INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

DICTIONARY PROJECTION PURSUIT: A WAVELET PACKET TECHNIQUE FOR ACOUSTIC SPECTRAL FEATURE EXTRACTION

by

GLEN A. RUTLEDGE
M.Sc., University of Victoria, 1994
B.Sc., McMaster University, 1990

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Mechanical Engineering

We accept this dissertation as conforming
to the required standard

Dr. G. McLean, Supervisor, Dept. of Mechanical Engineering

Dr. M. Nahon, Dept. of Mechanical Engineering

Dr. R. Podhorodeski, Dept. of Mechanical Engineering

Dr. W.S. Lu, Outside Member, Dept. of Electrical Engineering

Dr. V. Cuperman, External Examiner, Dept. of Electrical Engineering, UCSB

Supervisor: Dr. G. McLean

ABSTRACT

This thesis uses the powerful mathematics of wavelet packet signal processing to efficiently extract features from sampled acoustic spectra for the purpose of discriminating between different classes of sounds. An algorithm called dictionary projection pursuit (DPP) is developed which is a fast approximate version of the projection pursuit (PP) algorithm [P.J. Huber *Projection Pursuit*, Annals of Statistics, 13 (**2**) 435–525, 1985]. When used with a wavelet packet or cosine packet dictionary, this algorithm is significantly faster than the PP algorithm with relatively little degradation in performance provided that the multivariate vectors are samples of an underlying continuous waveform or image. The DPP algorithm is applied to the problem of approximating the Karhunen-Loève transform (KLT) in high dimensional spaces and simulations are performed to compare this algorithm to Wickerhauser's approximate KLT algorithm [M.V. Wickerhauser. *Adapted Wavelet Analysis from Theory to Software*, A.K. Peters Ltd, 1994]. Both algorithms perform very well relative to the eigenanalysis form of the KLT algorithm at a small fraction of the computational cost.

The DPP algorithm is then applied to the problem of finding discriminant features in acoustic spectra for sound recognition tasks; extensive simulations are performed to compare this algorithm to previously developed dictionary methods for discrimination such as Saito and Coifman's local discriminant bases [N. Saito and R. Coifman. Local Discriminant Bases and their Applications. *Journal of Mathematical Imaging and Vision*,5(**4**) 337–358, 1995] and Buckheit and Donoho's discriminant pursuit [J. Buckheit and D. Donoho. Improved Linear Discrimination Using Time-Frequency Dictionaries. *Proceedings of SPIE Wavelet Applications in Signal and Image Processing III Vol 2569*, 540–551, July, 1995]. It is found that each feature extraction algorithm performs well under different conditions, but the DPP algorithm is the most flexible and consistent performer.

Examiners:

Dr. G. McLean, Supervisor, Dept. of Mechanical Engineering

Dr. M. Nahon, Dept. of Mechanical Engineering

Dr. R. Podhorodeski, Dept. of Mechanical Engineering

Dr. W.S. Lu, Outside Member, Dept. of Electrical Engineering

Dr. V. Cuperman, External Examiner, Dept. of Electrical Engineering, UCSB

Table of Contents

List of Figures

List of Tables

List of Algorithms

Notation

$\mathbb{R}$	vector space of all real numbers
$\mathbb{R}^n$	vector space of all real vectors of length n
$\mathbb{R}^{n \times m}$	vector space of all real matrices of size $n \times m$
$\mathbb{C}$	vector space of all complex numbers with extensions to vectors and matrices
$\mathbb{Z}$	vector space of all integers with extensions to vectors and matrices
n, N, a, A	examples of scalars, which are typeset non-bold
M	dimensionality of the pattern vectors
N	total number of pattern vectors
K	number of classes
$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix}$	column vector or pattern vector, which are typeset in lower case bold
$\boldsymbol{x}^T = \begin{bmatrix} x_1 x_2 \cdots x_M \end{bmatrix}$	row vector or transpose of a pattern vector
$\|\boldsymbol{x}\|_p = [\sum \|x_i\|^p]^{1/p}$	p-norm of a vector
$\|\boldsymbol{x}\| = \|\boldsymbol{x}\|_2$	default norm of a vector is the 2-norm
$\|\boldsymbol{x}\| = \begin{bmatrix} \|x_1\| \\ \|x_2\| \\ \vdots \\ \|x_M\| \end{bmatrix}$	absolute value or magnitude of the components of a vector
$\boldsymbol{A}, \boldsymbol{B}$	examples of matrices, which are typeset in uppercase case bold
$p(\boldsymbol{\omega}^{(i)})$	*a priori* probability - scalar value representing the probability of class i ocurring

