xi/2). \tag{7.52}$$

Using (7.51), (7.52), we obtain

$$\hat{\Phi}(\xi) = |m_0(\xi/2)|^2 \hat{\Phi}(\xi/2), \tag{7.53}$$

$$\hat{\Psi}(\xi) = |m_1(\xi/2)|^2 \hat{\Phi}(\xi/2). \tag{7.54}$$

Equation (7.10) also implies that

$$\hat{\Phi}(\xi) + \hat{\Psi}(\xi) = \hat{\Phi}(\xi/2). \tag{7.55}$$

Since $|m_0(\xi)|^2$ is an even function, we have

$$|m_0(\xi)|^2 = \frac{1}{2} + \frac{1}{2} \sum_{k=1}^{L/2} a_{2k-1} \cos(2k-1)\xi, \tag{7.56}$$

where $\{a_k\}$ are the autocorrelation coefficients of the filter H,

$$a_k = 2 \sum_{l=0}^{L-1-k} h_l h_{l+k} \qquad \text{for} \quad k = 1, \ldots, L-1, \tag{7.57}$$

and

$$a_{2k} = 0 \qquad \text{for} \quad k = 1, \ldots, L/2 - 1. \tag{7.58}$$

The coefficients $\{a_{2k-1}\}_{1 \le k \le L/2}$ were used in [12] for computing representations of derivatives and convolution operators in the bases of compactly supported wavelets.

Using (7.7), (7.53), and (7.56), it is easy to derive

$$\Phi(x) = \Phi(2x) + \frac{1}{2} \sum_{l=1}^{L/2} a_{2l-1} \left(\Phi(2x - 2l + 1) + \Phi(2x + 2l - 1) \right), \tag{7.59}$$

$$\Psi(x) = \Phi(2x) - \frac{1}{2} \sum_{l=1}^{L/2} a_{2l-1} \left(\Phi(2x - 2l + 1) + \Phi(2x + 2l - 1) \right). \tag{7.60}$$

By direct examination of (7.59) and (7.60), we obtain that both Φ and Ψ are supported within the interval $[-L+1, L-1]$.

Finally, $\Phi(x)$ and $\Psi(x)$ have vanishing moments [12], namely

$$\mathcal{M}_{\Psi}^{m} = \int_{-\infty}^{+\infty} x^m \Psi(x)\, dx = 0, \qquad \text{for} \quad 0 \leq m \leq L, \tag{7.61}$$

$$\mathcal{M}_{\Phi}^{m} = \int_{-\infty}^{+\infty} x^m \Phi(x)\, dx = 0, \qquad \text{for} \quad 1 \leq m \leq L, \tag{7.62}$$

and

$$\int_{-\infty}^{+\infty} \Phi(x)\, dx = 1. \tag{7.63}$$

Since L consecutive moments of the autocorrelation function $\Psi(x)$ vanish (7.61), we have

$$\hat{\Psi}(\xi) = O(\xi^L). \tag{7.64}$$

Therefore, $\hat{\Psi}(\xi)$ may be viewed as the symbol of a pseudo-differential operator which behaves like an approximation of the derivative operator $(d/dx)^L$. We note that due to its definition this operator may be calculated recursively. The convolution with the function $\Psi(x)$ has two properties useful in edge detection (see [101]): it behaves essentially like a differential operator in detecting spatial intensity changes and it is designed to act at any desired scale.

Finally, we display functions $\Phi(x)$, $\varphi(x)$, $\Psi(x)$, $\psi(x)$, and the magnitudes of their Fourier transforms in Figures 7.3 and 7.4. In these figures, we have used the Daubechies wavelet with two vanishing moments and $L = 4$. It is easy to see that the autocorrelation functions $\Phi(x)$ and $\Psi(x)$ are smoother than the functions $\varphi(x)$ and $\psi(x)$ and $\hat{\Phi}(\xi)$ and $\hat{\Psi}(\xi)$ decay faster than $\hat{\varphi}(\xi)$ and $\hat{\psi}(\xi)$ respectively. We also note that both $\Phi(x)$ and $\Psi(x)$ are even.

Remark 7.2. It follows from (7.55) or (7.59) and (7.60) that

$$\Psi(x) = 2\Phi(2x) - \Phi(x). \tag{7.65}$$

This may be compared with the approximation of the Laplacian of a Gaussian function (the so-called Mexican-hat function) by the Difference of two Gaussian functions (the so-called

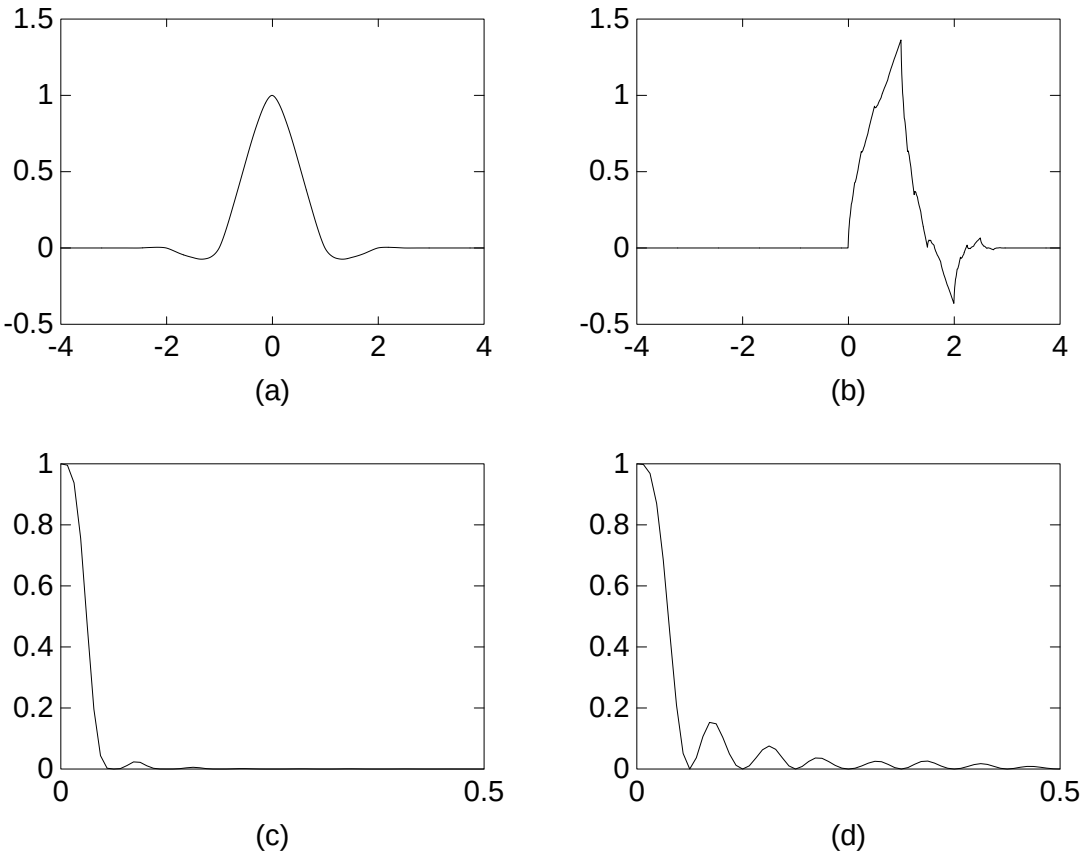


Figure 7.3: Plots of the autocorrelation function $\Phi(x)$ and the Daubechies scaling function $\varphi(x)$ with $L = 4$. (a) $\Phi(x)$. (b) $\varphi(x)$. (c) Magnitude of the Fourier transform of $\Phi(x)$. (d) Magnitude of the Fourier transform of $\varphi(x)$.

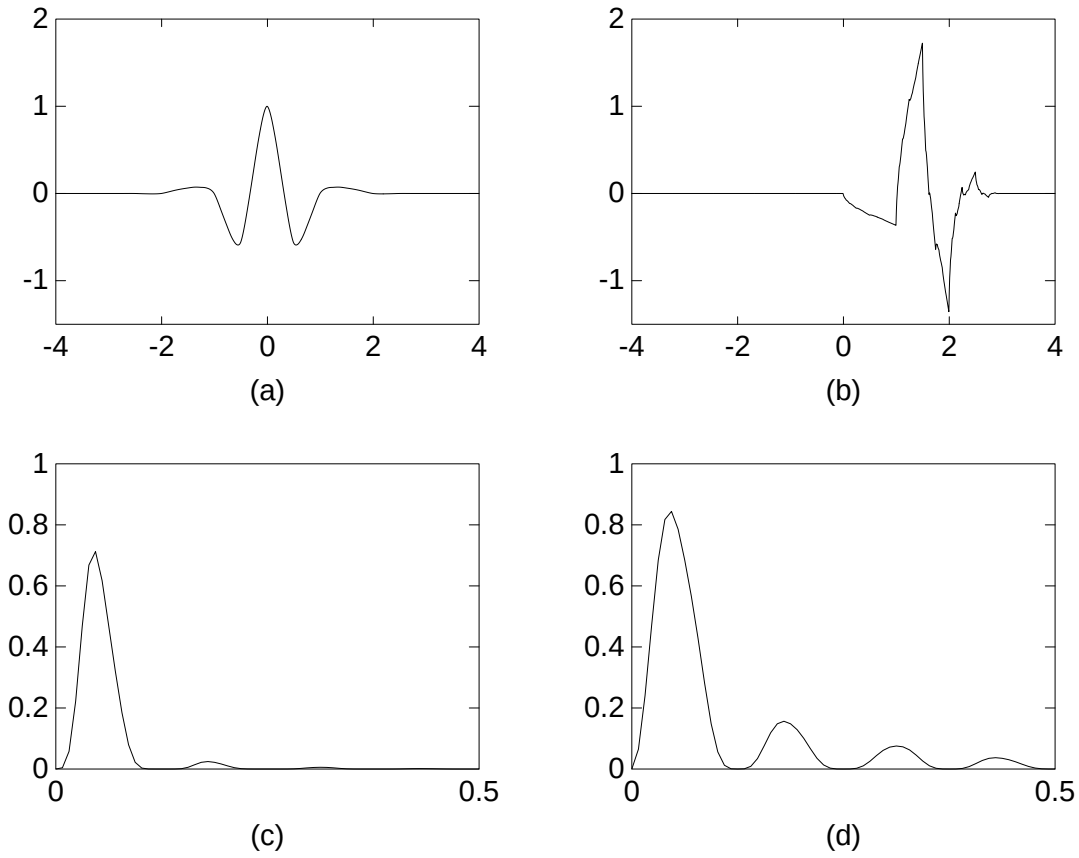


Figure 7.4: Plots of the autocorrelation function $\Psi(x)$ and the Daubechies wavelet $\psi(x)$ with two vanishing moments and $L = 4$. (a) $\Psi(x)$. (b) $\psi(x)$. (c) Magnitude of the Fourier transform of $\Psi(x)$. (d) Magnitude of the Fourier transform of $\psi(x)$.

DOG function) as

$$\frac{d^2}{dx^2}G(x;\sigma) \approx G(x;a\sigma) - G(x;\sigma) \tag{7.66}$$

$$= aG(ax;\sigma) - G(x;\sigma), \tag{7.67}$$

where

$$G(x;\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{-x^2/2\sigma^2}, \tag{7.68}$$

and $a = 1.6$ as Marr suggested in [101]. Interestingly enough, Marr and Hildreth compared the DOG function with the difference of two boxcar functions and the difference of two sinc functions (i.e., the ideal bandpass filter) in their edge detection performances [102]. They claimed the superiority of the DOG function over the other two by saying that the latter two functions are too localized either in space domain or in spatial frequency domain.

7.3.2 The autocorrelation shell of compactly supported wavelets

By analogy with the previous section, let us now consider the following family of functions,

$$\left\{\widetilde{\Psi}_{j,k}(x)\right\}_{1\le j\le J,\ 0\le k\le N-1} \quad \text{and} \quad \left\{\widetilde{\Phi}_{J,k}(x)\right\}_{0\le k\le N-1}, \tag{7.69}$$

where

$$\widetilde{\Psi}_{j,k}(x) = 2^{-j/2}\Psi(2^{-j}(x-k)), \tag{7.70}$$

$$\widetilde{\Phi}_{J,k}(x) = 2^{-J/2}\Phi(2^{-J}(x-k)). \tag{7.71}$$

We call this family a shell of autocorrelations of compactly supported wavelets for shifts in $\mathcal{V}_0$. Henceforth, we call this family an *autocorrelation shell* for short.

The relation to the orthonormal shell

Let us now consider the relation between the autocorrelation shell and the orthonormal shell. Let $f \in \mathcal{V}_0$ so that

$$f(x) = \sum_{k=0}^{N-1} s_k^0 \varphi(x-k), \tag{7.72}$$

and consider the function $\mathcal{A}f$,

$$\mathcal{A}f(x) = \sum_{k=0}^{N-1} s_k^0 \Phi(x-k). \tag{7.73}$$

These two functions are related via

$$\mathcal{A}f(x) = \int f(y)\varphi(y-x)\,dy, \tag{7.74}$$

and

$$\mathcal{A}f(k) = s_k^0, \qquad \text{for} \quad k = 0, 1, \ldots, N-1. \tag{7.75}$$

Similarly at the scale j, let us define functions $f_s^j(x)$ and $f_d^j(x)$ using the orthonormal shell coefficients s_k^j and d_k^j,

$$f_s^j(x) = \sum_{k=0}^{N-1} s_k^j \varphi(x-k), \tag{7.76}$$

$$f_d^j(x) = \sum_{k=0}^{N-1} d_k^j \varphi(x-k). \tag{7.77}$$

By correlating the functions $f_s^j(x)$ and $f_d^j(x)$ with the functions $2^{-j}\varphi(2^{-j}x)$ and $2^{-j}\psi(2^{-j}x)$ on each scale j, we obtain

$$\mathcal{A}_s^j f(x) = \int f_s^j(y)\ 2^{-j}\varphi(2^{-j}(y-x))\,dy = \sum_{k=0}^{N-1} S_k^j \Phi(x-k), \tag{7.78}$$

$$\mathcal{A}_d^j f(x) = \int f_d^j(y)\ 2^{-j}\psi(2^{-j}(y-x))\,dy = \sum_{k=0}^{N-1} D_k^j \Phi(x-k), \tag{7.79}$$

where we define the coefficients $\{S_k^j\}$ and $\{D_k^j\}$ to be the *autocorrelation shell coefficients* (averages and differences). From (7.47), the coefficients $\{S_k^j\}$ and $\{D_k^j\}$ are the values of $\mathcal{A}_s^j f(x)$ and $\mathcal{A}_d^j f(x)$ at integer points:

$$S_k^j = \mathcal{A}_s^j f(k) = \int f_s^j(y)\ 2^{-j}\varphi(2^{-j}(y-k))\,dy, \tag{7.80}$$

$$D_k^j = \mathcal{A}_d^j f(k) = \int f_d^j(y)\ 2^{-j}\psi(2^{-j}(y-k))\,dy. \tag{7.81}$$

To summarize, the coefficients $\{S_k^j\}$ and $\{D_k^j\}$ may be obtained as follows:

Step 1. Expand the function $f \in \mathcal{V}_0$ in the orthonormal shell and obtain the coefficients $\{s_k^j\}$ and $\{d_k^j\}$ (see Section 7.2).

Step 2. Expand the unit impulse $\{\delta_{0k}\}$ in the orthonormal shell and obtain the coefficients $\{v_k^j\}$ and $\{w_k^j\}$, which are the values of $2^{-j/2}\varphi^{\triangleleft}(2^{-j}x)$ and $2^{-j/2}\psi^{\triangleleft}(2^{-j}x)$ at integer points x.

Step 3. For each scale j, correlate the coefficients $\{s_k^j\}$ and $\{d_k^j\}$ with the coefficients of the unit impulse $\{v_k^j\}$ and $\{w_k^j\}$ respectively, and multiply the results by the factor $2^{-j/2}$.

Remark 7.3. In applications of autocorrelation shell representations, one may assume that the original continuous signal is the function $\mathcal{A}f(x)$ rather than $f(x)$. In this case $\{s_k^0\}$ are the values of $\mathcal{A}f(x)$ at integer points.

A fast algorithm for expanding in an autocorrelation shell

Let us now derive a fast algorithm for expanding in an autocorrelation shell, i.e., the pyramid algorithm for computing $\{S_k^j\}$ and $\{D_k^j\}$ from $\{S_k^{j-1}\}$. First, let us define the coefficients $\{p_k\}$ and $\{q_k\}$ by rewriting the equations (7.59) and (7.60) as

$$\frac{1}{\sqrt{2}}\Phi(x/2) = \sum_{k=-L+1}^{L-1} p_k \Phi(x-k), \tag{7.82}$$

$$\frac{1}{\sqrt{2}}\Psi(x/2) = \sum_{k=-L+1}^{L-1} q_k \Phi(x-k), \tag{7.83}$$

where

$$p_k = \begin{cases} 2^{-1/2} & \text{for } k = 0 \\ 2^{-3/2} a_{|k|} & \text{for } k = \pm 1, \pm 3, \ldots, \pm(L-1) \\ 0 & \text{for } k = \pm 2, \pm 4, \ldots, \pm(L-2) \end{cases} \tag{7.84}$$

$$q_k = \begin{cases} 2^{-1/2} & \text{for } k = 0 \\ -p_k & \text{otherwise} \end{cases} \tag{7.85}$$

We view these coefficients as filters $P = \{p_k\}_{-L+1 \le k \le L-1}$ and $Q = \{q_k\}_{-L+1 \le k \le L-1}$. It is clear that $p_k = p_{-k}$ and $q_k = q_{-k}$ (see (7.84) and (7.85)) and so there are only $L/2 + 1$ distinct non-zero coefficients. By taking Fourier transforms of both sides of (7.82) and (7.83) and using (7.53) and (7.54), we obtain

$$\sqrt{2}|m_0(\xi)|^2 = \sum_{k=-L+1}^{L-1} p_k \mathrm{e}^{\mathrm{i}k\xi}, \tag{7.86}$$

$$\sqrt{2}|m_1(\xi)|^2 = \sum_{k=-L+1}^{L-1} q_k \mathrm{e}^{\mathrm{i}k\xi}. \tag{7.87}$$

Thus, using Φ and Ψ for the decomposition is equivalent to viewing $\sqrt{2}|m_0(\xi)|^2$ and $\sqrt{2}|m_1(\xi)|^2$ as decomposition filters instead of $m_0(\xi)$ and $m_1(\xi)$ in the orthonormal case. We note that the pair of filters $\sqrt{2}|m_0(\xi)|^2$ and $\sqrt{2}|m_1(\xi)|^2$ is not a quadrature mirror pair since $2|m_0(\xi)|^4 + 2|m_1(\xi)|^4 \neq 1$. Using the filters P and Q, we obtain the pyramid algorithm for expanding into the autocorrelation shell,

$$S_k^j = \sum_{l=-L+1}^{L-1} p_l S_{k+2^{j-1}l}^{j-1}, \tag{7.88}$$

$$D_k^j = \sum_{l=-L+1}^{L-1} q_l S_{k+2^{j-1}l}^{j-1}. \tag{7.89}$$

As an example of the coefficients $\{p_k\}$, for Daubechies's wavelets with two vanishing moments, $L = 4$ and the coefficients are $2^{-1/2}(-\frac{1}{16}, 0, \frac{9}{16}, 1, \frac{9}{16}, 0, -\frac{1}{16})$.