$p(\boldsymbol{x})$	unconditional probability density function - a multivariate continuous function defined on $\mathbb{R}^M$ giving the probability that a sample drawn from any class has a value of $\boldsymbol{x}$
$p(\boldsymbol{x}\|\boldsymbol{\omega}^{(i)})$	conditional probability density function - a multivariate continuous function defined on $\mathbb{R}^M$ giving the probability that a random sample drawn from $\boldsymbol{\omega}^{(i)}$ has a value of $\boldsymbol{x}$
$p(\boldsymbol{\omega}^{(i)}\|\boldsymbol{x})$	*a posteriori* probability density function - a multivariate continuous function defined on $\mathbb{R}^M$ giving the probability that a sample with a value of $\boldsymbol{x}$ was drawn from $\boldsymbol{\omega}^{(i)}$
$p(\boldsymbol{x}, \boldsymbol{\omega}^{(i)})$	joint probability density function - a multivariate continuous function defined on $\mathbb{R}^M \times \mathbb{Z}$ giving the probability that a sample has both a value of $\boldsymbol{x}$ and was drawn from $\boldsymbol{\omega}^{(i)}$
$\mathcal{A}, \mathcal{B}$	examples of sets, which are typeset in uppercase calligraphy
$\emptyset$	empty set
$\boldsymbol{x} \in \mathcal{X}$	The vector $\boldsymbol{x}$ is an element of $\mathcal{X}$
$\boldsymbol{y} \notin \mathcal{X}$	The vector $\boldsymbol{y}$ is not an element of $\mathcal{X}$
$\mathcal{A} \subset \mathcal{B}$	$\mathcal{A}$ is a subset of $\mathcal{B}$
$\mathcal{A} \supset \mathcal{B}$	$\mathcal{A}$ is a superset of $\mathcal{B}$
$\mathcal{A} \cup \mathcal{B}$	union of $\mathcal{A}$ and $\mathcal{B}$
$\mathcal{A} - \mathcal{B}$	difference between $\mathcal{A}$ and $\mathcal{B}$ (*i.e.*, elements which are in $\mathcal{A}$ but not in $\mathcal{B}$)
$\mathcal{A} \cap \mathcal{B}$	intersection of $\mathcal{A}$ and $\mathcal{B}$
$\mathcal{A} \overline{\cap} \mathcal{B}$	elements not in the intersection of $\mathcal{A}$ and $\mathcal{B}$
$\{\boldsymbol{x}\|\cdots\}$	set builder notation to be read as "all $\boldsymbol{x}$ satisfying conditions $\cdots$"

$\Rightarrow$	implies
$\equiv$	equivalent
iff	if and only if
$\forall$	for all
$\exists$	there exists
$x \stackrel{\text{def}}{=} y^2$	x is defined to be y^2

Acknowledgements

First I would like to thank my supervisor Ged, who gave me the freedom to find my own path in the pot pourri of academic pursuits. I could not have found a better supervisor.

I would like to thank all of my marina friends, Tyro, Kelly, Ron, Amy, Stef, Cookie, Jim, Sandy, Ian, Nancy, Ray, Denise, Tim, Sue, Ted, etc. who have put up with me being a recluse while I worked on my thesis. Thanks to Mike for pulling me away mid-Phd to do a very interesting 'tree falls in a forest' project. Hope there are many more to come Mike! Thanks to my brother Dave, his wife Linda and son Liam for the great get-away trips to the great white north. Looking forward to seeing more of you in the future. Thanks to Chris, Dar and Dave Hicks for all the back-country ski trips, and sorry I couldn't go on any this winter. Next winter!! Thanks to my office-mates Jason and Cam, for being such great people. Sorry I left at the end. Thanks to Henk and Aja for giving me the opportunity to work in such a great office for the last part of my PhD. It made a wonderful difference! Thanks to the research group at IVL, and especially Peter, for providing such a great work environment and interesting discussions.

Probably the most thanks must go to my parents Fred and Shirley who have shown such great support not only for this PhD, but throughout my whole life. Much of what I am, I owe to you.

Last but in no way the least, thank-you Nicole for all the great times that we have shared. You have made my life whole. Let the trips begin!

Dedication

To my Parents, for all their love and support.

Chapter 1

Introduction

1.1 Objective

The primary objective of this thesis is to develop algorithms that take advantage of the powerful mathematics of wavelet packet signal processing to efficiently extract useful features from sampled acoustic spectra for the purpose of discriminating between different classes of sounds. The secondary objective of this thesis is to perform extensive classification experiments on real and synthetic data to evaluate the algorithms developed in this thesis with respect to traditional feature extraction techniques such as the Karhunen-Loève (KL) transform and with respect to other wavelet packet feature extraction techniques developed by other researchers in a logical, objective and unbiased manner in order to demonstrate the advantages and disadvantages of each technique.

1.2 High Level Overview

On the highest level, a typical acoustic pattern recognition system (human or artificial) can be understood using figure 1.1. The very first step is to perform some kind of *sound organization.* In a given acoustic environment, sounds can be produced by many different sources but when measured by the human ear or a microphone, the sounds are all mixed together into a single one dimensional signal. Therefore, some form de-mixer must be employed to assign each component of the signal to the correct sound source.

Once the signal has been decomposed into 'sound objects', each object is analyzed

independently and certain characteristic features are extracted in the *feature extraction* stage. Typical features could be the energy of the signal in a given frequency band, the rise time of the overall energy envelope, or many others. These features are then used by the *classifier* to categorize each sound object into one of many pre-defined classes or symbols. The number of classes can be very large as is the case for humans, or quite small (*i.e.*, two or three classes) as is the case for most engineering applications. The symbolic representation of the sound objects that are present in the acoustic environment are then used in the *high level inference and reasoning* stage to form a 'world view' of what is physically happening in the local environment. In many cases there is a control feedback loop that can affect how each of the previous stages of the system operate.

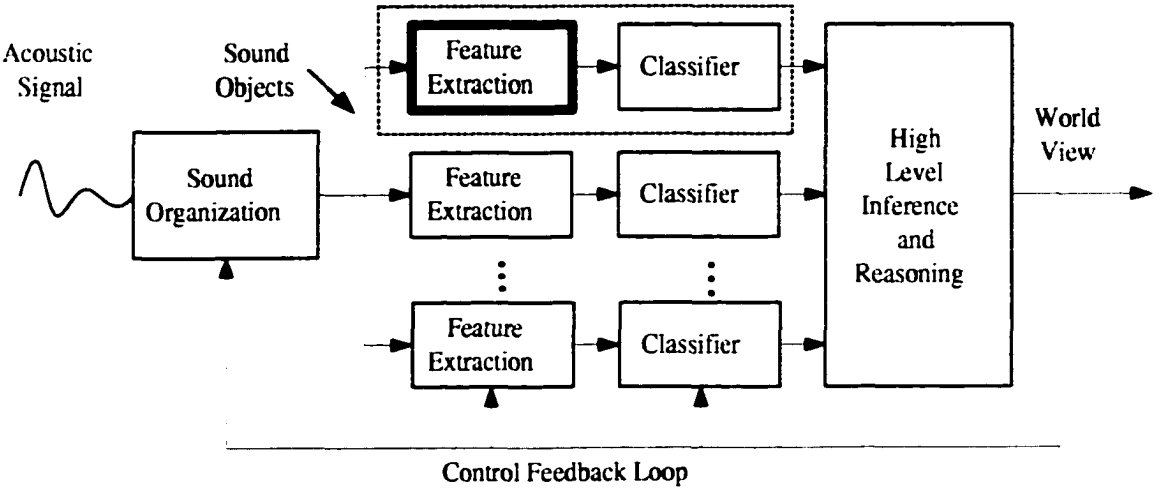

Figure 1.1. *High level overview of acoustic pattern recognition.*

Although the level of detail and complexity that was just described is necessary for the human audition system, many engineering applications only require the *feature extraction* and *classifier* stage, as depicted by the dotted line in figure 1.1. In these applications, the acoustic environment is usually assumed to be relatively inactive with only one sound source being present at a given time so the need for the *sound organization* stage is removed. The goal of the system is simply to classify the individual sounds that are heard into a small number of pre-defined classes, so the requirement for *high level reasoning and inference* is removed. It is these types of systems that are of interest in this thesis. In particular, only the *feature extraction*

stage of the acoustic pattern recognition system is addressed in a novel way, as depicted by the bold outline in figure 1.1. A traditional and well established classifier is used as a tool to evaluate the feature extraction algorithms that are developed.