The direct conversion from an autocorrelation shell to an orthonormal shell

We now consider an algorithm for the direct conversion of $\{S_k^j\}$ and $\{D_k^j\}$ to $\{s_k^j\}$ and $\{d_k^j\}$. Let us first derive a recursion relationship similar to (7.39) and (7.40) for the orthonormal shell expansion. From (7.88), (7.89), (7.86), and (7.87), we obtain

$$\hat{S}^j(\xi) = \hat{s}^0(\xi)\ 2^{j/2} \prod_{l=1}^{j} |m_0(2^{l-1}\xi)|^2, \tag{7.90}$$

$$\hat{D}^j(\xi) = \hat{s}^0(\xi)\ 2^{j/2} |m_1(2^{j-1}\xi)|^2 \prod_{l=1}^{j-1} |m_0(2^{l-1}\xi)|^2. \tag{7.91}$$

On defining the functions

$$m_0^j(\xi) = \prod_{l=1}^{j} m_0(2^{l-1}\xi), \tag{7.92}$$

so that

$$m_1^j(\xi) = m_1(2^{j-1}\xi) m_0^{j-1}(\xi), \tag{7.93}$$

we obtain from (7.39), (7.40) , (7.90), and (7.91)

$$\hat{S}^j(\xi) = m_0^j(\xi)\hat{s}^j(\xi), \tag{7.94}$$

$$\hat{D}^j(\xi) = m_1^j(\xi)\hat{d}^j(\xi). \tag{7.95}$$

Thus, to compute the autocorrelation shell coefficients from the orthonormal shell coefficients, we have to evaluate the following two equations:

$$\hat{s}^j(\xi) = \frac{\overline{m_0^j(\xi)}}{|m_0^j(\xi)|^2} \hat{S}^j(\xi), \tag{7.96}$$

$$\hat{d}^j(\xi) = \frac{\overline{m_1^j(\xi)}}{|m_1^j(\xi)|^2}\hat{D}^j(\xi). \tag{7.97}$$

In general, such computation leads to an unstable algorithm due to zeros of the denominators. In our case, however, this procedure is stable and the division by $|m_0^j(\xi)|^2$ and $|m_1^j(\xi)|^2$ in (7.96) and (7.97) does not cause problems because the numerators are always zero when the denominators are zero (see Proposition 7.4 below).

We observe that transforming back and forth between the orthonormal shell representation and the autocorrelation shell representation is done on each scale separately.

Properties of the autocorrelation shell

Let us define the following norm for the autocorrelation shell expansion:

$$\|f\|_{\mathcal{A}}^2 = \sum_{j=1}^{J} 2^{-j} \sum_{k=0}^{N-1} (D_k^j)^2 + 2^{-J} \sum_{k=0}^{N-1} (S_k^J)^2, \tag{7.98}$$

where the coefficients D_k^j and S_k^J are defined in (7.81) and (7.80). The norm (7.98) may be compared with the norm for the orthonormal shell (7.23). As for the orthonormal shell, the factor 2^{-j} in (7.98) is used to offset the redundancy of this representation. We now obtain the following estimate:

Proposition 7.4.

$$\frac{1}{J+1}\|f\|^2 \leq \|f\|_{\mathcal{A}}^2 \leq \|f\|^2, \tag{7.99}$$

where $\|f\|$ is the L^2 norm of the vector f in $\mathcal{V}_0$ defined in (7.19).

This inequality guarantees that there exists a stable reconstruction algorithm from the autocorrelation shell coefficients. If the number of dyadic scales $J \to \infty$, then the lower bound in (7.99) approaches zero and the reconstruction algorithm becomes unstable. In feasible practical applications, however, the number of dyadic scales does not exceed a

small constant (e.g., $J \leq 50$), and the estimate (7.99) is sufficient for a stable reconstruction. It also shows that our construction is limited to finite dimensional subspaces.

Proof. From (7.26) we have

$$\|f\|^2 = \|f\|_{\mathcal{S}}^2 = \sum_{j=1}^{J} 2^{-j} \sum_{k=0}^{N-1} (d_k^j)^2 + 2^{-J} \sum_{k=0}^{N-1} (s_k^J)^2. \tag{7.100}$$

Using Parseval's equality, we have the following Fourier domain expressions:

$$\sum_{k=0}^{N-1} (s_k^j)^2 = \frac{1}{N} \sum_{k=0}^{N-1} |\hat{s}_k^j|^2, \tag{7.101}$$

$$\sum_{k=0}^{N-1} (d_k^j)^2 = \frac{1}{N} \sum_{k=0}^{N-1} |\hat{d}_k^j|^2. \tag{7.102}$$

Using the expressions (7.39),(7.40),(7.92), and (7.93), we rewrite (7.100) as

$$\|f\|_{\mathcal{S}}^2 = \frac{1}{N} \left[\sum_{j=1}^{J} \sum_{k=0}^{N-1} |\hat{s}_k^0|^2 \; |m_1^j(\xi_k)|^2 + \sum_{k=0}^{N-1} |\hat{s}_k^0|^2 \; |m_0^J(\xi_k)|^2 \right], \tag{7.103}$$

where $\xi_k = 2\pi k/N$. Similarly, for the norm of the autocorrelation shell expansion (7.98), we have

$$\|f\|_{\mathcal{A}}^2 = \sum_{j=1}^{J} 2^{-j} \sum_{k=0}^{N-1} (D_k^j)^2 + 2^{-J} \sum_{k=0}^{N-1} (S_k^J)^2. \tag{7.104}$$

With the same arguments, the Fourier domain expression of (7.104) becomes

$$\|f\|_{\mathcal{A}}^2 = \frac{1}{N} \left[\sum_{j=1}^{J} \sum_{k=0}^{N-1} |\hat{s}_k^0|^2 \; |m_1^j(\xi_k)|^4 + \sum_{k=0}^{N-1} |\hat{s}_k^0|^2 \; |m_0^J(\xi_k)|^4 \right]. \tag{7.105}$$

Since

$$\sup_{0 \leq \xi \leq 2\pi} |m_0^j(\xi)|^2 \leq 1 \quad \text{and} \quad \sup_{0 \leq \xi \leq 2\pi} |m_1^j(\xi)|^2 \leq 1, \tag{7.106}$$

we immediately have from (7.105)

$$\|f\|_{\mathcal{A}}^2 \leq \|f\|_{\mathcal{S}}^2. \tag{7.107}$$

As for the rest of the inequality of (7.99), using the Schwarz inequality in (7.103) we have

$$\begin{aligned}
\|f\|_{\mathcal{S}}^2 &\le \frac{1}{N}\Bigg[\sum_{j=1}^{J}\left(\sum_{k=0}^{N-1}|\hat{s}_k^0|^2\right)^{1/2}\left(\sum_{k=0}^{N-1}|\hat{s}_k^0|^2\ |m_1^j(\xi_k)|^4\right)^{1/2} \\
&\qquad + \left(\sum_{k=0}^{N-1}|\hat{s}_k^0|^2\right)^{1/2}\left(\sum_{k=0}^{N-1}|\hat{s}_k^0|^2\ |m_0^J(\xi_k)|^4\right)^{1/2}\Bigg] \\
&= \frac{\|f\|}{\sqrt{N}}\left[\sum_{j=1}^{J}\left(\sum_{k=0}^{N-1}|\hat{s}_k^0|^2\ |m_1^j(\xi_k)|^4\right)^{1/2} + \left(\sum_{k=0}^{N-1}|\hat{s}_k^0|^2\ |m_0^J(\xi_k)|^4\right)^{1/2}\right] \qquad (7.108) \\
&\le \sqrt{J+1}\ \frac{\|f\|}{\sqrt{N}}\left(\sum_{j=1}^{J}\sum_{k=0}^{N-1}|\hat{s}_k^0|^2\ |m_1^j(\xi_k)|^4 + \sum_{k=0}^{N-1}|\hat{s}_k^0|^2\ |m_0^J(\xi_k)|^4\right)^{1/2} \\
&= \sqrt{J+1}\ \|f\|\ \|f\|_{\mathcal{A}}.
\end{aligned}$$

Since $\|f\| = \|f\|_{\mathcal{S}}$ (see (7.26)), we obtain (7.99). □

The following proposition (which is similar to Proposition 7.1) is essential in our approach to the reconstruction of signals from zero-crossings.

Proposition 7.5. *For any function $f \in \mathcal{V}_0$, $f(x) = \sum_{k=0}^{N-1} s_k^0 \varphi(x-k)$, the coefficients $\{S_k^j\}$ and $\{D_k^j\}$ defined in* (7.80) *and* (7.81) *satisfy the following identities*

$$\sum_{k=0}^{N-1} S_k^j \widetilde{\Phi}_{0,k} = \sum_{k=0}^{N-1} s_k^0 \widetilde{\Phi}_{j,k} \qquad (7.109)$$

$$\sum_{k=0}^{N-1} D_k^j \widetilde{\Phi}_{0,k} = \sum_{k=0}^{N-1} s_k^0 \widetilde{\Psi}_{j,k}, \qquad (7.110)$$

where $\widetilde{\Phi}_{0,k}(x) = \Phi_{0,k}(x)$, and $\widetilde{\Phi}_{j,k}$ and $\widetilde{\Psi}_{j,k}$ are defined in (7.71) *and* (7.70).

Proof. By taking the Fourier transform of the left hand side of (7.109) and using (7.90), we have

$$\hat{S}^j(\xi)\hat{\Phi}(\xi) = \hat{s}^0(\xi)\ 2^{j/2}\prod_{l=1}^{j}|m_0(2^{l-1}\xi)|^2\ \hat{\Phi}(\xi) = \hat{s}^0(\xi)\ 2^{j/2}\hat{\Phi}(2^j\xi), \qquad (7.111)$$

where we have used the identity

$$\hat{\Phi}(\xi) = \prod_{l=1}^{\infty} |m_0(2^{-l}\xi)|^2. \tag{7.112}$$

The inverse Fourier transform of (7.111) yields (7.109). The relation (7.110) may be derived similarly. □

Examples of the autocorrelation shell expansion

Let us illustrate the autocorrelation shell expansion using several examples. In Figure 7.5 we show the expansion of the unit impulse and its shifted version in the autocorrelation shell to illustrate the symmetry, smoothness, and shift invariance of this representation (see Figure 7.2 for comparison). In Figure 7.6 we display an example one-dimensional signal which is a natural radioactivity profile of certain subsurface formations used in Example 3.11 in Chapter 3. The autocorrelation shell coefficients of this signal are shown in Figure 7.7. Figure 7.8 shows the corresponding average coefficients. In this case, we have used the autocorrelation functions of the Daubechies wavelet with two vanishing moments and $L = 4$.

7.3.3 A direct reconstruction of signals from the autocorrelation shell coefficients

As we have shown in the previous subsection, the original signal may be reconstructed from the autocorrelation shell coefficients by converting them to the orthonormal shell coefficients followed by the reconstruction algorithm (7.44). Let us now construct an algorithm for reconstructing the original signal directly from the autocorrelation shell coefficients. Rewriting (7.88) and (7.89) and using the coefficients $\{a_k\}$ of (7.57), we have

$$S_k^j = \frac{1}{\sqrt{2}} \left[S_k^{j-1} + \frac{1}{2} \sum_{l=1}^{L/2} a_{2l-1} \left(S_{k+2^{j-1}(2l-1)}^{j-1} + S_{k-2^{j-1}(2l-1)}^{j-1} \right) \right], \tag{7.113}$$

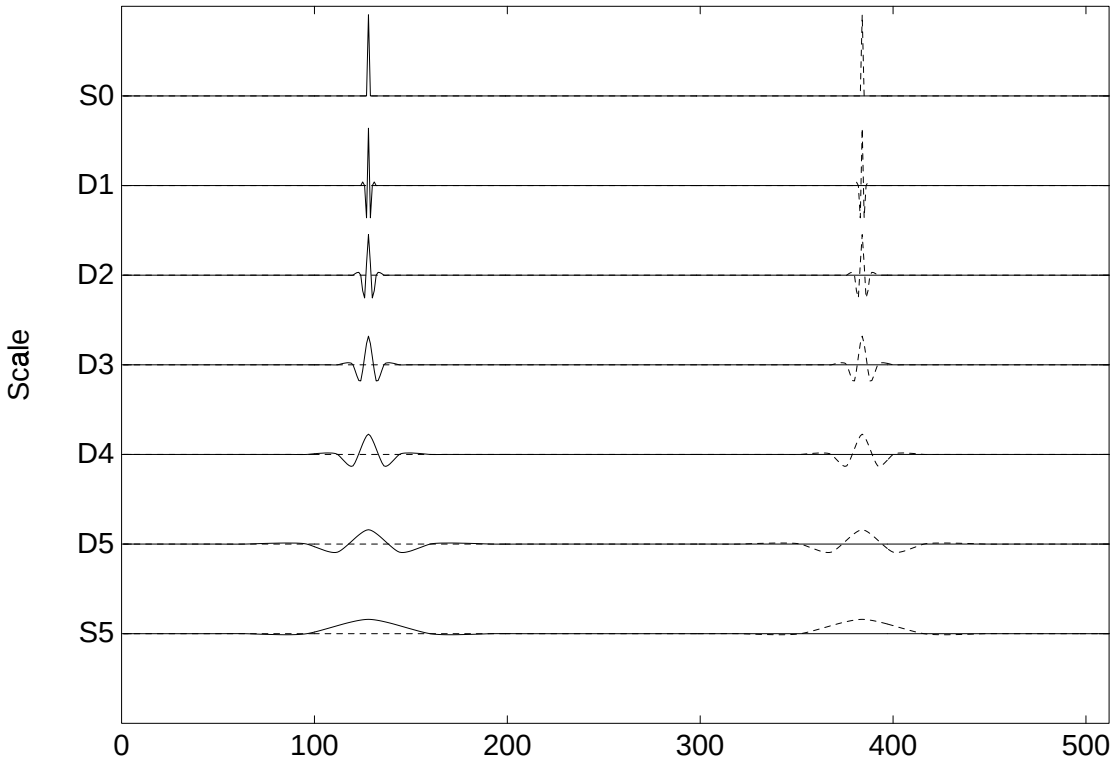


Figure 7.5: The expansion of two unit impulses in the autocorrelation shell using the autocorrelation functions of the Daubechies wavelet with $L = 2M = 4$.

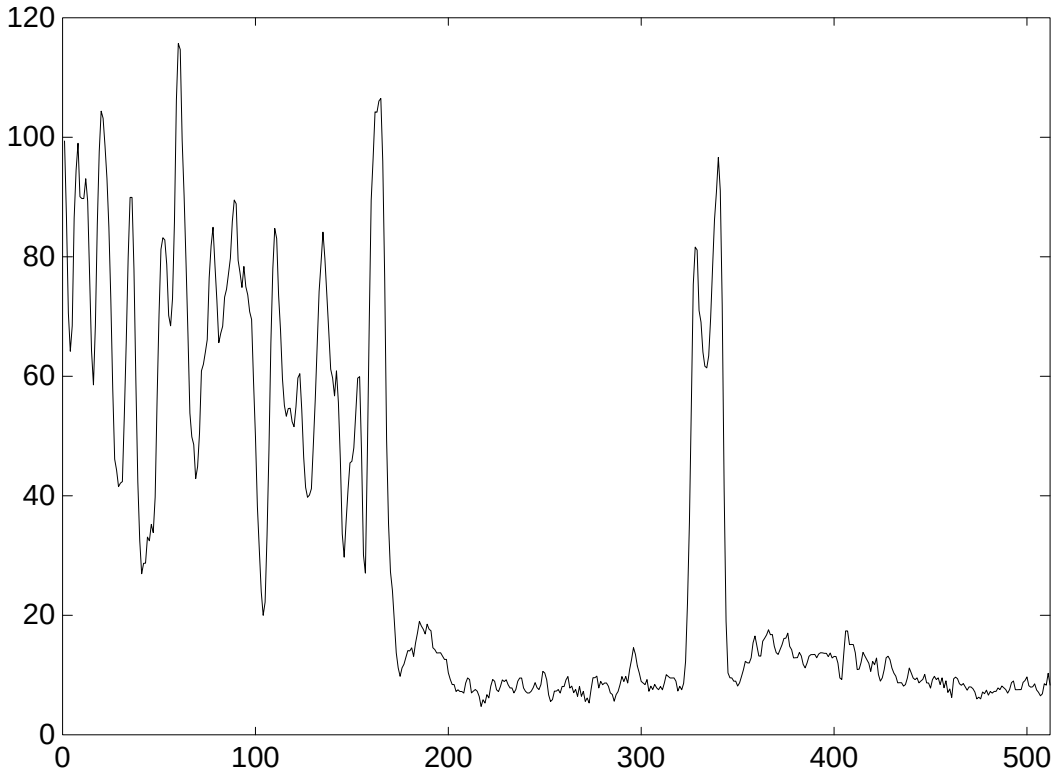


Figure 7.6: The original signal (representing a natural radioactivity of certain subsurface formations) which will be used as an example for the autocorrelation shell expansion.

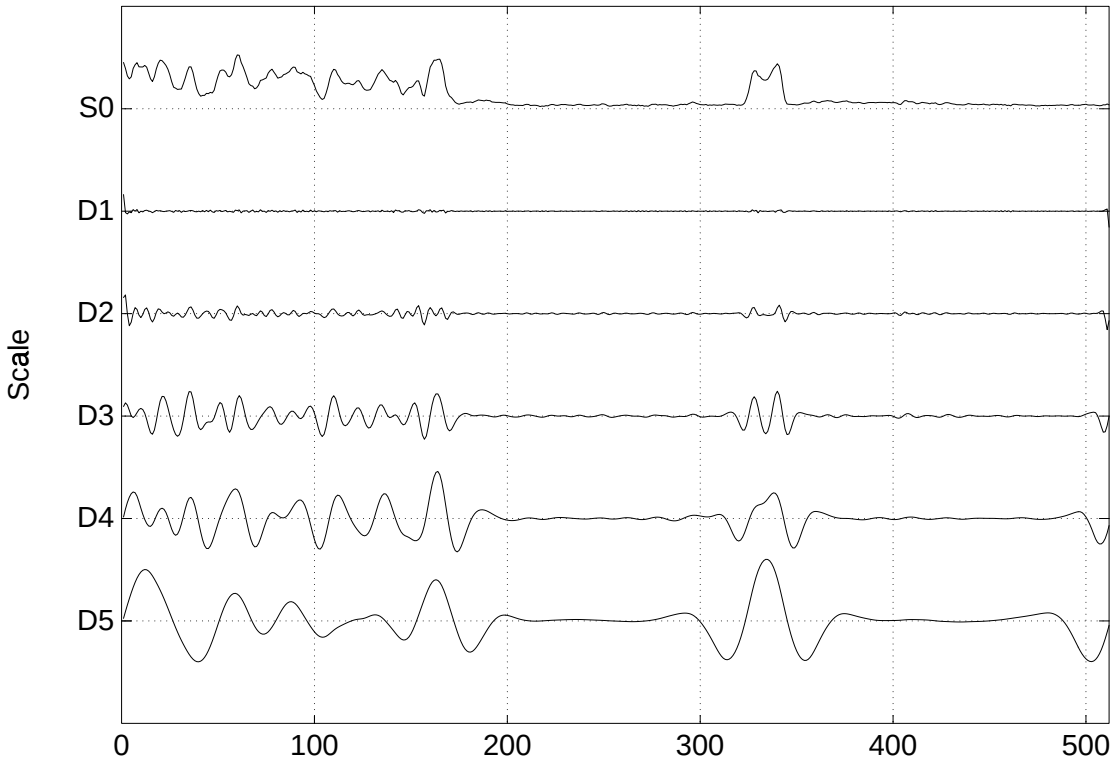


Figure 7.7: The expansion of the signal shown in Figure 7.6 in the autocorrelation shell using the autocorrelation functions of the Daubechies wavelet with $L = 2M = 4$. The top row is the original signal $\{S_k^0\}$. The coefficients $\{D_k^j\}_{1 \le j \le 5,\ 0 \le k \le 511}$ are shown from the second to the last row in this figure. Note that the locations of edges in the original signal correspond to the zero-crossings in this representation.