Wavelet packets are used in this thesis as a tool to efficiently search for discriminant features in the spectral representation of acoustic signals. Therefore all the features that are extracted are spectral features from the frequency domain. Time domain and time-frequency domain features are not addressed in this thesis. This point should be emphasized since it is tempting to automatically assume that time-frequency features are being extracted when wavelet packets are used in the feature extraction process.

The basic idea of the wavelet packet feature extraction algorithm developed in this thesis can be understood by referring to figure 1.2. The signals are projected onto the wavelet packet basis functions which means that the correlation between the signal and the wavelet packet is evaluated. If the signals from one class are well correlated and the signals from the other class are *not* well correlated with a particular wavelet packet (*i.e.*, class II is well correlated and class I is not well correlated with wavelet packet I), then this is considered to be a 'good' wavelet packet for pattern recognition, and the projection coefficients on this wavelet packet can be used as a feature for a classifier. If both or neither class are well correlated with a particular wavelet packet (both classes are well correlated with wavelet packet II, and neither are well correlated with wavelet packet III), then this is considered to be a 'poor' wavelet packet for pattern recognition.

Of course a class of signals has internal variability as well, so it is important to consider the separation of the ensemble of signals from the classes rather than the separation of individual signals from the classes as discussed above. Figure 1.3 shows the projection coefficients (*i.e.*, the strength of the correlation) for an ensemble of signals from the hypothetical classes and first two wavelet packet basis functions shown in figure 1.2. In case (a), the signals from each class are well behaved, show approximately Normal distributions and are well separated. Therefore the projection coefficient on wavelet packet I is indeed a good feature to use for pattern recognition. However, in case (b), the individual signals from class I and II plotted in figure 1.2 coincidentally show good separation. When looking at the distribution as a whole, it

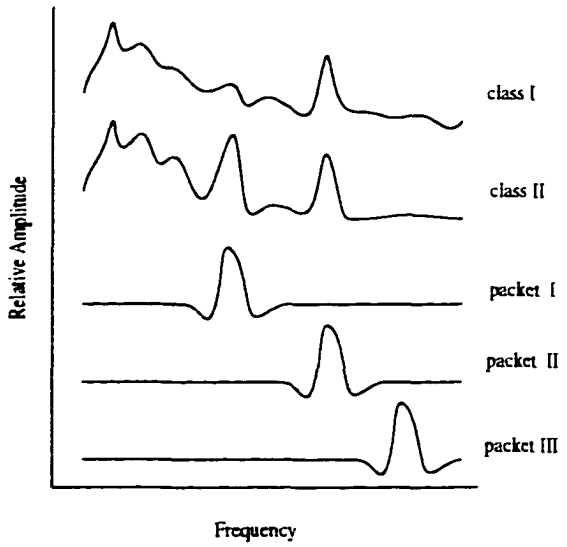

Figure 1.2. *The top signal shows a typical spectra from class I, the second signal shows a typical signal from class II, and the third, fourth and fifth signals show three basis functions from a wavelet packet dictionary.*

can be seen that the projection coefficient on wavelet packet I is not a very good feature to use for pattern recognition since it does not provide good separation between the classes in a statistical sense. This example should emphasize the importance of looking at the ensemble of signals from a class rather than the signals individually. The concept of separating classes based on an ensemble of example spectra from the classes will be made very clear in a quantitative sense later in this thesis.

The approach taken in this thesis is to use the currently available fast algorithms for computing projection coefficients on a dictionary of wavelet packet basis functions to efficiently search for those basis functions that provide the best separation between the classes. The algorithm developed in this thesis is called discriminant dictionary projection pursuit since it is an approximate fast version of the well known projection pursuit algorithm [61] applied to the discrimination problem. Although the algorithm is tested only on sound spectra it is also applicable to many other areas that require waveform recognition such as mass/gas spectroscopy in chemistry and stellar/galactic spectroscopy in astronomy. Also, the core algorithm developed in this thesis, called dictionary projection pursuit, should also be applicable to other multivariate estimation problems such as regression, density estimation, and finding outliers in a dataset,