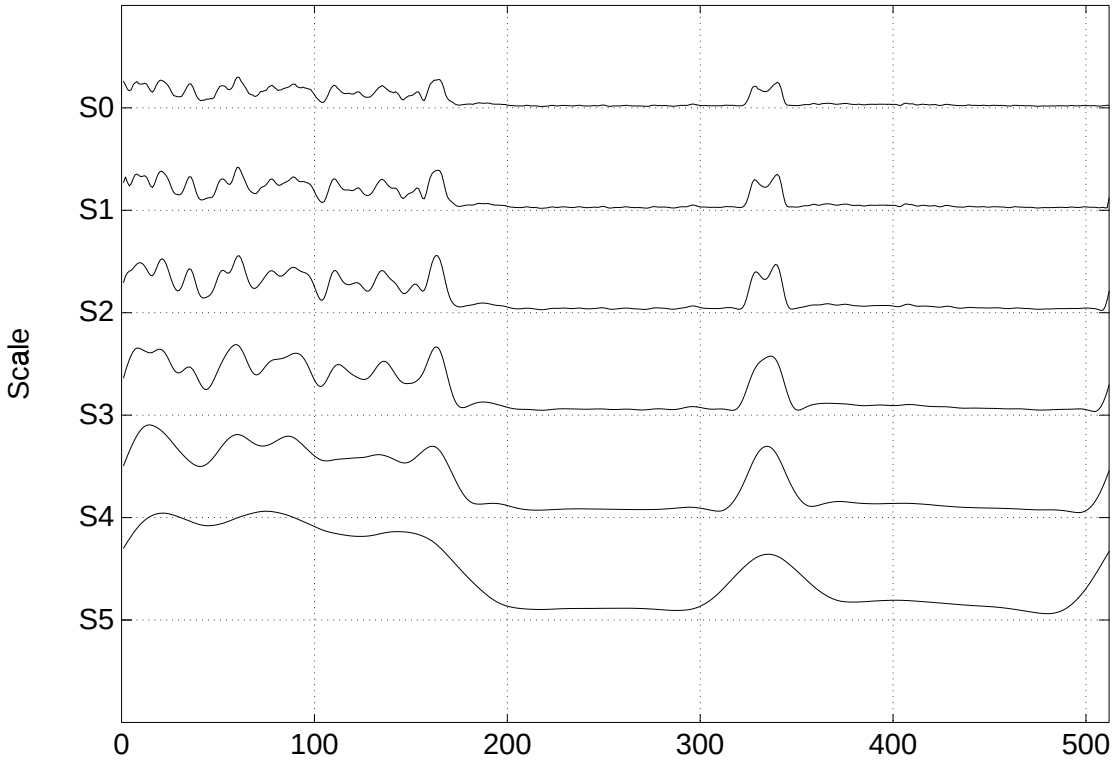


Figure 7.8: Plots of the average coefficients on different scales in the autocorrelation shell representation of the signal shown in Figure 7.6. The top row is the original signal.

$$D_k^j = \frac{1}{\sqrt{2}} \left[S_k^{j-1} - \frac{1}{2} \sum_{l=1}^{L/2} a_{2l-1} \left(S_{k+2^{j-1}(2l-1)}^{j-1} + S_{k-2^{j-1}(2l-1)}^{j-1} \right) \right], \tag{7.114}$$

for $j = 1, \ldots, J,\ k = 0, \ldots, N-1$. By adding (7.113) and (7.114), we obtain a simple reconstruction formula,

$$S_k^{j-1} = \frac{1}{\sqrt{2}} \left(S_k^j + D_k^j \right), \tag{7.115}$$

for $j = 1, \ldots, J,\ k = 0, \ldots, N-1$. Given the autocorrelation shell coefficients $\{D_k^j\}_{1 \le j \le J,\ 0 \le k \le N-1}$ and $\{S_k^J\}_{0 \le k \le N-1}$, (7.115) leads to

$$s_k^0 = 2^{-J/2} S_k^J + \sum_{j=1}^{J} 2^{-j/2} D_k^j, \tag{7.116}$$

for $k = 0, \ldots, N-1$.

Similarly to the orthonormal shell coefficients, $\{S_k^j\}$ and $\{D_k^j\}$ are redundant. This may be used to reconstruct a signal from subsampled autocorrelation shell coefficients.

7.3.4 The subsampled autocorrelation shell

We now turn to a question of reconstructing the original vector from a subsampled sequence of the autocorrelation shell coefficients. We demonstrate that it is possible to reconstruct the original signal if we keep a single additional number at each scale of the autocorrelation shell expansion, i.e., the Nyquist frequency components of the autocorrelation shell coefficients.

Let us define the *subsampled autocorrelation shell expansion* of a vector $f \in \mathcal{V}_0$ as follows.

$$\left\{ \sum_{k=0}^{2^{n-j}-1} d_k^j \Psi_{j,k}(x) \right\}_{1 \le j \le J} \quad \text{and} \quad \sum_{k=0}^{2^{n-J}-1} s_k^J \Phi_{J,k}(x), \tag{7.117}$$

where $\Phi_{j,k}$ and $\Psi_{j,k}$ are defined by

$$\Phi_{j,k}(x) = 2^{-j/2} \Phi(2^{-j}x - k), \tag{7.118}$$

$$\Psi_{j,k}(x) = 2^{-j/2}\Psi(2^{-j}x - k), \tag{7.119}$$

and the coefficients s_k^j and d_k^j are defined as the subsamples of the autocorrelation shell coefficients S_k^j and D_k^j as

$$s_k^j = S_{2^j k}^j, \tag{7.120}$$

$$d_k^j = D_{2^j k}^j \tag{7.121}$$

for $k = 0, 1, \ldots, 2^{n-j} - 1$. The notation s_k^j and d_k^j is local to this subsection (not to be confused with the one for the orthonormal wavelets).

The decomposition algorithm is similar to that for the orthonormal wavelets, and we compute

$$s_k^j = \sum_{l=-L+1}^{L-1} p_l s_{2k+l}^{j-1}, \tag{7.122}$$

$$d_k^j = \sum_{l=-L+1}^{L-1} q_l s_{2k+l}^{j-1}, \tag{7.123}$$

for $j = 1, \ldots, J,\ k = 0, \ldots, 2^{n-j} - 1$.

For the reconstruction, we first rewrite (7.122) and (7.123) in terms of the coefficients $\{a_k\}$ as

$$s_k^j = \frac{1}{\sqrt{2}} \left[s_{2k}^{j-1} + \frac{1}{2} \sum_{l=1}^{L/2} a_{2l-1} \left(s_{2k-2l+1}^{j-1} + s_{2k+2l-1}^{j-1} \right) \right], \tag{7.124}$$

$$d_k^j = \frac{1}{\sqrt{2}} \left[s_{2k}^{j-1} - \frac{1}{2} \sum_{l=1}^{L/2} a_{2l-1} \left(s_{2k-2l+1}^{j-1} + s_{2k+2l-1}^{j-1} \right) \right]. \tag{7.125}$$

By adding these two expressions, we obtain the coefficients of the $(j-1)$st scale with even indices,

$$s_{2k}^{j-1} = \frac{1}{\sqrt{2}} \left(s_k^j + d_k^j \right), \tag{7.126}$$

for $j = 1, \ldots, J,\ k = 0, \ldots, 2^{n-j} - 1$. As for the coefficients with odd indices, we first define the sequence,

$$\Delta_k^j = \frac{1}{\sqrt{2}} \left(s_k^j - d_k^j \right) \tag{7.127}$$

$$= \frac{1}{2} \sum_{l=1}^{L/2} a_{2l-1} \left(s_{2k-2l+1}^{j-1} + s_{2k+2l-1}^{j-1} \right). \tag{7.128}$$

By taking the Fourier transform of (7.127), we have

$$\hat{\Delta}^j(\xi) = \sum_{k=0}^{2^{n-j}-1} \Delta_k^j \mathrm{e}^{\mathrm{i}k\xi} \tag{7.129}$$

$$= \frac{1}{2} \sum_{k=0}^{2^{n-j}-1} \sum_{l=1}^{L/2} a_{2l-1} \left(s_{2k+2l-1}^{j-1} \mathrm{e}^{\mathrm{i}k\xi} + s_{2k-2l+1}^{j-1} \mathrm{e}^{\mathrm{i}k\xi} \right) \tag{7.130}$$

$$= \hat{s}_{odd}^{j-1}(\xi/2) \sum_{l=1}^{L/2} a_{2l-1} \cos((2l-1)\xi/2), \tag{7.131}$$

where

$$\hat{s}_{odd}^{j-1}(\xi/2) = \sum_{k=0}^{2^{n-j}-1} s_{2k+1}^{j-1} \mathrm{e}^{\mathrm{i}(2k+1)\xi/2}. \tag{7.132}$$

The division by $\sum_{l=1}^{L/2} a_{2l-1} \cos((2l-1)\xi/2)$ is not defined at the Nyquist frequency at $\xi = \pi$. For the uniqueness of the reconstruction, we compute the Nyquist frequency component at each scale and store it as a part of the decomposition algorithm. We then supply these data at the reconstruction stage. The Nyquist frequency component of odd samples at the $(j-1)$st scale may be simply computed as

$$\sigma_{Nyq}^{j-1} = \hat{s}_{odd}^{j-1}(\pi/2) = \mathrm{i} \sum_{k=0}^{2^{n-j}-1} (-1)^k s_{2k+1}^{j-1}. \tag{7.133}$$

In summary, the coefficients with odd indices can be recovered as the inverse Fourier transform of the following quantity:

$$\hat{s}_{odd}^{j-1}(\xi/2) = \begin{cases} \dfrac{\hat{\Delta}^j(\xi)}{\sum_{l=1}^{L/2} a_{2l-1} \cos((2l-1)\xi/2)} & \text{for } 0 \le \xi < \pi, \\ \\ \sigma_{Nyq}^{j-1} & \text{for } \xi = \pi. \end{cases} \tag{7.134}$$

Remark 7.6. In [20], Burt introduced three different multiresolution image representation schemes: 1) "standard-density" pyramid, 2) "double-density" pyramid, and 3) "full-density" pyramid. The scheme 1) and 3) correspond to the representations using the subsampled shell (or the standard wavelets) and the shell, respectively. The "double-density pyramid" has twice as many expansion coefficients as the standard-density pyramid, at each scale. According to Burt, this scheme is often used in practice for image processing applications. This comment makes sense in our scheme too; if we double the number of coefficients at each scale in the subsampled autocorrelation shell expansion, then there is no need to keep the Nyquist frequency component at each scale.

7.4 A Review of Dubuc's Iterative Interpolation Scheme

The representation using the autocorrelation functions of compactly supported wavelets has a natural interpolation algorithm associated with it. This interpolation scheme is due to Dubuc [55] and has been extended in [47] by Deslauriers and Dubuc. This interpolation scheme may be arrived at by considering the autocorrelation function of the scaling function $\varphi(x)$ (see also [139]). Namely, for every wavelet (not necessarily compactly supported), the autocorrelation function $\Phi(x)$ of the scaling function $\varphi(x)$, gives rise to a symmetric iterative interpolation scheme and the autocorrelation function $\Phi(x)$ is exactly the "fundamental function" $F(x)$ introduced in [55], [47]. In particular, the Daubechies wavelet with the quadrature mirror filters of length $L = 4$ and of two vanishing moments yields the scheme in [55]. And the general case of Daubechies's wavelets with M vanishing moments ($L = 2M$) leads to the iterative interpolation scheme using the Lagrange polynomials of degree $2M$ [47]. Examples of schemes other than the Lagrange iterative interpolation may be obtained using compactly supported wavelets other than those explicitly described in [42] (see e.g., [13], [44]). This interpolation scheme allows us to use a natural algorithm for computing zero-crossings and slopes at the zero-crossings in the autocorrelation shell expansion in an

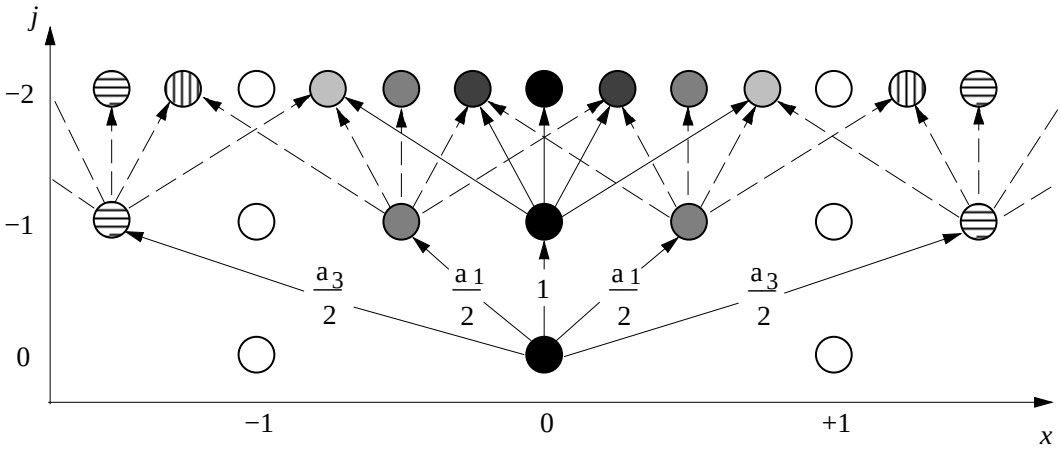


Figure 7.9: The Lagrange iterative interpolation of the unit impulse sequence with the associated quadrature mirror filter of length $L = 4$, i.e., $a_1 = 9/8$ and $a_3 = -1/8$. Black nodes at $x = 0$ indicate 1 and white nodes at $x = \pm 1$ have value 0. Shaded nodes have values other than 0 or 1. Note that the values of nodes existing at the jth scale do not change at the $(j-1)$st scale and higher. The result of repeating this procedure converges to $\Phi(x)$ as $j \to -\infty$.

efficient manner and with the prescribed accuracy.

Dubuc in [55] and Deslauriers and Dubuc in [47] considered the following problem: if B_n is the set of dyadic rationals $m/2^n, m = 0, 1, \ldots$, and the values of $f(x)$ are already given on the set B_0, then how may one extend f to $B_1, B_2, \ldots$ in an iterative manner so that the limit of this interpolation yields a uniformly continuous function?

Let $h = 1/2^{n+1}$ be a unit interval in the set B_{n+1}. For $x \in B_{n+1} \setminus B_n$, Dubuc have suggested the following formula to compute the value $f(x)$,

$$f(x) = \frac{9}{16}\left(f(x-h) + f(x+h)\right) - \frac{1}{16}\left(f(x-3h) + f(x+3h)\right). \tag{7.135}$$

Figure 7.9 illustrates a few steps of this iterative process applied to the unit impulse.

Deslauriers and Dubuc generalized this interpolation scheme as follows:

$$f(x) = \sum_{k\in\mathbb{Z}} F(k/2) f(x+kh), \quad \text{for} \quad x \in B_{n+1} \setminus B_n \text{ and } h = 1/2^{n+1}, \tag{7.136}$$

where the coefficients $F(k/2)$ are computed by generating the function satisfying

$$F(x/2) = \sum_{k\in\mathbb{Z}} F(k/2)F(x-k). \tag{7.137}$$

By comparing (7.136) and (7.137), we observe that the function $F(x)$ is an interpolation of the unit impulse $\{\delta_{0k}\}_{k\in\mathbb{Z}}$. Using this fact, the equation (7.136) may be rewritten as

$$f(x) = \sum_{k\in\mathbb{Z}} f(k)F(x-k) \qquad \text{for} \quad x \in \mathbb{R}. \tag{7.138}$$

In particular, they discussed an example connected with the Lagrange polynomial with $L = 2M$ nodes,

$$f(x) = \sum_{k=-M+1}^{M} \mathcal{P}_{2k-1}^{L-1}(0)\ f(x+(2k-1)h) \tag{7.139}$$

$$= \sum_{k=1}^{M} \mathcal{P}_{2k-1}^{L-1}(0)\ \left(f(x-(2k-1)h) + f(x+(2k-1)h)\right), \tag{7.140}$$

where $\{\mathcal{P}_{2k-1}^{L-1}(x)\}_{-M+1\le k\le M}$ is a set of the Lagrange polynomials of the degree $L-1$ with nodes $\{-L+1, -L+3, \ldots, L-3, L-1\}$,

$$\mathcal{P}_{2k-1}^{L-1}(x) = \prod_{l=-M+1, l\neq k}^{M} \frac{x-(2l-1)}{(2k-1)-(2l-1)}. \tag{7.141}$$

In this case, the equation (7.137) reduces to

$$F(x) = F(2x) + \sum_{k=1}^{M} \mathcal{P}_{2k-1}^{L}(0)\ \left(F(2x-2k+1) + F(2x+2k-1)\right). \tag{7.142}$$

This special case of (7.136) or (7.138) is called the "Lagrange iterative interpolation." The original Dubuc's scheme (7.135) corresponds to $L = 4$ in the scheme (7.139).

For general wavelets, we have the following proposition:

Proposition 7.7.

$$F(x) = \Phi(x), \tag{7.143}$$

where $F(x)$ is the fundamental function defined in (7.137) *and $\Phi(x)$ is the autocorrelation function of the scaling function $\varphi(x)$.*

Proof. Let us compute the quantity $\Phi(k/2)$ using the two-scale difference equation (7.59)

$$\Phi(k/2) = \Phi(k) + \frac{1}{2}\sum_{l\in\mathbb{N}} a_{2l-1}\left(\Phi(k-2l+1) + \Phi(k+2l-1)\right). \tag{7.144}$$

Using the property (7.47), we have from (7.144)

$$\Phi(k/2) = a_k/2. \tag{7.145}$$

In other words, the two-scale difference equation for the function Φ in (7.59) may be rewritten as

$$\Phi(x/2) = \sum_{k\in\mathbb{Z}} \Phi(k/2)\Phi(x-k). \tag{7.146}$$

Equivalence of (7.137) and (7.146) combined with the uniqueness of the nontrivial L^1-solution to these equations (Theorem 2.1 of [45]) implies (7.143). □

The vanishing moments of $\Phi(x)$ (see (7.62) and (7.63)) and Proposition 7.7 yield

Proposition 7.8 (Deslauriers & Dubuc [47]). *For any polynomial P of degree smaller than L, the Lagrange iterative interpolation of the sequence $f(n) = P(n)$, $n \in \mathbb{Z}$ via* (7.139)*, is precisely the function $f(x) = P(x)$ for any $x \in \mathbb{R}$.*

The regularity of the fundamental function $F(x)$ may also be derived from the results of Daubechies and Lagarias[46]. To compute the derivative of the interpolated function, we differentiate (7.138):

Proposition 7.9. *If $h = 2^{-n}$, and if $x \in B_m$, where $m \leq n$, then the derivative of an interpolation function $f(x)$ is computed via*

$$f'(x) = \sum_{k=1}^{L-2} r_k \left(f(x+kh) - f(x-kh)\right), \tag{7.147}$$

where

$$r_k = \int_{-\infty}^{+\infty} \varphi(x-k)\frac{d}{dx}\varphi(x)\,dx. \tag{7.148}$$

Note that the coefficients r_k coincide with those derived in [12] for the representation of the derivative operator in the orthonormal wavelet basis and may be computed using

Proposition 7.10 (Beylkin [12]). *1. If the integral in* (7.148) *exists, then the coefficients* $\{r_k\}_{k\in\mathbb{Z}}$ *in* (7.148) *satisfy the following system of linear algebraic equations*

$$r_k = 2\left[r_{2k} + \tfrac{1}{2}\sum_{l=1}^{L/2} a_{2l-1}(r_{2k-2l+1} + r_{2k+2l-1})\right], \tag{7.149}$$

and

$$\sum_{k\in\mathbb{Z}} k\, r_k = -1, \tag{7.150}$$

where the coefficients a_{2l-1} *are given in* (7.57).