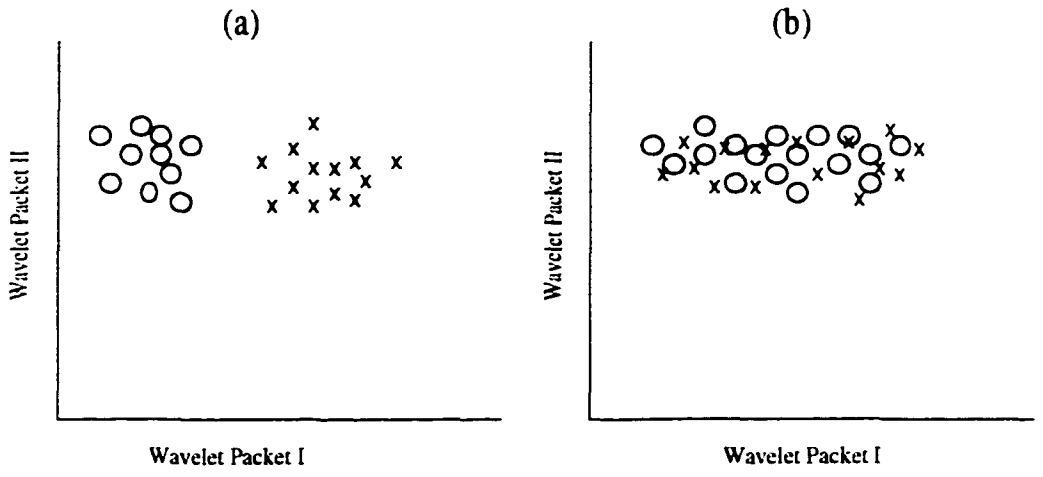

Figure 1.3. *The projection coefficients of signals from class I ('o') and II ('×') on the wavelet packet basis functions I and II. Typical spectra from these classes and the wavelet packet basis functions are plotted in figure 1.2.*

but these paths are not studied in this thesis.

1.3 Motivation

The motivation for this work comes from a wide variety of engineering applications requiring sound recognition capabilities, but in particular from environmental sound recognition tasks where there are a wide variety of possible sound sources with potentially very different spectral features that discriminate them from each other. Feature extraction is an extremely important component of any pattern recognition system, but is especially important for the case where the features are samples from an underlying continuous waveform such as an acoustic spectrum. Feature extraction serves two functions within an acoustic pattern recognition system:

1. It defines the dimensionality of the feature vector which in turn defines the complexity of the classification task; thus by minimizing the dimensionality, the system can be trained more efficiently and accurately with a smaller number of

training samples.

2. The features that are extracted from the acoustic spectra indicate which patterns in the spectrum provide the most discriminant information; this increases our understanding on a physical level of the spectral differences between various classes of sounds.

The wavelet packet feature extraction techniques developed in this thesis are particularly well suited to problems where the number of samples available to train the pattern recognition system is small in comparison to the number of samples in the acoustic spectra. It is in this regime that many traditional methods of feature extraction such as Fisher's method break down[1]. An additional advantage of wavelet packet feature extraction techniques over the more traditional KL transform (also called principal component analysis) is the speed with which the feature extractor can be trained. This becomes increasingly important as the number of samples in the acoustic spectra increases and/or when the system must be trained on regular intervals to account for changing environmental or class characteristics.

Due to the many small but important details that are involved with designing and evaluating an acoustic pattern recognition system, it is almost useless to compare a feature extraction technique developed and evaluated in one research paper to another feature extraction technique developed and evaluated by a *different* author in another research paper. Reliable and valid comparisons can only be made under very controlled conditions where all variables except the feature extraction technique are fixed in the comparison. The desire for a valid quantitative comparison between competing feature extraction techniques is the main motivation for the extensive classification experiments that are performed in this thesis. In this way, every conclusion that is made in this thesis can be backed up by quantitative evidence.

The motivations briefly discussed here are elaborated on in the remainder of this section.

[1] It is shown in this thesis that the KL transform (another traditional feature extraction technique) actually performs quite well in this regime contrary to some authors' beliefs [13].

1.3.1 Motivation for Research on Acoustic Pattern Recognition

Sound is a major component of our physical environment. It is used in speech for verbal communication, in music for non-verbal communication, and in general as a means of learning about the vibrations and impacts of objects in our surroundings. Corresponding to each of these tasks, three fields of acoustic pattern recognition research have emerged as shown in figure 1.4. Sound is an invaluable resource for humans trying to cope in a very complex world which provides strong motivation to both

1. increase our scientific understanding of how the human audition system works, and
2. develop sound recognition technologies to enhance or improve various engineering tasks.