2. If the number of vanishing moments of the wavelet $M \geq 2$, *then equations* (7.149) *and* (7.150) *have a unique solution with a finite number of non-zero* r_k, *namely,* $r_k \neq 0$ *for* $-L+2 \leq k \leq L-2$ *and*

$$r_k = -r_{-k}. \tag{7.151}$$

Let us describe an algorithm for evaluating functions $\Phi(x)$ and $\Phi'(x)$ at any given point $x \in \mathbb{R}$ for any given accuracy. This algorithm is used in the next section for both generating the zero-crossings representation of a signal and reconstructing the signal from that representation. We use the iterative interpolation scheme to zoom in the interval around x until we reach the interval $[x-\epsilon, x+\epsilon]$, where ϵ is the prescribed accuracy. Once in this interval, the derivative at the center of this interval is computed using (7.147). The evaluation of the functions $\Psi(x)$ and $\Psi'(x)$ may be obtained from the formula (7.65), i.e., $\Psi(x) = 2\Phi(2x) - \Phi(x)$.

Using this algorithm, we computed $\Phi(x)$ and $\Phi'(x)$ which are shown in Figure 7.10. The same Figure may be obtained if we directly apply the iterative interpolation scheme to the unit impulse (see Figure 7.9; see also Dubuc [55] and Daubechies & Lagarias [45, 46]).

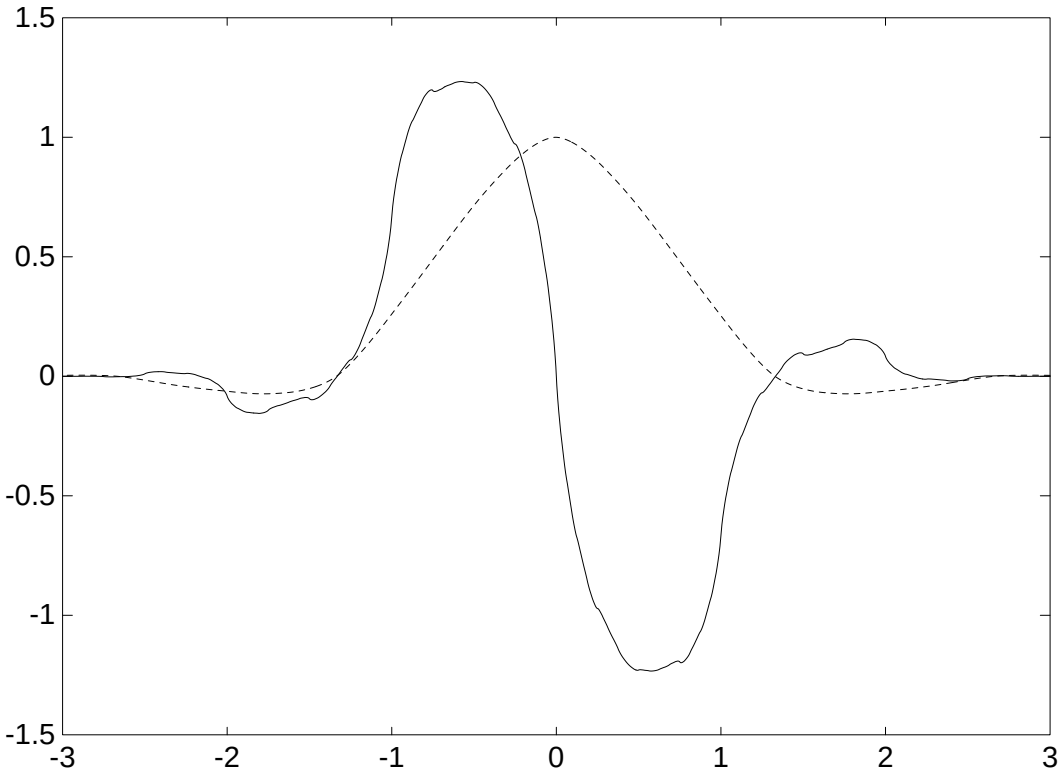


Figure 7.10: The autocorrelation function $\Phi(x)$ (dashed line) and its derivative $\Phi'(x)$ (solid line) with $L = 4$. Note the rough shape of $\Phi'(x)$.

Remark 7.11. The interpolation scheme discussed in this section "fills the gap" between the following two extreme cases:

If the number of vanishing moments $M = 1$ and the length of the quadrature mirror filter $L = 2$, then $\{\psi_{j,k}(x)\}$ is the Haar basis and we have

$$\Phi_{\text{Haar}}(x) = \begin{cases} 1 + x & \text{for } -1 \leq x \leq 0, \\ 1 - x & \text{for } 0 \leq x \leq 1, \\ 0 & \text{otherwise.} \end{cases} \tag{7.152}$$

The interpolation process is exactly *linear* interpolation. (The autocorrelation function of the characteristic function Φ_{Haar} is often called the *hat* function.)

Let us now consider the case where $M \to \infty$. Using the expressions (3.49)–(3.52) of [12], the relation (7.56) for the 2π periodic function $|m_0(\xi)|^2$ may be rewritten in terms of M as follows:

$$\begin{aligned} |m_0(\xi)|^2 &= \frac{1}{2} + \frac{1}{2}\sum_{k=1}^{M} a_{2k-1}\cos(2k-1)\xi \\ &= \frac{1}{2} + \frac{1}{2}C_M \sum_{k=1}^{M} \frac{(-1)^{k-1}\cos(2k-1)\xi}{(2k-1)\ (M-m)!\ (M+k-1)!}, \end{aligned} \tag{7.153}$$

where

$$C_M = \left[\frac{(2M-1)!}{(M-1)!\ 4^{M-1}}\right]^2. \tag{7.154}$$

If $M \to \infty$, then

$$|m_0(\xi)|^2 \to \frac{1}{2} + \frac{1}{2}\sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{2k-1}\cos(2k-1)\xi, \tag{7.155}$$

which is exactly the Fourier coefficient of the characteristic function $\chi_{[-\pi/2,\pi/2]}(\xi)$. This implies that the corresponding autocorrelation function is

$$\Phi_\infty(x) = \text{sinc}(x) = \frac{\sin \pi x}{\pi x}. \tag{7.156}$$

The interpolation process then corresponds to the so-called *band-limited* interpolation. Daubechies [44] noticed that if the number M of the vanishing moments of the compactly supported wavelet $\psi(x)$ approaches infinity, then the corresponding scaling function $\varphi(x)$ itself also approaches

$$\varphi_\infty(x) = \mathrm{sinc}(x). \tag{7.157}$$

As a result, we have the following relation,

$$\varphi_\infty(x) = \Phi_\infty(x), \tag{7.158}$$

and

$$\sqrt{2}h_k = \frac{a_k}{2} = \frac{\sin \pi k/2}{\pi k/2} \quad \text{for} \quad k \in \mathbb{Z}. \tag{7.159}$$

The autocorrelation function Φ of the wavelet corresponding to the quadrature mirror filter with M vanishing moments always satisfies the two-scale difference equation

$$\Phi(x/2) = \sum_{k=-M}^{M} \Phi(k/2)\Phi(x-k), \tag{7.160}$$

and (7.152) and (7.156) may be considered as the two extreme examples. Thus, the quadrature mirror filters with M vanishing moments, where $1 \leq M < \infty$, provide a parameterized family of the symmetric iterative interpolation schemes.

Remark 7.12. It turns out to be easy to adjust the auto-correlation shell to "life on the interval" (see [28] for a more delicate construction for wavelets on the interval). Since our filter coefficients p_k are obtained by evaluating the Lagrange polynomials at the origin $x = 0$ (see (7.139)), it is natural to adjust the filter coefficients for the boundaries by simply generating them by evaluating these polynomials at the desired points. For example, for the lowpass filter coefficients $2^{-1/2}\{-\frac{1}{16}, 0, \frac{9}{16}, 1, \frac{9}{16}, 0, -\frac{1}{16}\}$ based on Daubechies's QMF with $L = 2M = 4$, the adjusted lowpass filter coefficients for the left boundary are $2^{-1/2}\{\frac{5}{16}, 1, \frac{15}{16}, 0, -\frac{5}{16}, 0, \frac{1}{16}\}$. These coefficients are convolved with the leftmost 7 points of the signal to obtain the second leftmost point of the next scale.

7.5 On Reconstructing Signals from Zero-Crossings

In this section, we formulate the problem of reconstructing signals from zero-crossings (and slopes at these points) in the autocorrelation shell. The outline of our approach is as follows: we compute and record the zero-crossings (and slopes at these zero-crossings) on each scale of the autocorrelation shell expansion within the prescribed numerical accuracy using the Dubuc's iterative interpolation scheme. For reconstruction, we set up a system of linear algebraic equations, where the unknown vector is the original signal itself and the entries of the matrix are computed from the values of the autocorrelation function and its derivative at the integer translates of the zero-crossings. The signal is reconstructed by solving this linear system.

Reconstructing a signal from its zero-crossings by solving a linear system of equations has been proposed by S. Curtis and A. Oppenheim [41]. Their method requires a solution of a linear system where the unknowns are the Fourier coefficients and, therefore, the linear system is dense. It also requires multiple threshold-crossings rather than zero-crossings, and moreover, the quality of the reconstruction strongly depends on the choice of the thresholds. We would like to note that in our approach we take advantage of multiresolution properties of the autocorrelation shell and, specifically, of Proposition 7.5. This proposition allows us to set the linear system directly for the unknown signal rather than the coefficients of its expansion. We note that this proposition does not hold if we were to use the scale-space filtering with Gaussians, for example. If we were to use biorthogonal wavelet bases, then it is possible to set up a linear system similar to that constructed in this chapter. In this case the difficulty is in keeping both the difference and average coefficients sufficiently smooth and balance this requirement with the computational efficiency. In this chapter, however, we limit ourselves to considering only the autocorrelation shell.

7.5.1 Zero-crossing detection and computation of slopes

Using the iterative interpolation scheme described in the previous section, we locate the zero-crossing locations of the set of functions $\{\sum_{k=0}^{N-1} D_k^j \Phi(x-k)\}_{1\le j\le J}$ within the prescribed numerical accuracy (in our examples we compute in double precision with $\epsilon = 10^{-14}$). To compute the locations of zero-crossings, we recursively subdivide the unit interval bracketing the zero-crossing until the length of the subdivided interval bracketing that zero-crossing becomes less than the accuracy ϵ. The iterative interpolation scheme allows us to zoom in as much as we want around the zero-crossing. This process requires at most $O(-L\log_2\epsilon)$ operations per zero-crossing. Once the zero-crossing is found, the computation of the slope is merely the convolution of the $2(L-2)$ points around the zero-crossing with the filter coefficients $\{r_l\}_{-L+2\le l\le L-2}$ in (7.147).

7.5.2 An algorithm for reconstructing a signal from its zero-crossings representation

We address the following problem:

Given the coarsest subsampled average coefficients $\{S^J_{2^J k}\}_{0\le k\le 2^{n-J}-1}$, *and the zero-crossings and the slopes at these zero-crossings* $\{x_m^j, v_m^j\}_{1\le j\le J,\ 0\le m\le N_z^j-1}$, *where* N_z^j *is the number of zero-crossings of the function* $\sum_{k=0}^{N-1} D_k^j\Phi(x-k)$, *reconstruct the original vector* $\{s_k^0\}_{0\le k\le N-1}$.

Proposition 7.5 in Section 7.3 provides a simple mechanism for defining a linear system which relates the unknown signal $\{s_k^0\}$ and the values of the function Φ and its derivative at the integer translates of zero-crossings. Using the autocorrelation shell expansion of the signal, we immediately find the following relationships at all zero-crossings.

$$\sum_{k=0}^{N-1} D_k^j \Phi(x_m^j - k) = 0, \tag{7.161}$$

$$\sum_{k=0}^{N-1} D_k^j \Phi'(x_m^j - k) = v_m^j, \tag{7.162}$$

where $1 \le j \le J,\ 0 \le m \le N_z^j - 1$.

Applying Proposition 7.5 to (7.161) and (7.162), we have

$$\sum_{k=0}^{N-1} s_k^0 \widetilde{\Psi}_{j,k}(x_m^j) = 0, \tag{7.163}$$

$$\sum_{k=0}^{N-1} s_k^0\ 2^{-j} \widetilde{\Psi}'_{j,k}(x_m^j) = v_m^j. \tag{7.164}$$

Since it is easy to evaluate $\widetilde{\Psi}_{j,k}(x)$ and $\widetilde{\Psi}'_{j,k}(x)$ for any $x \in \mathbb{R}$ within the prescribed accuracy as described in the previous section, we interpret (7.163) and (7.164) as a system of linear algebraic equations where the original signal $\{s_k^0\}$ itself is the unknown vector.

Using the same proposition for the available average coefficients, we have

$$\sum_{k=0}^{N-1} s_k^0 \widetilde{\Phi}_{J,k}(x) = \sum_{k=0}^{N-1} S_k^J \Phi_{0,k}(x). \tag{7.165}$$

If we evaluate (7.165) at the integer point $x = 2^J l$, then we have

$$\sum_{k=0}^{N-1} s_k^0 \widetilde{\Phi}_{J,k}(2^J l) = S_{2^J l}^J, \tag{7.166}$$

for $l = 0, 1, \ldots, N_s - 1$, where $N_s = 2^{n-J}$.

We rewrite (7.163),(7.164), and (7.166) in a vector-matrix form as

$$\boldsymbol{A}\,\boldsymbol{s} = \boldsymbol{v}, \tag{7.167}$$

where $\boldsymbol{s} \in \mathbb{R}^N$ is a shorthand notation of the original signal $\{s_k^0\}$, $\boldsymbol{v} \in \mathbb{R}^{2N_z + N_s}$ is a data vector including the slopes and available coarsest subsampled coefficients, i.e.,

$$\boldsymbol{v} = (0, v_0^1, \ldots, 0, v_{N_z^1 - 1}^1, \ldots, 0, v_0^J, \ldots, 0, v_{N_z^J - 1}^J, S_0^J, S_{2^J}^J, \ldots, S_{N-2^J}^J)^T, \tag{7.168}$$

and $\boldsymbol{A}$ is a $(2N_z + N_s) \times N$ matrix and has the following structure:

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}^1 \\ \boldsymbol{A}^2 \\ \vdots \\ \boldsymbol{A}^J \\ \boldsymbol{S}^J \end{pmatrix}, \tag{7.169}$$

where $\boldsymbol{A}^j$ is a $2N_z^j \times N$ submatrix whose entries are

$$(\boldsymbol{A}^j)_{2k,l} = \widetilde{\Psi}_{j,l}(x_k^j), \tag{7.170}$$

$$(\boldsymbol{A}^j)_{2k+1,l} = 2^{-j}\widetilde{\Psi}'_{j,l}(x_k^j), \tag{7.171}$$

for $k = 0, \ldots, N_z^j - 1$ and $l = 0, \ldots, N - 1$ and

$\boldsymbol{S}^J$ is an $N_s \times N$ submatrix where

$$(\boldsymbol{S}^J)_{k,l} = \widetilde{\Phi}_{J,l}(2^J k), \tag{7.172}$$

for $k = 0, \ldots, N_s$ and $l = 0, \ldots, N-1$. (Note that we use vector and matrix indices starting from 0 rather than 1.) We note that the kth row of the matrix $\boldsymbol{S}^J$ comprises the average coefficients at the scale J of the autocorrelation shell expansion of the shifted unit impulse $\{\delta_{2^J k,l}\}_{0 \le l \le N-1}$.

Since the autocorrelation function $\widetilde{\Psi}_{j,k}(x)$ is compactly supported, the matrix $\boldsymbol{A}$ is sparse by construction. It is easy to check that the support of the function $\widetilde{\Psi}_{j,k}(x)$ is $2^{j+1}(L-1)$. Thus, the number $N_{\boldsymbol{A}}$ of non-zero entries of the matrix $\boldsymbol{A}$ is as follows:

$$N_{\boldsymbol{A}} = \sum_{j=1}^{J} 2N_z^j \; 2^{j+1}(L-1) + N_s \; 2^{J+1}(L-1) = 2(L-1)\left[\sum_{j=1}^{J} 2^j \; 2N_z^j + N\right]. \tag{7.173}$$

The number of zero-crossings usually decreases as the scale j increases. As a result, the number of the non-zero entries of the matrix $\boldsymbol{A}$ is essentially $O(N)$. The sparsity of this matrix allows one to solve the system (7.167) efficiently.

Whether we can solve the linear system (7.167) depends on the condition number of the matrix (7.169), which is affected by the distribution of locations of zero-crossings. If there are very few zero-crossings (which means that the signal is zero over a significant part of its support) as, for example, in the expansion of the unit impulse $\{s_k^0 = \delta_{k_0,k}\}$ with only $2L$ zero-crossings at each scale, then we need to use additional constraints for solving the linear system (7.167). There might be several approaches to introduce these additional constraints. One approach (which might be sufficient in some applications) would be to consider the generalized inverse of (7.169). Another possible approach (that we have experimented with), is to introduce a heuristic constraint that the distance between the adjacent zero-crossings at the jth scale does not exceed $2^{j+1}(L-1)$. To impose these constraints on the solution of the system of linear equations (7.167), we rewrite (7.167) in terms of the difference coefficients $\{D_k^j\}$. To do this, let $\boldsymbol{d} \in \mathbb{R}^{NJ+N_s}$ be a vector of the autocorrelation shell coefficients including the subsampled coarsest averages:

$$\boldsymbol{d} = (D_0^1, \ldots, D_{N-1}^1, \ldots, D_0^J, \ldots, D_{N-1}^J, S_0^J, S_{2^J}^J, \ldots, S_{N-2^J}^J)^T. \tag{7.174}$$

Also, let $\boldsymbol{T} \in \mathbb{R}^{(NJ+N_s)\times N}$ be a transformation matrix from $\boldsymbol{s}$ to $\boldsymbol{d}$:

$$\boldsymbol{d} = \boldsymbol{T}\,\boldsymbol{s}. \tag{7.175}$$

Then, the linear system (7.167) on $\boldsymbol{s}$ can be rewritten as

$$\boldsymbol{L}\,\boldsymbol{d} = \boldsymbol{v}, \tag{7.176}$$

where $\boldsymbol{L} = \boldsymbol{A}\,\boldsymbol{T}$ is a $(2N_z + N_s) \times (NJ + N_s)$ matrix,

$$\begin{pmatrix} \boldsymbol{L}^1 & \boldsymbol{0} & \cdots & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{L}^2 & \cdots & \cdots & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & \cdots & \cdots & \boldsymbol{L}^J & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \cdots & \boldsymbol{0} & \boldsymbol{I}_{N_s} \end{pmatrix}, \tag{7.177}$$

where $\boldsymbol{I}_{N_s}$ is the N_s-dimensional identity matrix, and $\boldsymbol{L}^j$ is a $2N_z^j \times N$ block matrix whose entries are

$$L^j_{2k,l} = \Phi(x^j_k - l), \tag{7.178}$$

$$L^j_{2k+1,l} = \Phi'(x^j_k - l), \tag{7.179}$$

for $k = 0, \ldots, N_z^j - 1$ and $l = 0, \ldots, N - 1$. We now impose a constraint that if there are $2^{j+1}(L-1)$ or more consecutive null columns in the submatrix $\boldsymbol{L}^j$, then the corresponding coefficients must be zero. These constraints may be written as

$$\boldsymbol{C}\,\boldsymbol{d} = \boldsymbol{0}, \tag{7.180}$$

and $\boldsymbol{C}$ is an $(NJ + N_s)$-dimensional square matrix of the form

$$\begin{pmatrix} \boldsymbol{C}^1 & \boldsymbol{0} & \cdots & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}^2 & \cdots & \cdots & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & \cdots & \cdots & \boldsymbol{C}^J & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \cdots & \boldsymbol{0} & \boldsymbol{0} \end{pmatrix}, \tag{7.181}$$

where the submatrix $\boldsymbol{C}^j$ is an N-dimensional diagonal matrix as

$$(\boldsymbol{C}^j)_{k,k} = \begin{cases} 1 & \text{if } D^j_k \text{ must be zero,} \\ 0 & \text{otherwise.} \end{cases} \tag{7.182}$$

To eliminate $\boldsymbol{d}$ in favor of $\boldsymbol{s}$ in equation (7.180), we use the transformation matrix $\boldsymbol{T}$, and define the matrix $\boldsymbol{B} = \boldsymbol{C}\,\boldsymbol{T}$ where $\boldsymbol{B} \in \mathbb{R}^{(NJ+N_s)\times N}$. Then, (7.180) can be written as $\boldsymbol{B}\,\boldsymbol{s} = \boldsymbol{0}$. Hence the problem may now be stated as follows:

$$\textit{Minimize} \quad \|\boldsymbol{A}\,\boldsymbol{s} - \boldsymbol{v}\| \quad \textit{subject to} \quad \boldsymbol{B}\,\boldsymbol{s} = \boldsymbol{0}. \tag{7.183}$$

Using the method of Lagrange multipliers, we obtain the least squares solution

$$\hat{\boldsymbol{s}} = (\boldsymbol{A}^T\boldsymbol{A} + \lambda \boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{A}^T\boldsymbol{v}. \tag{7.184}$$

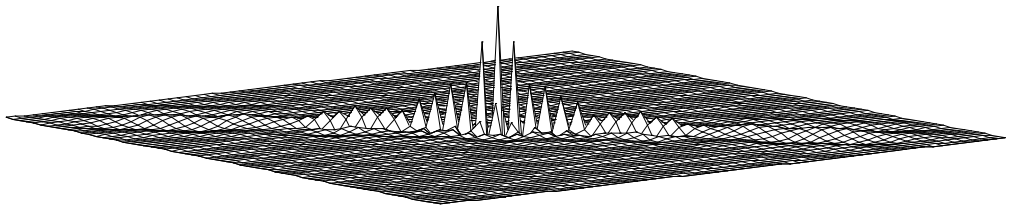
Since we consider minimizing $\|\boldsymbol{A}\,\boldsymbol{s} - \boldsymbol{v}\|$ and satisfying $\boldsymbol{B}\,\boldsymbol{s} = \boldsymbol{0}$ equally important, we assume $\lambda = 1$. We note that our formulation is completely linear except for the process of the zero-crossing detection. It is clear from (7.167) and (7.183), that the slope information is essential for signal reconstruction since if there is no slope information, we have only the trivial solution, $\boldsymbol{s} = \boldsymbol{0}$. Previously, this fact was examined only empirically [78].