Although this thesis is primarily motivated by engineering applications of sound recognition technologies (section 1.3.1.2), a brief review of the work being done to increase our scientific understanding of the human audition system is given in section 1.3.1.1 to show the engineering potential of some of the research in this field.

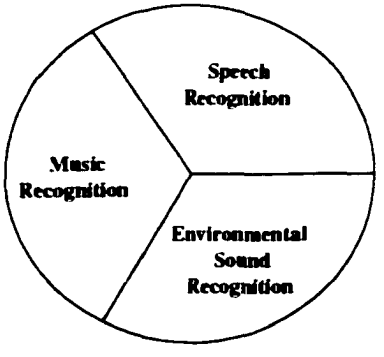

Figure 1.4. *Classification of acoustic pattern recognition research.*

1.3.1.1 Scientific Understanding

The research in this area is primarily concerned with the human perception of acoustic stimuli. There are a large number of books on the basic relationships between quan-

Physical Variable	Psychological Variable
Intensity	Loudness
Frequency	Pitch
Waveform	Timbre

Table 1.1. *Physical and psychological variables*

titative measures of acoustic stimuli and the resultant human perception [6, 24, 41, 57, 65]. Examples of physical variables and the corresponding psychological variables are shown in table 1.1.

Although these relationships are interesting, the main focus of this section is on a relatively new field of study that seeks to understand how the mind organizes and identifies the sources of sounds in a complex acoustic environment. This work has the potential to significantly improve the sound source identification problem of engineering applications which today is essentially ignored.

Through a series of clever psycho-acoustic experiments, Bregman and others have studied the perceptual experience that humans have when listening to various types of acoustic signals. From these experiments, Bregman has developed a set of hypotheses and general principles that describe how the human mind processes auditory information– a process that he calls "auditory scene analysis". This work has been nicely summarized in Bregman's book [8], which can be thought of as the auditory counterpart to Marr's book on vision [81]. The field of computational auditory scene analysis (CASA) has developed to provide an objective tool to test Bregman's hypotheses quantitatively, or in some cases, to directly apply his principles [10, 22, 39, 83], in an attempt to better understand the human audition system. The emphasis tends to be on producing computational models that are capable of reproducing certain features of the human audition system, a component of which is the ability to recognize sound sources in complex environments.

Although research in this area has been plentiful in the last decade (see [107] and references therein), the work is still at a highly theoretical stage, and thus practical applications of this research are yet to emerge. The emphasis is on the organization (*i.e.*, segmentation) of complex acoustic environments into individual sound objects, and the recognition of those sound objects is considered secondary. For instance, while

the sound source recognition problem is mentioned in several articles, there is only one system, namely the 'Sound Understanding Testbed (SUT)' instantiation of the 'Integrated Processing and Understanding of Signals (IPUS)' system, that provides a quantitative evaluation of the recognition rates achieved [67, 74] . Quoting from Ellis [39]

> "... there must be a process of organization or *segmentation* of the auditory signal that is applied prior to (although most likely in conjunction with) the function of recognizing and describing the individual sound objects."

The goal of sound organization in these systems is to collect elements of the acoustic signal into groups that are perceived as a single entity by the human mind as described by Bregman [8]. For example, a harmonic series of tones would be grouped into a single entity with properties of pitch and timbre, rather than distinct tones (harmonicity principle). Low-band and high-band energy would be grouped into a single entity (at least for a few hundred milliseconds) if they have a common onset time (common onset principle). Other grouping principles are common amplitude modulation, and common frequency modulation.

In our everyday lives, the sounds that are heard are mainly from physical objects vibrating, oscillating or slamming into each other. These objects can move around in space, so it is common for sounds coming from an object to have related properties like common onset, common amplitude modulation etc., and this is most likely the reason that the mind developed unconscious grouping mechanisms. That is, since the primary purpose of hearing is to acquire information about the state of physical objects in an environment, the grouping principles are likely an evolutionary result of the mind trying to relieve conscious thinking processes from commonly occurring tasks.

The apparent importance of sound organization when the mind solves the sound recognition problem suggests that a computer automated sound source recognition system should also organize sound to some degree in the process of recognition. However, since the research in this field is still in its infancy, there has not yet been any *efficient* algorithms developed to apply these grouping principles. For this reason, engineering applications discussed in section 1.3.1.2 and the algorithms developed

in this thesis attempt to recognize sounds based on fixed time segmentation, which makes it very difficult to recognize sounds when more than one source is present. Hopefully one day, CASA research will provide an efficient and powerful sound segmentation procedure for engineering applications but in the meantime (*i.e.*, in this thesis), the currently available tools must be used to the best of their ability.