We also note that our approach may be modified to produce the maxima-based representation of Mallat and Zhong [98] by considering $\int_{-\infty}^{x} \Psi(y)\,dy$ instead of $\Psi(x)$ and the corresponding two-scale difference equation. Using the symmetric iterative interpolation, we have better numerical control than by using the approaches developed by Mallat and Zhong and by Hummel and Moniot [78]. Recently, however, Berman and Baras showed in [11] that the dyadic wavelet maxima representation of Mallat and Zhong and the dyadic wavelet zero-crossings representation of Mallat [95] are, in general, nonunique. In our future project, we will study the applicability of their results to our zero-crossings representation scheme.

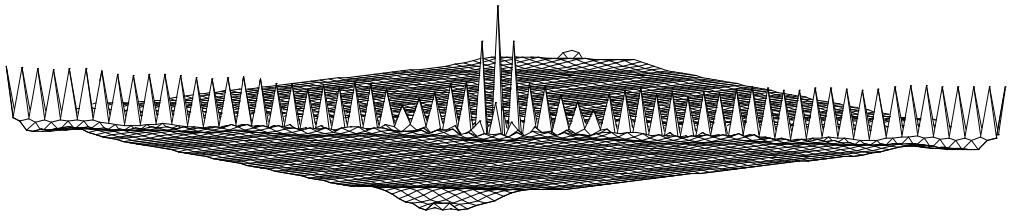
7.5.3 Examples

We first use the signal shown in the Figure 7.6 in Section 7.3 as an example. In this case, the size of the matrix $\boldsymbol{A}$ is 1852 by 512. The relative ℓ^2 error of the reconstructed signal compared with the original signal (see Figure 7.6) is 5.674436×10^{-13}. The accuracy threshold ϵ was set to 10^{-14} in this case.

Next, let us consider the reconstruction of the unit impulse $\{\delta_{k_0,k}\}$ from its zero-crossings and slopes in the autocorrelation shell expansion. In this case, the size of the matrix $\boldsymbol{A}$ is 56×64. In Figure 7.11a we display the matrix $\boldsymbol{A}^T\boldsymbol{A}$ in (7.184). In Figure 7.11b we show the matrix with the constraints $\boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{B}^T\boldsymbol{B}$. It is easy to see that the constraints



(a)



(b)

Figure 7.11: The effect of the constraints in the reconstruction of the unit impulse from zero-crossing and slopes. (a) The unconstrained matrix $\boldsymbol{A}^T\boldsymbol{A}$. (b) The constrained matrix $\boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{B}^T\boldsymbol{B}$.

serve to condition the linear system. The relative ℓ^2 error with the constraints is 7.417360×10^{-15} whereas the error of the solution by the generalized inverse without the constraints is 3.247662×10^{-4}.

7.6 Summary

In this chapter we have proposed the autocorrelation shell representation, a "hybrid" shift-invariant multiresolution representation using the autocorrelation functions of compactly

supported wavelets. The autocorrelation functions of the corresponding scaling functions induce the symmetric iterative interpolation of Dubuc [55] and Deslauriers and Dubuc [47] which allows us to interpolate efficiently on all dyadic rationals. This property of the autocorrelation functions enables us to compute zero-crossings and slopes at the zero-crossings of the autocorrelation shell representation. This representation also gives us an explicit relation between the original signal and its expansion coefficients so that we can set up a system of linear algebraic equations for reconstructing the original signal from these zero-crossings and slopes. The original signal is reconstructed within prescribed numerical accuracy by solving this linear system.

Chapter 8

Further Development

8.1 Introduction

So far we have used a library of orthonormal bases and a library of non-orthogonal bases generated by the autocorrelation functions of wavelets for the various signal analysis tasks. In the time-frequency plane, we can associate each basis function in our libraries with a rectangular box called a "Heisenberg box" or a "phase cell" [105], [157]. In particular, an orthonormal basis in the library corresponds to a disjoint cover (or tiling) of the time-frequency plane by these rectangular boxes (see e.g., [157], [69], [5] for pictures of various tiling patterns). This is because our basis functions are generated by applying certain translation operators both in time and frequency axes as well as dilation operators to a set of elementary functions such as the scaling functions for wavelets/wavelet packets and "bell" functions for local trigonometric bases. Thus, the tiling of the time-frequency plane by a particular basis in our library is completely governed by the Heisenberg box of the corresponding elementary function as well as the translation and dilation operators. For a certain class of signals which we have seen so far, these basis functions have worked quite well; however, for other types of signals such as frequency-modulated signals (often called

chirp signals), our basis functions may not be too efficient. In other words, energy of these signals may not be captured efficiently by our basic analysis tools which are constrained in rectangular boxes whose edges are parallel to the time and frequency axes; the entropy of the signal represented in such a basis may be large.

Recently, Coifman, Matviyenko, and Meyer proposed one approach toward breaking this limitation: tiling the time-frequency plane by a set of "oblique" boxes which consist of *linear chirps* (which have a modulation factor of the form $e^{i\omega t(at+b)}$) localized by smooth cut-off functions [29]. Baraniuk and Jones made a similar observation [5] and for computing actual expansion coefficients onto such a chirp basis, they suggested to "warp" the time axis of a signal with the predefined unitary operator followed by the standard wavelet expansion so that the standard software can be used.

We can certainly add these new bases in our library; in this chapter, however, to extend the usefulness of our standard library, we propose another exploratory approach using the entropy minimization principle. Unlike the approach of Baraniuk and Jones which uses the predefined warping functions, our approach exploratively finds a time-warping (or deformation or distortion) function which monotonically increasing nonlinear function of time. This function makes the observed signal complicated and makes our analysis tools inefficient. By undoing the warping effect (or "unwarping" the signal), we can obtain a simpler version of the signal suitable for the analysis and interpretation using our library of bases. This idea is also related to the minimum description length principle. We can think of the shorter description of the observed signal as two-part codes, i.e., the sum of the codelengths of: 1) the unwarped (simplified) signal and 2) the warping function itself. We can obtain a particularly compact representation of a signal if its warping function is a smooth function such as a polynomial. For simple chirp signals, our method "discovers" the modulation laws of such signals. Finding the modulation law of chirp signals essentially amounts to "straightening" their energy distributions in the time-frequency plane. This

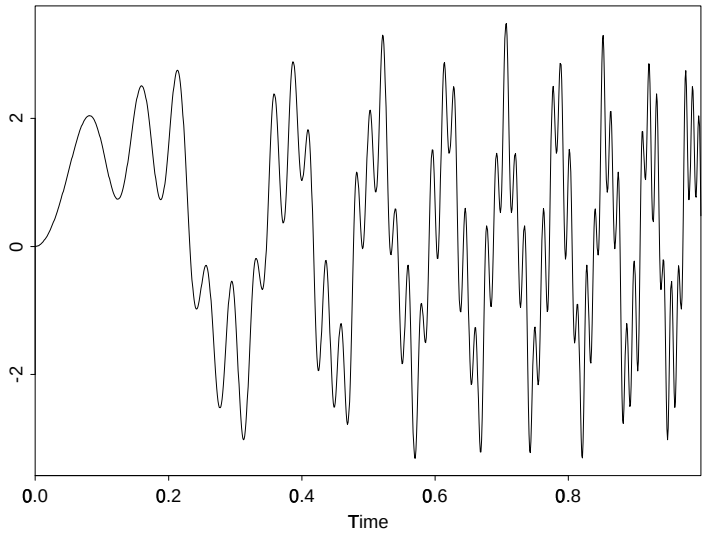


Figure 8.1: An example of chirp signal.

gives rise to a simpler representation of the signal. This idea is best explained by figures. In Figure 8.1, we display an example chirp signal specified by

$$x(t) = \sin(97\pi u(t)) + 2\sin(17\pi u(t)) + 0.5\sin(53\pi u(t)), \quad \text{for } 0 \le t \le 1, \tag{8.1}$$

where $u(t) = t^2$, and t is sampled 1024 times. The corresponding energy distribution in the time-frequency plane using the local sine best basis of this signal is shown in Figure 8.2. We can see that dominant energies increase linearly in this plane. The normalized entropy of this signal is 4.6909 bits in the local sine best basis. Figure 8.3 shows our result: the unwarped or "demodulated" version of the above chirp signal. All linear patterns in Figure 8.2 are straightened now in Figure 8.4. The normalized entropy of this unwarped signal in its own local sine best basis is just 2.7537 bits. The warping information costs 0.4576 bit with its own local sine best basis. The sum is 3.2113 bits which is smaller than the original entropy[1].

[1] In the strict MDL formulation, we need additional cost for describing two different best bases.

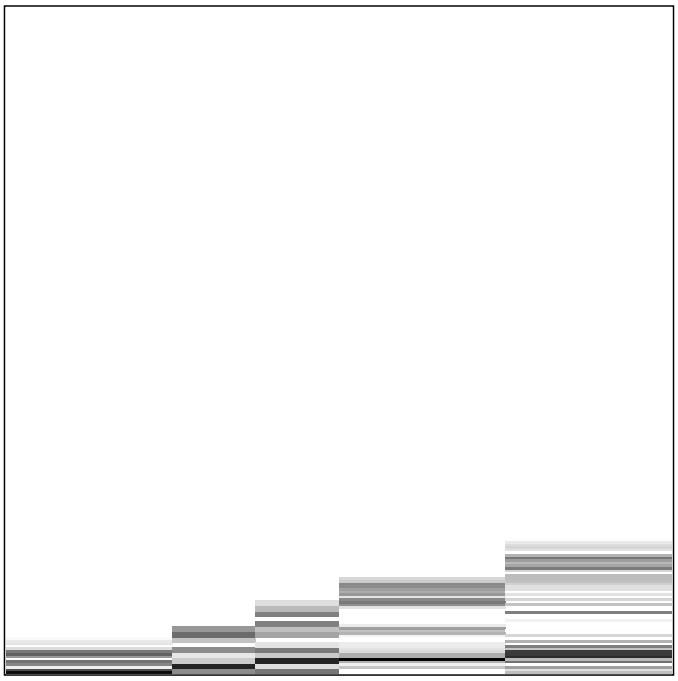


Figure 8.2: The time-frequency energy distribution of the chirp signal shown in Figure 8.1 in the local sine best basis coordinate. The horizontal and vertical axes represent time (increasing from left to right) and frequency (increasing from bottom to top), respectively. Gray levels in the image is proportional to the logarithm of squares of the expansion coefficients.

8.2 Discovering Time-Warping Functions by the Entropy Minimization Principle

In this section, we describe an algorithm for finding such a time-warping function. We do not take a strict MDL approach at this point: we do not minimize the sum of the description lengths of the unwarped signal and the warping function. Instead, we focus our attention on the first term, i.e., the simplicity of the unwarped signal, by the entropy minimization principle. We will study the strict MDL-based formulation in our future project.

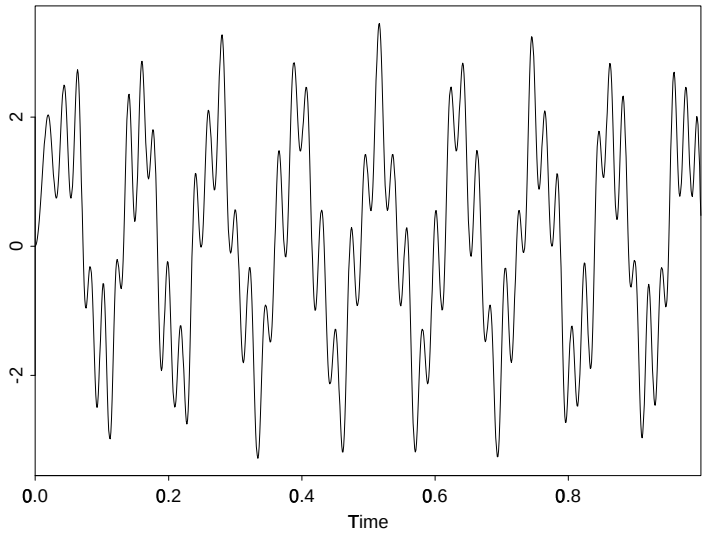


Figure 8.3: The chirp signal of Figure 8.1 after the "demodulation"

8.2.1 Problem formulation

Let $\boldsymbol{x} = (x_0, \ldots, x_{N-1}) \in \mathbb{R}^N$ be a signal vector at hand. By unwarping, the samples on the regular grids in original domain are transformed to those on the irregular grids. To compute the samples on the regular grids after unwarping, we construct a continuous function $x(t)$ of $t \in [0, N-1]$ from the original signal samples $\{x_k\}$. To do this, we use the autocorrelation function of scaling function $\Phi(t)$ defined in (7.45):

$$x(t) = \sum_{k=0}^{N-1} x_k \Phi(t-k). \tag{8.2}$$

We simply use (8.2) to obtain the interpolated values at any t with $0 \leq t \leq N-1$. Let us assume that a *warping function* $t = v(\tau)$ is a monotonically increasing function of $\tau \in [0, N-1]$ with $v(0) = 0$ and $v(N-1) = N-1$. Let $\mathcal{U} \subset C^1[0, N-1]$ denote a space of such functions. We also assume its inverse *unwarping function* $\tau = v^{-1}(t)$ exists and $v^{-1} \in \mathcal{U}$. Let u denote this inverse v^{-1} for short. Both time warping and unwarping are simply a change of variable of the function x. We want to find a nonlinear mapping u^*

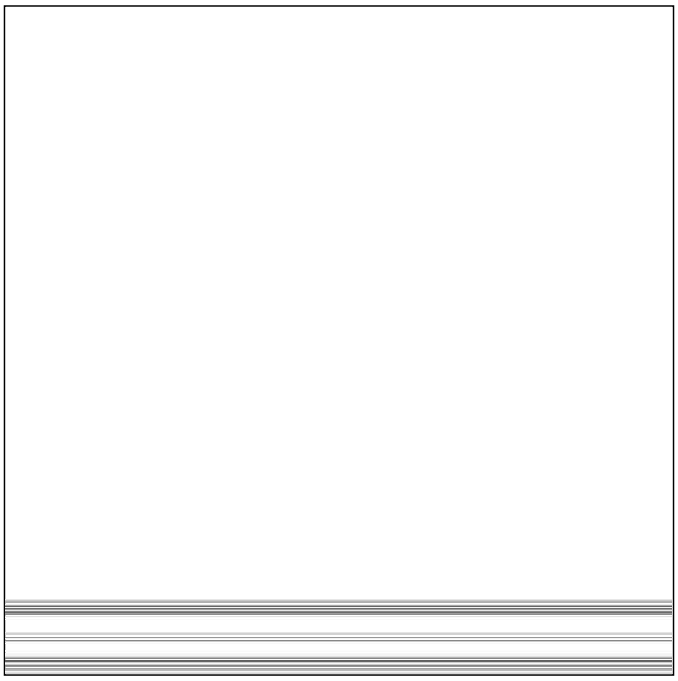


Figure 8.4: The time-frequency energy distribution of the chirp signal after "demodulation" in its own local sine best basis coordinate.

from t-domain to τ-domain so that changing the variable from t to $\tau = u^*(t)$ minimizes the entropy of the signal.

To do this, we need to define the entropy of the unwarped signal. The samples on the regular grids $\{0, 1, \ldots, N-1\}$ in τ-domain correspond to the irregular time positions $\{v(0), v(1), \ldots, v(N-1)\}$ in the original t-domain. Let $\boldsymbol{x}(v) = (x(v(0)), x(v(1)), \ldots, x(v(N-1)))$. This vector $\boldsymbol{x}(v)$ represents the unwarped signal regularly sampled in τ-domain. Now we can define the entropy of the unwarped signal $\boldsymbol{x}(v)$ as the normalized entropy of the sequence $\boldsymbol{x}(v)$ with respect to its own best basis selected from our library $\mathfrak{L}$:

$$H_{\mathfrak{L}}(\boldsymbol{x}(v)) = \min_{\boldsymbol{W} \in \mathfrak{L}} H_2(\boldsymbol{W}^T \boldsymbol{x}(v)), \tag{8.3}$$

where $H_2(\cdot)$ is the normalized entropy of a sequence defined as (2.2) in Chapter 2:

$$H_2(\boldsymbol{y}) = -\sum_k \left(\frac{y_k}{\|\boldsymbol{y}\|}\right)^2 \log \left(\frac{y_k}{\|\boldsymbol{y}\|}\right)^2,$$

and $\boldsymbol{W} \in \mathrm{O}(N)$ denotes a matrix corresponding to a basis in our library. For our problem, the normalization is critical since the change of variable does not conserve the signal's energy.

Our problem is now described as follows: *find a function* $v^*(\tau)$ *such that*

$$H_{\mathfrak{L}}\left(\boldsymbol{x}(v^*)\right) = \min_{v \in \mathcal{U}} H_{\mathfrak{L}}\left(\boldsymbol{x}(v)\right). \tag{8.4}$$

8.2.2 Numerical implementation

Since this is a nonlinear problem, it is difficult in general to find the v^* giving the global minimum of (8.4). Moreover, if we do not restrict the space of solutions $\mathcal{U}$, it may become computationally infeasible. Therefore, we use a one-parameter family of model functions to approximate $\mathcal{U}$ since this reduces the problem (8.4) to a one-dimensional nonlinear optimization problem. In particular, we use a family of hyperbolas mainly because: 1) it generates a set of smoothly varying functions covering the warping range $[0, N-1] \times [0, N-1]$ on the Cartesian product of t and τ spaces, and 2) it allows explicit algebraic expressions for both v and u which save computational cost. Let ε denote this parameter. Let us write $v_\varepsilon^{-1}(t) = u_\varepsilon(t) = t + h_\varepsilon(t)$, where $h_\varepsilon(t)$ is a hyperbola from this family:

$$h_\varepsilon(t) = \sqrt{a_\varepsilon^2 + b^2} - \sqrt{a_\varepsilon^2 + (t-b)^2}, \quad \text{for } 0 \le t \le N-1, \tag{8.5}$$

where $b = (N-1)/2$, i.e., the midpoint of the interval $[0, N-1]$, and $a_\varepsilon = -b(\varepsilon - 1/\varepsilon)/2$. This hyperbola takes its maximum or minimum $b\varepsilon$ at the midpoint $t = b$ depending on the sign of ε. Also, we have $h_\varepsilon(0) = h_\varepsilon(N-1) = 0$, and $h_\varepsilon(t) \equiv 0$ for $\varepsilon = 0$. After some algebraic manipulations, its inverse $v_\varepsilon(\tau)$ can be written as

$$v_\varepsilon(\tau) = \tau \frac{(N-1)(1+\varepsilon^2) - 2\varepsilon\tau}{(N-1)(1+\varepsilon^2) + 2\varepsilon(n-1-2\tau)}. \tag{8.6}$$

This formula is necessary to evaluate the entropy of the unwarped signal $\boldsymbol{x}(v_\varepsilon)$.

In the first step, what we are after is ε^* minimizing the entropy $H_{\mathfrak{L}}(\boldsymbol{x}(v_{\varepsilon^*}))$. Any standard method can be used for this one-dimensional nonlinear optimization problem;

in general, however, there is no guarantee for the method to find the global minimum. Even after obtaining the optimal v_{ε^*}, we may still face a problem: the globally-minimizing solution v^* to (8.4) may not be well-approximated by the single parametric function v_{ε^*}. Therefore, we use an iterative approach (successive approximation): replace the original signal $\boldsymbol{x}$ by the unwarped signal $\boldsymbol{x}(v^{\varepsilon^*})$ obtained by the single optimization process and iterate the whole process on this new $\boldsymbol{x}$.

For the example shown in the last section, we used the autocorrelation of the Haar function (i.e., the hat function) in (8.2) since it permit us to use the linear interpolation which has the computational advantage to the other interpolation schemes. We also used only a local sine dictionary as the library member. In Step 2, we adopted Brent's method [116, pp.402–405]. The unwarped signal shown in Figure 8.3 was obtained after three iterations of the successive approximation process mentioned above.

We show another example in Figure 8.5. The original signal was computed by the same formula (8.1) with the arctangent unwarping function,

$$u(t) = \frac{1}{2}\left[\tan^{-1}\left(\tan(1)\cdot(2t-1)\right)+1\right], \quad \text{for } 0 \le t \le 1.$$

After five iterations of the above algorithm, it reached to a minimum entropy solution (this may still be a local minimum). Figure 8.6 shows how the unwarping function obtained at each iteration is successively approaching to the true unwarping function.

8.3 Discussion

The above algorithm is designed for signals warped by smooth global functions. For a signal deformed locally, i.e., a signal whose time-frequency energy distribution has more complicated local structures, the hierarchical algorithm similar to the best-basis algorithm should be used:

Step 0: Segment a signal into a binary tree of dyadic blocks using the smooth cut-off

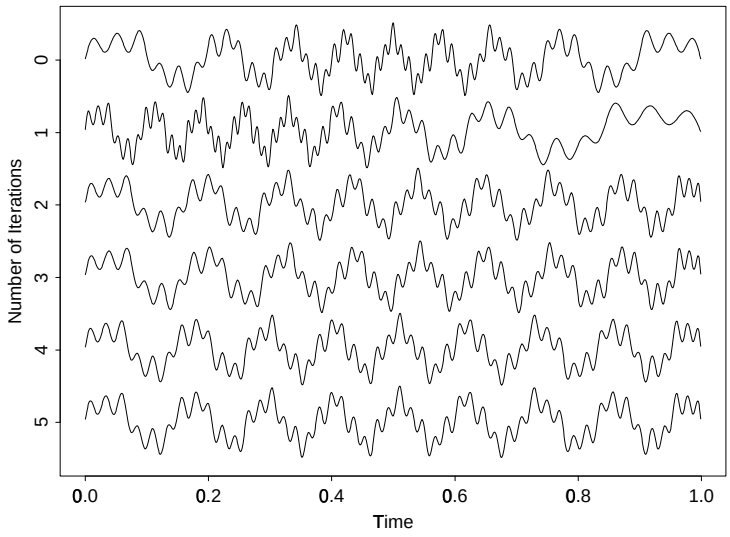


Figure 8.5: Unwarping a signal warped by a tangent function. The original signal is shown in the top row.

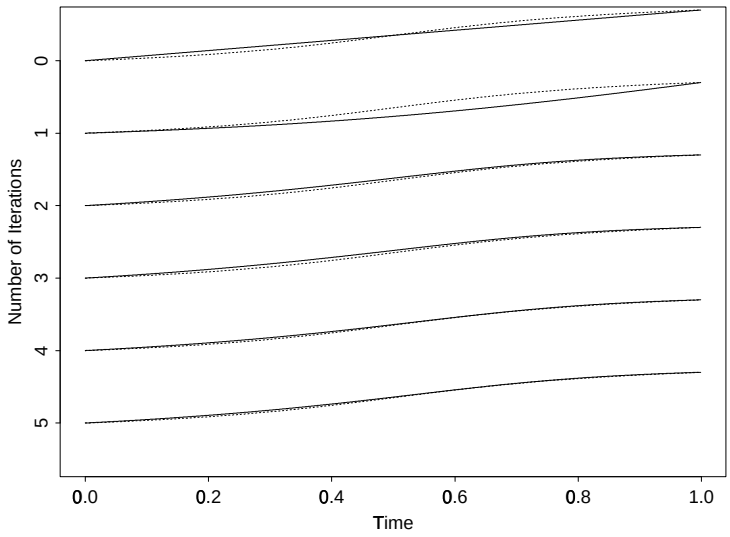


Figure 8.6: Discovering the modulation law of the signal shown in Figure 8.5. Dotted lines indicate the true unwarping function. Solid lines indicate the approximation to the true unwarping function obtained at each iteration.

functions similarly to the local trigonometric transforms.

Step 1: At each node, compute the best unwarping function and the minimum entropy value.

Step 2: Prune the binary tree: at each parent node of the binary tree, eliminate its two children nodes if the sum of the minimum entropy by unwarping the signals in two child nodes separately is greater than or equal to the minimum entropy of unwarped signal in the parent node.

Step 3: Construct the unwarping function by combining unwarping functions at all the survived nodes.

We will implement the multiple window unwarping algorithm with careful boundary treatment in our future project.

Straightening the time-frequency energy distribution is an important issue not only for the compact representations but also for understanding the pattern/signal generating mechanisms. This problem is also related to the nonuniform sampling of signals [25], [159]. In speech recognition, the concept of time-warping is used for matching distorted spoken words to the standard templates [136]. For images, deformation or domain warping has important applications in computer vision [108], [64], medical image analysis [8], and structural geology [70]. We will continue our research in this important area.

Chapter 9

Conclusion

In this thesis, we have explored many problems related to feature extraction using a *library of bases*. Our philosophy, the *best-basis paradigm*, is to select the best possible basis for the problem at hand from a library of bases. Wavelet packet dictionaries (including wavelet bases) and local trigonometric dictionaries are the main constituents of our library. In Chapters 4, 5, 6, we have seen that this array of tools provides us with better understanding and insight on the data-generating mechanisms or underlying physical phenomena than the conventional methods. Moreover, we have shown that these tools provide efficient and concrete numerical algorithms for the problem at hand.

We believe that these tools *unify* or extend various conventional methodologies or concepts. Table 9.1 summarizes the correspondences discussed or obtained in this thesis. We have also indicated that the best-basis paradigm provides a bridge between the statistical and syntactic approaches to pattern recognition problems; the best-basis paradigm allows one to represent a class of signals by a basis selected from a set of tree-structured bases and one can interpret its tree structure as a grammar or a rule specifying that class. The best-basis paradigm may also be viewed as a unifying tool in the area of computational and descriptive complexities. Tom Cover says in his stimulating book [40, page 3]:

Object	Conventional Concept	"Library" Concept
Coordinate System	Standard Euclidean Basis Fourier Basis	Wavelet Packet Bases Local Trigonometric Bases
Interpolation Scheme	Linear Interpolation Band-Limited Interpolation	Symmetric Iterative Interpolation
Edge Operator	Difference of Boxcars Difference of Gaussians	Difference of Autocorrelation Functions of Wavelets
Compression	Karhunen-Loève Basis	Joint Best Basis
Classification	Linear Discriminant Analysis	Local Discriminant Basis
Regression	Linear Regression	Local Regression Basis

Table 9.1: Summary of the correspondences between the conventional concepts and the new concepts based on the "best-basis paradigm/a library of bases" reviewed, discussed, or developed in this thesis.

> "One can think about computational complexity (time complexity) and Kolmogorov complexity (program length or descriptive complexity) as two axes corresponding to program running time and program length. Kolmogorov complexity focuses on minimizing along the second axis, and computational complexity focuses on minimizing along the first axis. Little work has been done on the simultaneous minimization of the two."

We believe that the best-basis paradigm approaches a simultaneously-minimizing solution in this plane.

Many future projects have been mentioned in the course of this thesis. This thesis concludes with a list of most important problem areas which could not be treated here and will be studied in the future:

1) *Robustness:* In a real dataset, the existence of outliers is not uncommon. Therefore, robust estimation techniques [76] should be considered. These methods use median, median absolute deviation, and errors measured in ℓ^1 norm, instead of mean, standard deviation, ℓ^2-based errors which have been used in this thesis. To make the results of this thesis more robust, it is necessary to incorporate these robust methodologies in the basis selection

mechanisms and error estimates. Another robustness problem is the shift sensitivity of the expansion coefficients onto our orthonormal bases in the library. One possible solution is to create a few circularly-shifted versions of the original signals—shifts either in time or in frequency or in both—as discussed in Chapter 3. This reduces the sensitivity of the coefficients to the shifts and at the same time increases the number of training samples for classification and regression problems.

2) *Invariance:* Many real data, such as hand written characters, speech signals, or subsurface formation patterns, are deformed or distorted from their standard (or normal) templates and patterns. Classifying such dataset requires splitting the input space into the equivalent classes, each of which contains all kinds of distorted patterns generated from a single template. Simpler distortions include translation, dilation, and rotation operators and their combinations. Invariance in pattern recognition was studied by many researchers, in particular, Amari [2], Grenander [64], Kanatani [79], Lenz [89], Otsu [112], and Segman et al. [137]. Research to deal with general nonlinear distortion functions is still in its infancy. It is necessary to extend the ideas proposed in Chapter 8 and make them robust.

3) *How to combine the dictionaries:* The current best-basis paradigm first selects the best basis (JBB/LDB/LRB) for the problem at hand from each dictionary *individually* in the library. Then, it selects the best of these best bases in the library. For signals having composite features, such as the signals containing spikes and sinusoids in Figure 4.6, the best-basis paradigm iteratively "peels off" the features: it first extracts the major features by some best basis, then the residual signals are supplied to the algorithm to get the secondary features, and it repeats the process. A more efficient (in the sense of descriptive complexity) way to handle this problem is to describe such a signal as a linear combination of the basis functions from different dictionaries; computationally, however, it may be very expensive; see also the related proposal called "matching pursuit" by Mallat and Zhang [97].

4) *Higher dimensional signals:* All the approaches proposed in this thesis can be extended

to images and multidimensional signals; however, except for the denoising applications, we have not applied our methods to real images. Extending the library for images, e.g., adding the non-separable two-dimensional wavelets, should be implemented. Also, research on deformations and distortions of two-dimensional features is quite challenging but may provide us with fascinating new ideas.

Appendix A

MDL-Based Tree Pruning Algorithms

A.1 Introduction

In Chapters 4,5, and 6, we have applied pruning algorithms to eliminate unimportant branches of the fully-grown trees. Pruning trees are important to avoid "overtraining" [18]. The larger or more complex a tree becomes, the better the performance on the training dataset that was used to generate that tree; however, in general, the performance on the test dataset gets worse since it learns too many specific features of the training dataset and loses its generalization power. On the other hand, if a tree is too small, then it may not capture some important information in the training dataset for the problem at hand. An important question is how to obtain the "right-sized" tree. This situation is similar to determining the number of basis coefficients to retain in the problem of simultaneous noise suppression and signal compression in Chapter 3. In this appendix, we consider two pruning algorithms and compare their performances.

A.2 Minimal Cost-Complexity Pruning

Breiman et al. suggested the so-called "minimal cost-complexity pruning" method [18] (see also [26]). To explain their algorithm, we need some terminology. Let $T_{\max}$ denote a fully-grown tree where each node are either "sparse" (e.g., it only contains less than 10 samples) or "pure" (e.g., the node deviance becomes less than 1% of the root node). Let t_k denote a node in the tree, and in particular, let t_0 denote the root node. A branch (or subtree) of a tree T consisting of the node t and all descendants of t is denoted by T_t and we write this relationship as $T \succ T_t$. Let $\widetilde{T}$ denote a set of terminal nodes in T, and $|\widetilde{T}|$ denote the number of terminal nodes in T and we call this number the "size" of the tree T. Then the cost complexity of T is defined as

$$D_\alpha(T) \stackrel{\Delta}{=} D(T) + \alpha|\widetilde{T}|, \tag{A.1}$$

where $D(T)$ is the deviance of the tree T, i.e., the entropy for CTs, the residual sum of squares (RSS) for RTs. The parameter $\alpha \geq 0$ is called the complexity parameter. We can easily see the second term is the regularization term and is closely related to Barron's complexity regularization technique [6] (see also Section 3.6). The minimal complexity (sub)tree $T(\alpha)$ for a fixed α is defined by

$$D_\alpha(T(\alpha)) \stackrel{\Delta}{=} \min_{T \preceq T_{\max}} D_\alpha(T), \tag{A.2}$$

and

$$\text{If} \quad D_\alpha(T) = D_\alpha(T(\alpha)), \quad \text{then} \quad T(\alpha) \preceq T. \tag{A.3}$$

This definition selects the smallest minimizer of D_α if there are ties. Breiman et al. showed that for every $\alpha \geq 0$, there exists a smallest minimizing subtree as defined by (A.2) and (A.3). Moreover, using the finiteness of the number of subtrees of $T_{\max}$, Breiman et al. showed how to obtain the sequence of subtrees

$$T_{\max} = T_1 \succ T_2 \succ \cdots \succ \{t_0\},$$

where $T_k = T(\alpha_k) = T(\alpha)$ for $\alpha_k \leq \alpha < \alpha_{k+1}$ with $\alpha_1 = 0$.

Thus, the remaining important question is how to find an optimal α and equivalently, an optimal subtree from this sequence. If we use the resubstitution estimates of D (i.e., using the training dataset on which the full tree was grown), the values of D become too optimistic. To overcome this problem, Breiman et al. suggested using the so-called "cross-validation" technique. Given a sequence of the subtree sizes, $|\widetilde{T}_{\max}| = |\widetilde{T}_1| > \ldots > |\{t_0\}| = 1$, this procedure first randomly divides the training dataset into mutually exclusive M subsets (say, $M = 10$) and uses $M - 1$ subsets for constructing the sequence of subtrees of given sizes. Then, the remaining subset is used to evaluate the sequence: this subset is used as an independent test dataset and the deviance of each subtree is computed. This process is iterated for M times, i.e., each subset is used for a test dataset and the deviances are accumulated and then averaged. Figure A.1 compares deviances computed by the resubstitution and the cross-validation. The tree was grown on the dataset of Example 4.6 represented in the LRBP coordinates. The full tree has been already displayed in Figure 5.1. If we use the resubstitution estimates, we get the full tree as the minimal complexity tree; on the other hand, a subtree with 8 terminal nodes is chosen using the cross-validation estimates. The major problems of this cross-validation approach are: 1) how to determine the parameter M, i.e., into how many subsets we should divide the training dataset (why not $M = 2$? why not the extreme case, $M = N$?), and 2) the computational burden of the repeated tree-growing and pruning processes. Because of these problem, we prefer to use the MDL-based pruning algorithm which we discuss in the next section.

A.3 MDL-Based Pruning Algorithms

Use of the MDL principle for tree pruning have been considered by several researchers, such as Rissanen [128, Section 7.2], Quinlan and Rivest [117], Wallace and Patrick [148]. In this section, we modify the algorithm proposed by Rissanen specifically for CART. An

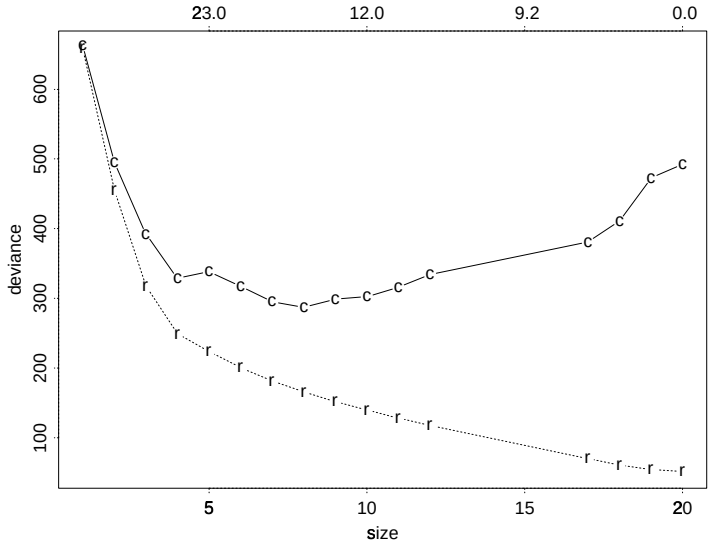


Figure A.1: A comparison of curves of subtree size versus deviance using the resubstitution estimates (dotted line) and the cross-validation estimates (solid line). The dataset of Example 4.6 represented in the LRBP coordinates is used. The algorithm of Breiman et al. does not necessarily generate a subtree sequence with regularly decreasing sizes. This situation is indicated by the symbols c and r on the curves. The values of the corresponding complexity parameter α is shown on the top of the frame.

essential idea behind [128], [117], and [148] is again the two-stage encoding: description of a tree and the data using that tree in binary strings. The MDL principle suggests picking the tree giving the shortest code length. As we saw in Chapter 3, this principle frees the user from any parameter setting such as M or α in the minimal complexity pruning. More precisely, the codelength of this two-stage code $L(\boldsymbol{y}, T)$ for the response vector $\boldsymbol{y}$ in the training dataset and a given tree T can be written as

$$L(\boldsymbol{y}, T) = L(\boldsymbol{y} \,|\, T) + L(T), \tag{A.4}$$

where $L(\boldsymbol{y} \,|\, T)$ represents the codelength for describing $\boldsymbol{y}$ given the tree T and $L(T)$ is the codelength of the tree itself. The MDL principle suggests picking a tree T^* minimizing (A.4),

$$MDL(\boldsymbol{y}, T^*) = \min_{T \preceq T_{\max}} L(\boldsymbol{y}, T). \tag{A.5}$$

To derive an actual algorithm to compute T^*, we first consider the second term $L(T)$ of (A.4), i.e., how to encode a tree generated by the CART algorithm. Now let us check how a tree is represented in the S and S-PLUS package. Let us consider the small tree shown in Figure A.2 which is a pruned (by our MDL algorithm) version of the full CT displayed in Figure 5.1. The following is the printout of this tree.

```
node), split, n, deviance, yval, (yprob)
      * denotes terminal node

1) root 300 659.20 class1 ( 0.33330 0.33330 0.33330 )
  2) x.3<-4.13682 131 175.30 class2 ( 0.28240 0.70230 0.01527 )
    4) x.6<-0.182365 77  36.99 class2 ( 0.02597 0.94810 0.02597 ) *
    5) x.6>-0.182365 54  70.05 class1 ( 0.64810 0.35190 0.00000 ) *
  3) x.3>-4.13682 169 279.90 class3 ( 0.37280 0.04734 0.57990 )
    6) x.6<-1.1214 87  53.42 class3 ( 0.00000 0.09195 0.90800 ) *
    7) x.6>-1.1214 82  88.78 class1 ( 0.76830 0.00000 0.23170 ) *
```

We note that the representation of an RT has the averages of the responses instead of the class names and has no class probabilities. Suppose T under consideration has k

terminal nodes. Since this is a binary tree, it is easy to see that the total number of nodes including internal nodes is $2k-1$. Encoding the tree structure is very simple: use 0 for internal nodes (in this example, the root and nodes #2 and #3) and 1 for terminal nodes (nodes #4–#7). Any nontrivial tree, however, always starts with the root node (which is an internal node) and ends with the terminal node. Hence it is not necessary to describe the first 0 and the last 1. Now the structure of T can be represented as a binary string of length $2k-3$ containing $k-1$ 1s fot $k > 1$. The code for the example shown in Figure A.2 is 01101. From (3.4), the description length of such binary string is $\log \binom{2k-3}{k-1}$. Except the tree structure, we still need to encode the information contained in each node. Each node $t \in T$ has: 1) the coordinate index m ($1 \leq m \leq N = \dim(\boldsymbol{x})$), 2) the coordinate threshold θ_m which is a real number, and 3) the node value (i.e., the class label for CT, and the average of the responses for RT which is a real number). We note that the deviance is not necessary for a tree description. These codelengths (ignoring the integer requirement again) can be easily computed using the examples studied in Chapter 3. Let us use the same notation as the previous chapters, i.e., N and C denote the number of total training samples and the number of classes in the training dataset. Then the codelengths of the above three terms are: 1) $\log N$, 2) $(1/2)\log N$, and 3) $\log C$ for CT and $(1/2)\log N$ for RT, respectively. We still can reduce the codelength by noting that each splitting rule is duplicated with only difference in inequality directions as shown in the printout of the tree representation in S and S-PLUS. Nodes pointed by a left branch has always "<" and the ones pointed by a right branch has always ">" in the splitting rule. Thus, we can reduce the description length of the coordinate indices and thresholds by half. Also, it is not necessary to describe these attributes for the root node; only information encoded at the root node is the node value whose description length is simply a constant $c = \log \gamma$ where $\gamma = C$ for classification and

$\gamma = \sqrt{N}$ for regression. Hence the total codelength to describe the tree is

$$\begin{aligned} L(T) &= \log \binom{2k-3}{k-1} + (2k-2)\left(\frac{1}{2}\log n + \frac{1}{4}\log N + \log \gamma\right) + c \\ &= \log \binom{2k-3}{k-1} + (k-1)\log(n\sqrt{N}\gamma^2) + c. \end{aligned} \tag{A.6}$$

Now we proceed to the encoding of the response vector $\boldsymbol{y}$ given the tree T, i.e., the second term of (A.4). For the CTs, each vector $\boldsymbol{x}_i$ in the training class reaches one of the terminal nodes in the tree. Let us record y_i in that node. Then after dropping all the training vectors into this tree, each terminal node has a subsequence of the class sequence $\boldsymbol{y} = (y_1, \ldots, y_N)$. Rissanen suggests using the sum of the description lengths of these subsequences. More precisely, let $\boldsymbol{y}(t)$ denote the subsequence of $\boldsymbol{y} = (y_1, \ldots, y_N)$ which reached t. Let $N_c(t)$ be the number of class c samples reached t and let $N(t) = \sum_{c=1}^{C} N_c(t)$. As we can see, the subsequence $\boldsymbol{y}(t)$ can be described as a string of C-ary alphabet of length $N(t)$ whose complexity can be computed via (3.5). Thus, we have

$$L(\boldsymbol{y} \mid T) = \sum_{t \in \widetilde{T}} L(y(t) \mid T), \tag{A.7}$$

where

$$L(y(t) \mid T) = \log \binom{N(t) + C - 1}{N(t)} + \log \binom{N(t)}{N_1(t) \cdots N_C(t)}. \tag{A.8}$$

For the RTs, we need to assume some probability distributions. Since S and S-PLUS assume the i.i.d. Gaussian model and the RSS is used as the deviance for growing trees, we also follow this assumption, i.e., $y_i \sim \mathcal{N}(\mu_i, \sigma^2)$ for $i = 1, \ldots, N$, and σ^2 is unknown. The same procedure in Section 3.4 can be applied (i.e., the unknown σ^2 and μ_i are estimated by the maximum likelihood technique constrained by the tree-based regression), and we have

$$L(\boldsymbol{y} \mid T) = \frac{N}{2} \log \|\boldsymbol{y} - \widehat{\boldsymbol{y}}(T)\|^2 + c', \tag{A.9}$$

where $\widehat{\boldsymbol{y}}(T)$ denotes the RT estimate of $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_N)$, i.e., a piecewise constant function, given T and c' is a constant independent of T. We can now compute the MDL value (A.4) for any tree in CART combining (A.6), (A.7), (A.8), and (A.9).

Remark A.1. We note that to compute the terms of log of binomial, say, $\log \binom{p}{q}$ where $p \geq q$ are both nonnegative integers, it is faster to use the built-in log-gamma function if available than the plain multiplication for large p. Since $p! = \Gamma(p+1)$ we should expand

$$\log \binom{p}{q} = \log \Gamma(p+1) - \log \Gamma(q+1) - \log \Gamma(p-q+1)).$$

Then we only need 3 times of log-gamma function calls) rather than directly computing $\log(p!/q!(p-q)!)$ (i.e., $2(p-q)$ multiplications, 1 division, and 1 log function call).

Figure A.3 shows the description lengths of the subtrees in the sequence. This plot suggests using the subtree with 4 terminal node for this example.

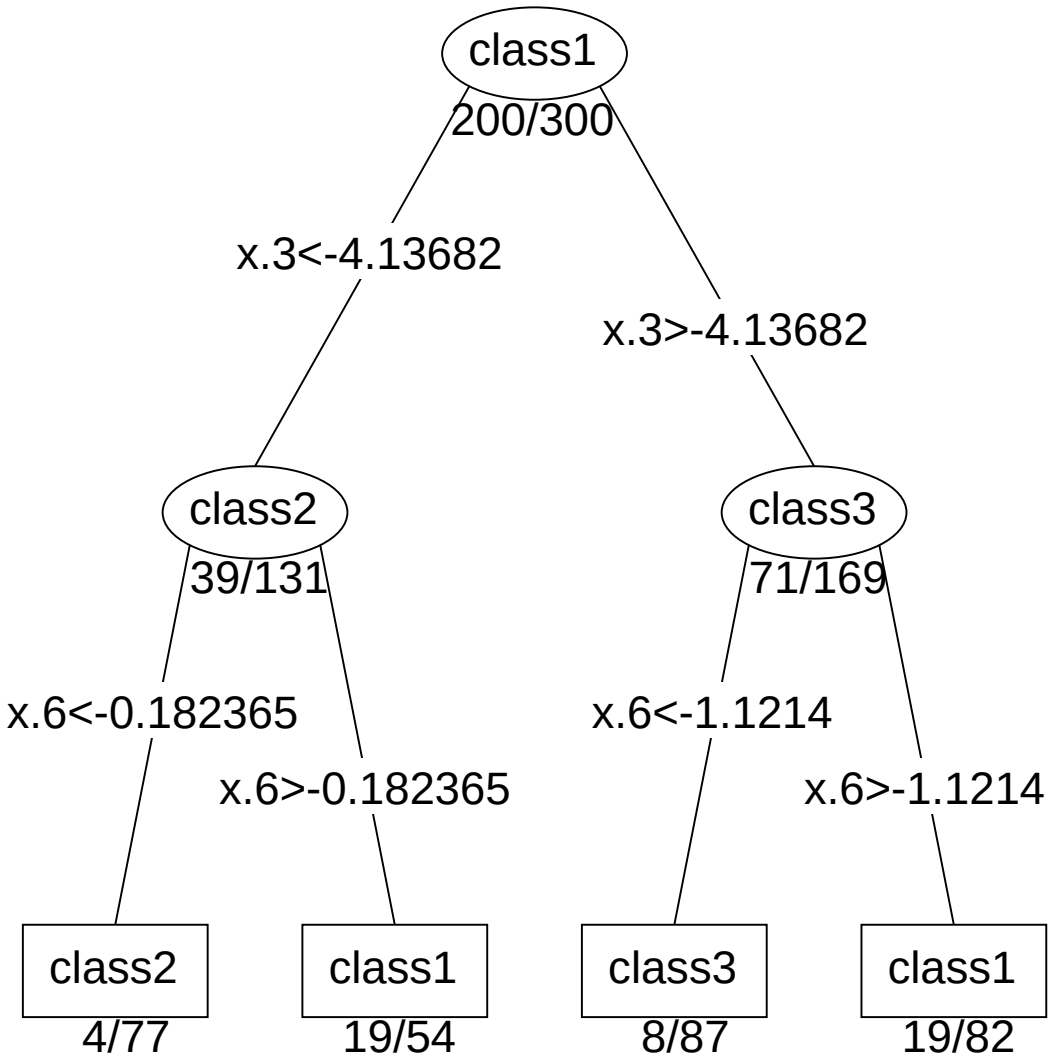


Figure A.2: The pruned classification tree (by the MDL-based pruning algorithm) from the full tree shown in Figure 5.1.

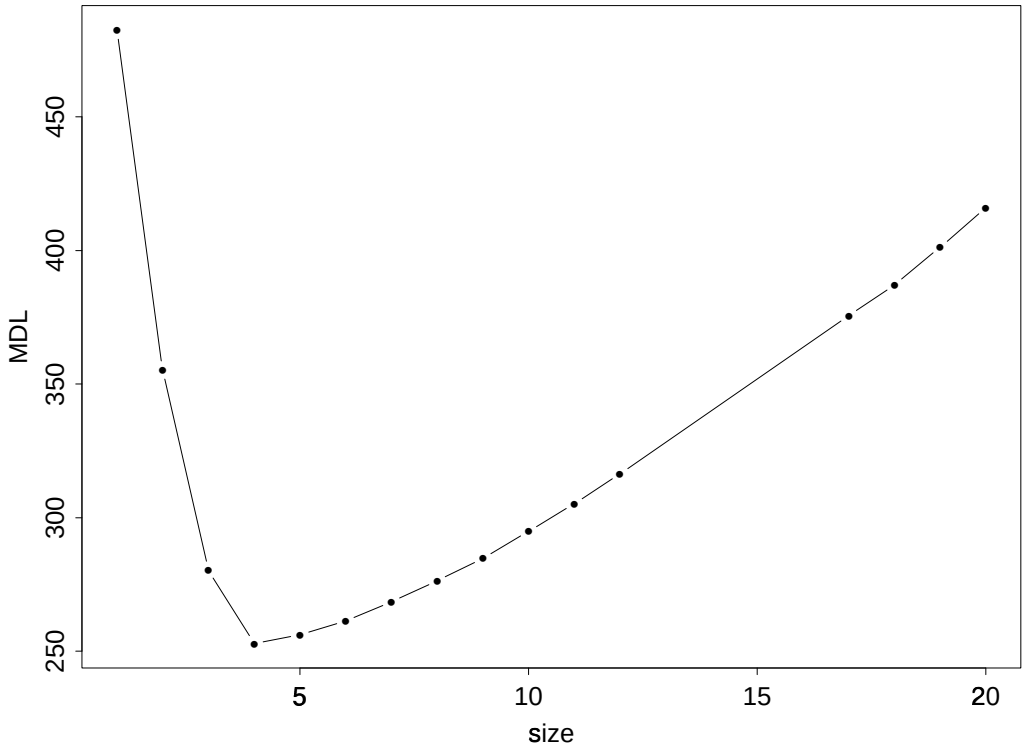


Figure A.3: A curve of subtree size versus MDL value of the tree shown in Figure 5.1. The subtree with 4 terminal nodes gives the minimum value.

Bibliography

[1] N. Ahmed and K. R. Rao, *Orthogonal Transforms for Digital Signal Processing*, Springer-Verlag, New York, 1975.

[2] S. Amari, *Invariant structures of signal and feature space in pattern recognition problems*, RAAG Memoirs **4** (1968), no. 1-2, 553–566.

[3] P. Auscher, G. Weiss, and M. V. Wickerhauser, *Local sine and cosine bases of Coifman and Meyer and the construction of smooth wavelets*, Wavelets: A Tutorial in Theory and Applications (C. K. Chui, ed.), Academic Press, 1992, pp. 237–256.

[4] B. R. Bakshi and G. Stephanopoulos, *Wave-net: a multiresolution, hierarchical neural network with localized learning*, AIChE Journal **39** (1993), no. 1, 57–81.

[5] R. G. Baraniuk and D. L. Jones, *Shear madness: new orthonormal bases and frames using chirp functions*, IEEE Trans. Signal Processing **41** (1993), no. 12, 3543–3549.

[6] A. R. Barron, *Complexity regularization with application to artificial neural networks*, Nonparametric Function Estimation and Related Topics (G. Roussas, ed.), Kluwer Academic Publishers, Dordrecht, The Netherlands, 1991, pp. 561–576.

[7] A. R. Barron and T. M. Cover, *Minimum complexity density estimation*, IEEE Trans. Inform. Theory **37** (1991), no. 4, 1034–1054.

[8] K. A. Bartels and A. C. Bovik, *Shape change analysis and shape modeling using three dimensional biomedical images*, Proc. ICASSP-93, vol. 5, IEEE, 1993, pp. 93–96.

[9] M. Basseville, *Distance measures for signal processing and pattern recognition*, Signal Processing **18** (1989), no. 4, 349–369.

[10] R. A. Becker, J. M. Chambers, and A. R. Wilks, *The New S Language: A Programming Environment for Data Analysis and Graphics*, Chapman & Hall, Inc., New York, 1988.

[11] Z. Berman and J. S. Baras, *Properties of the multiscale maxima and zero-crossings representations*, IEEE Trans. Signal Processing **41** (1993), no. 12, 3216–3231.

[12] G. Beylkin, *On the representation of operators in bases of compactly supported wavelets*, SIAM J. Numer. Anal. **29** (1992), no. 6, 1716–1740.

[13] G. Beylkin, R. Coifman, and V. Rokhlin, *Fast wavelet transforms and numerical algorithms I*, Comm. Pure Appl. Math. **44** (1991), 141–183.

[14] G. Beylkin and B. Torrésani, *Implementation of operators via filter banks, autocorrelation shell and Hardy wavelets*, Appl. Comput. Harmonic Anal. (1994), submitted.

[15] T. I. Boubez and R. L. Peskin, *Multiresolution neural networks*, Wavelet Applications (H. H. Szu, ed.), Apr. 1994, Proc. SPIE 2242, pp. 649–660.

[16] B. Bradie, R. Coifman, and A. Grossmann, *Fast numerical computations of oscillatory integrals related to acoustic scattering, I*, Appl. Comput. Harmonic Anal. **1** (1993), no. 1, 94–99.

[17] J. N. Bradley and C. M. Brislawn, *Image compression by vector quantization of multiresolution decompositions*, Physica D **60** (1992), 245–258.

[18] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Chapman & Hall, Inc., New York, 1993, previously published by Wadsworth & Brooks/Cole in 1984.

[19] P. J. Burt, *Fast filter transforms for image processing*, Comput. Graphics and Image Processing **16** (1981), 20–51.

[20] ———, *Algorithms and architectures for smart sensing*, Proc. Image Understanding Workshop, Apr. 1988, pp. 139–153.

[21] P. J. Burt and E. H. Adelson, *The Laplacian pyramid as a compact image code*, IEEE Trans. Communications **31** (1983), no. 4, 532–540.

[22] D. E. Cannon and G. R. Coates, *Applying mineral knowledge to standard log interpretation*, Trans. Soc. Prof. Well Log Anal. 31st Annual Logging Symposium, 1990, Paper V.

[23] J. M. Chambers and T. R. Hastie, *Statistical Models in S*, Chapman & Hall, Inc., New York, 1992.

[24] B. Cheng and D. M. Titterington, *Neural networks: a review from a statistical perspective (with comments)*, Statist. Sci. **9** (1994), no. 1, 2–54.

[25] J. J. Clark, M. R. Palmer, and P. D. Lawrence, *A transformation method for the reconstruction of functions from nonuniformly spaced samples*, IEEE Trans. Acoust., Speech, Signal Processing **ASSP-33** (1985), no. 4, 1151–1165.

[26] L. A. Clark and D. Pregibon, *Tree-based models*, Statistical Models in S (J. M. Chambers and T. R. Hastie, eds.), Chapman & Hall, Inc., New York, 1992, pp. 377–419.

[27] A. Cohen, I. Daubechies, and J.-C. Feauveau, *Biorthogonal bases of compactly supported wavelets*, Comm. Pure Appl. Math. **45** (1992), no. 5, 485–560.

[28] A. Cohen, I. Daubechies, and P. Vial, *Wavelets on the interval and fast wavelet transforms*, Appl. Comput. Harmonic Anal. **1** (1993), no. 1, 54–81.

[29] R. Coifman, G. Matviyenko, and Y. Meyer, *Wigner-Ville distributions and related atomic decompositions*, Appl. Comput. Harmonic Anal. (1994), submitted.

[30] R. R. Coifman and F. Majid, *Adapted waveform analysis and denoising*, Progress in Wavelet Analysis and Applications (Y. Meyer and S. Roques, eds.), Editions Frontieres, B.P.33, 91192 Gif-sur-Yvette Cedex, France, 1993, pp. 63–76.

[31] R. R. Coifman and Y. Meyer, *Nouvelles bases orthonormées de $L^2(\mathbb{R})$ ayant la structure du systèm de Walsh*, preprint, Dept. of Mathematics, Yale University, New Haven, CT, Aug. 1989.

[32] ———, *Orthonormal wave packet bases*, preprint, Dept. of Mathematics, Yale University, New Haven, CT, 1990.

[33] ———, *Remarques sur l'analyse de fourier à fenêtre*, Comptes Rendus Acad. Sci. Paris, Série I **312** (1991), 259–261.

[34] R. R. Coifman and N. Saito, *Constructions of local orthonormal bases for classification and regression*, Comptes Rendus Acad. Sci. Paris, Série I **319** (1994), no. 2, 191–196.

[35] R. R. Coifman and M. V. Wickerhauser, *Entropy-based algorithms for best basis selection*, IEEE Trans. Inform. Theory **38** (1992), no. 2, 713–719.

[36] ———, *Wavelets and adapted waveform analysis*, Wavelets: Mathematics and Applications (J. Benedetto and M. Frazier, eds.), CRC Press, Boca Raton, FL, 1993.

[37] J. W. Cooley and J. W. Tukey, *An algorithm for the machine calculation of complex Fourier series*, Math. Comp. **19** (1965), 297–301.

[38] T. M. Cover, *The best two independent measurements are not the two best*, IEEE Trans. Syst. Man Cybern. **SMC-4** (1974), no. 1, 116–117.

[39] T. M. Cover and P. Hart, *Nearest neighbor pattern classification*, IEEE Trans. Inform. Theory **IT-13** (1967), 21–27.

[40] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley Interscience, New York, 1991.

[41] S. R. Curtis and A. V. Oppenheim, *Reconstruction of multidimensional signals from zero crossings*, J. Opt. Soc. Amer. A **4** (1987), no. 1, 221–231.

[42] I. Daubechies, *Orthonormal bases of compactly supported wavelets*, Comm. Pure Appl. Math. **41** (1988), 909–996.

[43] ———, *Ten Lectures on Wavelets*, CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 61, SIAM, Philadelphia, 1992.

[44] ———, *Orthonormal bases of compactly supported wavelets II. Variations on a theme*, SIAM J. Math. Anal. **24** (1993), no. 2, 499–519.

[45] I. Daubechies and J. C. Lagarias, *Two-scale difference equations I. Existence and global regularity of solutions*, SIAM J. Math. Anal. **22** (1991), no. 5, 1388–1410.

[46] ———, *Two-scale difference equations II. Local regularity, infinite products of matrices and fractals*, SIAM J. Math. Anal. **23** (1992), no. 4, 1031–1079.

[47] G. Deslauriers and S. Dubuc, *Symmetric iterative interpolation processes*, Constructive Approximation **5** (1989), 49–68.

[48] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice-Hall International, Inc., London, 1982.

[49] R. A. DeVore, B. Jawerth, and B. J. Lucier, *Image compression through wavelet transform coding*, IEEE Trans. Inform. Theory **38** (1992), no. 2, 719–746.

[50] D. L. Donoho, *Interpolating wavelet transforms*, Technical Report 408, Dept. Statistics, Stanford University, Stanford, CA, Oct. 1992.

[51] ———, *Nonlinear wavelet methods for recovery of signals, densities, and spectra from indirect and noisy data*, Proc. Symp. Appl. Math. (I. Daubechies, ed.), AMS, Providence, RI, 1993, pp. 173–205.

[52] ———, *Wavelet shrinkage and W.V.D.: A 10-minute tour*, Progress in Wavelet Analysis and Applications (Y. Meyer and S. Roques, eds.), Editions Frontieres, B.P.33, 91192 Gif-sur-Yvette Cedex, France, 1993, pp. 109–128.

[53] D. L. Donoho and I. M. Johnstone, *Ideal spatial adaptation by wavelet shrinkage*, Biometrika (1994), to appear.

[54] N. R. Draper and H. Smith, *Applied Regression Analysis*, second ed., John Wiley & Sons, New York, 1981.

[55] S. Dubuc, *Interpolation through an iterative scheme*, J. Math. Anal. Appl. **114** (1986), 185–204.

[56] P. Dutilleux, *An implementation of the "algorithme à trous" to compute the wavelet transform*, Wavelets, Time-Frequency Methods and Phase Space (J. M. Combes, A. Grossmann, and Ph. Tchamitchian, eds.), Springer-Verlag, New York, 1989, pp. 298–304.

[57] M. Duval-Destin, M.A. Muschietti, and B. Torrésani, *Continuous wavelet decompositions, multiresolution and contrast analysis*, SIAM J. Math. Anal. **24** (1993), no. 3, 739–755.

[58] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*, Chapman & Hall, Inc., New York, 1993.

[59] R. A. Fisher, *The use of multiple measurements in taxonomic problems*, Ann. Eugenics **7** (1936), 179–188.

[60] J. H. Friedman and W. Stuetzle, *Projection pursuit regression*, J. Amer. Statist. Assoc. **76** (1981), 817–823.

[61] J. H. Friedman and J. W. Tukey, *A projection pursuit algorithm for exploratory data analysis*, IEEE Trans. Comput. **23** (1974), 881–890.

[62] K. S. Fu, *Syntactic Pattern Recognition and Applications*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1982.

[63] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, second ed., Academic Press, San Diego, CA, 1990.

[64] U. Grenander, *General Pattern Theory: A Mathematical Study of Regular Structures*, Oxford Univ. Press, Oxford, 1993.

[65] K. Gröchenig and W. R. Madych, *Multiresolution analysis, Haar bases, and self-similar tilings of R^n*, IEEE Trans. Inform. Theory **38** (1992), no. 2, 556–568.

[66] A. Grossmann, *Wavelet transforms and edge detection*, Stochastic Processes in Physics and Engineering (S. Albeverio, P. Blanchard, M. Hazewinkel, and L. Streit, eds.), D. Reidel Publishing Company, 1988, pp. 149–157.

[67] H. Guo and S. B. Gelfand, *Classification trees with neural network feature extraction*, IEEE Trans. Neural Networks **3** (1992), no. 6, 923–933.

[68] W. S. Harlan, J. F. Claerbout, and F. Rocca, *Signal/noise separation and velocity estimation*, Geophysics **49** (1984), no. 11, 1869–1880.

[69] C. Herley, J. Kovačević, K. Ramchandran, and M. Vetterli, *Tilings of the time-frequency plane: construction of arbitrary orthogonal bases and fast tiling algorithms*, IEEE Trans. Signal Processing **41** (1993), no. 12, 3341–3359.

[70] V. Hirsinger and B. E. Hobbs, *A general harmonic coordinate transformation to simulate the states of strain in inhomogeneously deformed rocks*, J. Struct. Geol. **5** (1983), no. 3/4, 307–320.

[71] R. E. Hoard, *Sonic waveform logging: a new way to obtain subsurface geologic information*, Trans. SPWLA 24th Annual Logging Symposium, 1983, Paper XX.

[72] M. Holschneider, R. Kronland-Martinet, J. Morlet, and A. Grossmann, *A real-time algorithm for signal analysis with the help of the wavelet transform*, Wavelets, Time-Frequency Methods and Phase Space (J. M. Combes, A. Grossmann, and Ph. Tchamitchian, eds.), Springer-Verlag, New York, 1989, pp. 286–297.

[73] T. Hopper, *Compression of gray-scale fingerprint images*, Wavelet Applications (H. H. Szu, ed.), Apr. 1994, Proc. SPIE 2242, pp. 180–187.

[74] H. Hotelling, *Analysis of a complex of statistical variables into principal components*, J. Educ. Psych. **24** (1933), 417–441;498–520.

[75] K. Hsu, *Wave separation and feature extraction of acoustic well-logging waveforms using Karhunen-Loeve transformation*, Geophysics **55** (1990), no. 2, 176–184.

[76] P. J. Huber, *Robust Statistics*, John Wiley & Sons, 1981.

[77] ———, *Projection pursuit (with discussion)*, Ann. Statist. **13** (1985), no. 2, 435–525.

[78] R. Hummel and R. Moniot, *Reconstruction from zero crossings in scale space*, IEEE Trans. Acoust., Speech, Signal Processing **37** (1989), no. 12, 2111–2130.

[79] K. Kanatani, *Group-Theoretical Methods in Image Understanding*, Springer-Verlag, 1990.

[80] K. Karhunen, *Über Linearen Methoden in der Wahrscheinlichkeitsrechnung*, Ann. Acad. Sci. Fennicae, Ser. A **37** (1947), no. 1.

[81] C. V. Kimball and T. L. Marzetta, *Semblance processing of borehole acoustic array data*, Geophysics **49** (1984), no. 3, 274–281.

[82] A. N. Kolmogorov, *Three approaches to the quantitative definition of information*, Problems Inform. Transmission **1** (1965), no. 1, 1–7.

[83] G. Korvin, *Fractal Models in the Earth Sciences*, Elsevier, Amsterdam, The Netherlands, 1992.

[84] J. Kovačević and M. Vetterli, *Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for R^n*, IEEE Trans. Inform. Theory **38** (1992), no. 2, 533–555.

[85] R. Kronland-Martinet, J. Morlet, and A. Grossmann, *Analysis of sound patterns through wavelet transforms*, J. Pattern Recognition and Artificial Intell. **1** (1987), no. 2, 273–302.

[86] J. B. Kruskal, *Toward a practical method which helps uncover the structure of a set of multivariate observations by finding the linear transformation which optimizes a new 'index of condensation'*, Statistical Computation (R. C. Milton and J. A. Nelder eds., eds.), Academic Press, New York, 1969, pp. 427–440.

[87] S. Kullback and R. A. Leibler, *On information and sufficiency*, Ann. Math. Statist. **22** (1951), 79–86.

[88] Y. G. Leclerc, *Constructing simple stable descriptions for image partitioning*, Intern. J. Computer Vision **3** (1989), 73–102.

[89] R. Lenz, *Group Theoretical Methods in Image Processing*, Lecture Notes in Computer Science, vol. 413, Springer-Verlag, 1990.

[90] M. Li and P. Vitányi, *An Introduction to Kolmogorov Complexity and Its Applications*, Springer-Verlag, New York, 1993.

[91] M. Loève, *Sur les fonctions aléatoires stationnaires de second ordere*, Rev. Sci. **83** (1945), 297–310.

[92] J. Lu, J. B. Weaver, and D. M. Healy, Jr., *Noise reduction with multiscale edge representation and perceptual criteria*, Proc. IEEE Intern. Symp. Time-Frequency and Time-Scale Analysis, Victoria, British Columbia, Oct. 1992, pp. 555–585.

[93] S. Mallat, *Multiresolution approximations and wavelet orthonormal bases in* $\mathbf{L}^2(\mathbf{R})$, Trans. Amer. Math. Soc. **315** (1989), 69–87.

[94] ———, *A theory for multiresolution signal decomposition*, IEEE Trans. Pattern Anal. Machine Intell. **11** (1989), no. 7, 674–693.

[95] ———, *Zero-crossings of a wavelet transform*, IEEE Trans. Inform. Theory **37** (1991), no. 4, 1019–1033.

[96] S. Mallat and W. L. Hwang, *Singularity detection and processing with wavelets*, IEEE Trans. Inform. Theory **38** (1992), no. 2, 617–643.

[97] S. Mallat and Z. Zhang, *Matching pursuit with time-frequency dictionaries*, IEEE Trans. Signal Processing **41** (1993), no. 12, 3397–3415.

[98] S. Mallat and S. Zhong, *Characterization of signals from multiscale edges*, IEEE Trans. Pattern Anal. Machine Intell. **14** (1992), no. 7, 710–732.

[99] H. S. Malvar, *The LOT: transform coding without blocking effects*, IEEE Trans. Acoust., Speech, Signal Processing **37** (1989), 553–559.

[100] ———, *Lapped transforms for efficient transform/subband coding*, IEEE Trans. Acoust., Speech, Signal Processing **38** (1990), 969–978.

[101] D. Marr, *Vision*, W.H.Freeman and Company, 1982.

[102] D. Marr and E. Hildreth, *Theory of edge detection*, Proc. Royal Soc. London, Ser. B **207** (1980), 187–217.

[103] G. J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*, John Wiely & Sons, New York, 1992.

[104] Y. Meyer, *Ondelettes et functions splines*, Technical report, séminaire EDP, Ecole Polytechnique, Paris, France, 1986.

[105] ———, *Wavelets: Algorithms and Applications*, SIAM, Philadelphia, PA, 1993, Translated and revised by R. D. Ryan.

[106] ———, *Wavelets and Operators*, Cambridge Studies in Advanced Mathematics, vol. 37, Cambridge Univ. Press, New York, 1993, Translated by D. H. Salinger.

[107] M. L. Minsky and S. A. Papert, *Perceptrons*, expanded edition ed., MIT Press, Cambridge, MA, 1988, (First Edition, 1969).

[108] D. Mumford, *Pattern theory: a unifying perspective*, Proc. First European Congress of Mathematicians, Birkhauser, 1993.

[109] W. Murphy, A. Reischer, and K. Hsu, *Modulus decomposition of compressional and shear velocities in sand bodies*, Geophysics **58** (1993), no. 2, 227–239.

[110] W. Niblack, *MDL Methods in Image Analysis and Computer Vision*, New York, Jun. 1993, IEEE Conf. Comput. Vision, Pattern Recognition, Tutorial note.

[111] S. Osher and L. I. Rudin, *Feature-oriented image enhancement using shock filters*, SIAM J. Numer. Anal. **27** (1990), no. 4, 919–940.

[112] N. Otsu, *Mathematical studies on feature extraction in pattern recognition*, Researches of the Electrotechnical Laboratory 818, Electrotechnical Laboratory, 1-1-4, Umezono, Sakura-machi, Niihari-gun, Ibaraki, JAPAN, Jul. 1981, in Japanese.

[113] Y. Pati and P. Krishnaprasad, *Analysis and synthesis of feed forward neural networks using discrete affine wavelet transforms*, IEEE Trans. Neural Networks **4** (1993), no. 1, 73–85.

[114] T. Pavlidis, *Structural Pattern Recognition*, Springer-Verlag, New York, 1977.

[115] P. Perona and J. Malik, *Scale-space and edge detection using anisotropic diffusion*, IEEE Trans. Pattern Anal. Machine Intell. **12** (1990), no. 7, 629–639.

[116] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, second ed., Cambridge Univ. Press, 1992.

[117] J. R. Quinlan and R. L. Rivest, *Inferring decision trees using the minimum description length principle*, Information and Control **80** (1989), 227–248.

[118] J. Quirein, S. Kimminau, J. LaVigne, J. Singer, and F. Wendel, *A coherent framework for developing and applying multiple formation evaluation models*, Trans. Soc. Prof. Well Log Anal. 27th Annual Logging Symposium, 1986, Paper DD.

[119] M. G. Rahim, *A neural tree network for phoneme classification*, Proc. ICASSP-92, IEEE, 1992, pp. 345–348.

[120] C. R. Rao, *Linear Statistical Inference and Its Applications*, second ed., John Wiley & Sons, New York, 1973.

[121] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, and Applications*, Academic Press, San Diego, CA, 1990.

[122] O. Rioul, *Regular wavelets: a discrete-time approach*, IEEE Trans. Signal Processing **41** (1993), no. 12, 3572–3579.

[123] O. Rioul and P. Duhamel, *Fast algorithms for discrete and continuous wavelet transforms*, IEEE Trans. Inform. Theory **38** (1992), no. 2, 569–586.

[124] O. Rioul and M. Vetterli, *Wavelets and signal processing*, IEEE SP Magazine **8** (1991), no. 4, 14–38.

[125] B. D. Ripley, *Statistical aspects of neural networks*, Networks and Chaos: Statistical and Probabilistic Aspects (O. E. Barndorff-Nielsen, J. L. Jensen, D. R. Cox, and W. S. Kendall, eds.), Chapman & Hall, Inc., New York, 1993, pp. 40–123.

[126] J. Rissanen, *A universal prior for integers and estimation by minimum description length*, Ann. Statist. **11** (1983), no. 2, 416–431.

[127] ———, *Universal coding, information, prediction, and estimation*, IEEE Trans. Inform. Theory **30** (1984), no. 4, 629–636.

[128] ———, *Stochastic Complexity in Statistical Inquiry*, World Scientific, Singapore, 1989.

[129] N. Saito, *Simultaneous noise suppression and signal compression using a library of orthonormal bases and the minimum description length criterion*, Wavelets in Geophysics (E. Foufoula-Georgiou and P. Kumar, eds.), Academic Press, San Diego, CA, 1994, pp. 299–324.

[130] ———, *Simultaneous noise suppression and signal compression using a library of*

orthonormal bases and the minimum description length criterion, Wavelet Applications (H. H. Szu, ed.), Apr. 1994, Proc. SPIE 2242, pp. 224–235.

[131] N. Saito and G. Beylkin, *Multiresolution representations using the auto-correlation functions of compactly supported wavelets*, Tech. report, Schlumberger-Doll Research, Aug. 1991, Expanded abstract in Proceedings of ICASSP–92, vol. 4, pp. 381–384, Mar. 1992.

[132] ———, *Multiresolution representations using the auto-correlation functions of compactly supported wavelets*, Proc. ICASSP-92, vol. 4, IEEE, 1992, pp. 381–384.

[133] ———, *Multiresolution representations using the auto-correlation functions of compactly supported wavelets*, IEEE Trans. Signal Processing **41** (1993), no. 12, 3584–3590.

[134] ———, *Multiresolution representations using the auto-correlation functions of wavelets*, Progress in Wavelet Analysis and Applications (Y. Meyer and S. Roques, eds.), Editions Frontieres, B.P.33, 91192 Gif-sur-Yvette Cedex, France, 1993, pp. 721–726.

[135] N. Saito and R. R. Coifman, *Local discriminant bases*, Mathematical Imaging: Wavelet Applications in Signal and Image Processing (A. F. Laine and M. A. Unser, eds.), Jul. 1994, Proc. SPIE 2303.

[136] H. Sakoe and S. Chiba, *A dynamic programming approach to continuous speech recognition*, Proc. 7th Intern. Congress Acoust., Budapest, Hungary, 1971, Paper 20C-13.

[137] J. Segman, J. Rubinstein, and Y. Y. Zeevi, *The canonical coordinates method for pattern deformation: theoretical and computational considerations*, IEEE Trans. Pattern Anal. Machine Intell. **14** (1992), no. 12, 1171–1183.

[138] J. M. Shapiro, *Image coding using the embedded zerotree wavelet algorithm*, Math-

ematical Imaging: Wavelet Applications in Signal and Image Processing (A. F. Laine, ed.), 1993, Proc. SPIE 2034, pp. 180–193.

[139] M. J. Shensa, *The discrete wavelet transform: wedding the à trous and Mallat algorithms*, IEEE Trans. Signal Processing **40** (1992), no. 10, 2464–2482.

[140] StatSci, *S-PLUS Reference Manual, Vol. 1 & 2, version 3.2*, Seattle, WA, Dec. 1993.

[141] J.-E. Strömberg, J. Zrida, and A. Isaksson, *Neural trees – using neural nets in a tree classifier structure*, Proc. ICASSP-91, IEEE, 1991, pp. 137–140.

[142] H. H. Szu, B. Telfer, and S. Kadambe, *Neural network adaptive wavelets for signal representation and classification*, Opt. Eng. **31** (1992), no. 9, 1907–1916.

[143] R. H. Tatham, V_p/V_s *and lithology*, Geophysics **47** (1982), no. 3, 336–344.

[144] J. Tittman, *Geophysical Well Logging*, Academic Press, Orlando, FL, 1986.

[145] G. T. Toussaint, *Note on optimal selection of independent binary-valued features for pattern recognition*, IEEE Trans. Inform. Theory **IT-17** (1971), no. 5, 618.

[146] D. L. Turcotte, *Fractals and Chaos in Geology and Geophysics*, Cambridge Univ. Press, New York, 1992.

[147] M. Vetterli and C. Herley, *Wavelets and filter banks: theory and design*, IEEE Trans. Signal Processing **40** (1992), no. 9, 2207–2232.

[148] C. S. Wallace and J. D. Patrick, *Coding decision trees*, Machine Learning **11** (1993), 7–22.

[149] R. S. Wallace, *Finding natural clusters through entropy minimization*, Ph.D. thesis, School of Comput. Science, Carnegie Mellon Univ., Pittsburgh, PA 15213, Jun. 1989.

[150] S. Watanabe, *Karhunen-Loève expansion and factor analysis: theoretical remarks and applications*, Trans. 4th Prague Conf. Inform. Theory, Statist. Decision Functions,

Random Processes (Prague), Publishing House of the Czechoslovak Academy of Sciences, 1967, pp. 635–660.

[151] ———, *Pattern recognition as a quest for minimum entropy*, Pattern Recognition **13** (1981), no. 5, 381–387.

[152] ———, *Pattern Recognition: Human and Mechanical*, John Wiley & Sons, New York, 1985.

[153] M. Wax and T. Kailath, *Detection of signals by information theoretic criteria*, IEEE Trans. Acoust., Speech, Signal Processing **ASSP–33** (1985), no. 2, 387–392.

[154] J. E. White, *Underground Sound: Applications of Seismic Waves*, Methods in Geochemistry and Geophysics, vol. 18, Elsevier, New York, 1983.

[155] M. V. Wickerhauser, *Fast approximate factor analysis*, Curves and Surfaces in Computer Vision and Graphics II, Oct. 1991, Proc. SPIE 1610, pp. 23–32.

[156] ———, *High-resolution still picture compression*, Digital Signal Processing: A Review Journal **2** (1992), no. 4, 204–226.

[157] ———, *Adapted Wavelet Analysis from Theory to Software*, A K Peters, Ltd., Wellesley, Massachusetts, 1994, with diskette.

[158] K. Winkler and W. Murphy III, *Acoustic velocity and attenuation in porous rocks*, AGU Review volume, Amer. Geophys. Union, 1994, in press.

[159] Y. Y. Zeevi and E. Shlomot, *Nonuniform sampling and antialiasing in image representation*, IEEE Trans. Signal Processing **41** (1993), no. 3, 1223–1236.

[160] Q. Zhang and A. Benveniste, *Wavelet networks*, IEEE Trans. Neural Networks **3** (1992), 889–898.