1.3.1.2 Engineering Applications

The engineering motivation for sound recognition technologies has been heavily driven by the speech recognition field which among other things, focusses on improving human-machine interfaces. This field is primarily concerned with identifying and interpreting phonemes, words, and sentences from spoken language, or electronically producing spoken language from typed text [102]. The practical applications of this technology in the business world has fueled much of the research in this field. Because of this, classes of speech sounds are well established and the features that discriminate between them have evolved in a Darwinian sense and thus can be improved on very little. The research in this area is now focussed on other problems such as modelling language and grammar. The research in this thesis is thus not directed towards speech recognition but rather towards 'new' sound recognition problems where the 'good' features have not yet been discovered. Despite this fact, an example from the speech community on phoneme recognition is studied in chapter 6 and interestingly the discriminant dictionary projection pursuit algorithm developed in this thesis finds the same types of features as many decades of manual searching did!

While music recognition technology does not have the same level of economic drive as speech recognition, there are still many engineering applications that motivate research in this area. Some practical applications of music recognition are transcribing musical notation from live or recorded performances, extracting expressive performance information from recorded audio for MIDI encoding[112], recognizing voiced and unvoiced parts of someone singing [20], recognizing specific instruments [92], etc. Since music is a perceptual experience, the research in this area is tied strongly to psychology, which makes it very difficult to evaluate the recognition rates of the classification system in a meaningful way. For this reason, no music-related examples were studied in this thesis, although the algorithms should be directly applicable to

many problems.

Applications of environmental sound recognition technology are quite numerous, but relatively little research has been done in this area. The most likely reason for this is that applications in both speech and music can control their environment to a large extent (*e.g.* most speech recognition systems today require that you speak clearly into a microphone with relatively little background noise), but most applications of environmental sound recognition must operate in an uncontrolled environment. For example a noise monitoring system that is designed to classify noise sources as either planes or trains must operate outside where other sounds will inevitably be detected such as wind, storms, factories, cars, etc. This creates two problems. First, the many new sounds that were not used to train the acoustic pattern recognition system can confuse the system and result in many false positive detections. Second, it is much more likely that more than one sound will occur at a given time which suggests that a sound de-mixer should be used. However, as discussed in section 1.3.1.1, the technology to perform this task has not yet been developed. This thesis sidesteps these issues by only focussing on the feature extraction problem, but it should be remembered that for a field application of this technology, much more work is needed.

The feature extraction algorithms developed in this thesis are well suited to environmental sound recognition problems since there are often new and different classes of sounds that have not been extensively studied. Therefore, one of the first problems is to find the features that best discriminate between them. Common practice is to use the same features that are used for speech recognition, but the work in this thesis proves that this is not the best choice. The spectral features that best discriminate between phonemes are not the features that best discriminate between car and truck sounds.

The following list of environmental sound recognition applications gives the reader an idea of the large number of possibilities for this technology and the references indicate some of the research that has been done in this area. A more thorough list of applications and references can be found in Couvreur's thesis [25].

Surveillance- mostly enforcement or military applications for the detection of ground vehicles [122], gunshots, submarines using passive sonar [66] etc. Also in this area are home alarm systems that can detect footsteps, breaking glass alarms

[52, 95] etc. and alarm systems for underground parking or subway systems that detect screams or breaking glass.

Noise Monitoring - recognition of the noise sources in a noise monitoring system (NMS) [25, 27]. Typically the noise sources are planes, trains, cars etc.

Biological Sound Classification - identification of certain types of birds [3], frogs [99, 100] etc., or whales and shrimp using passive sonar [62, 71].

Robot Hearing - this is an engineering application of the CASA research discussed above which has been investigated by the IPUS group [37, 67, 73, 74, 90, 131].

This non-exhaustive list of engineering applications for environmental sound recognition provides motivation for research in this field, and with the falling price of digital signal processing chips that are capable of doing acoustic processing in real-time, the number of application areas are growing rapidly.

1.3.2 Motivation for Research on Feature Extraction Techniques

Feature extraction is arguably the most important component of designing an acoustic pattern recognition system since even the best classifier will perform poorly if the features are not chosen well. In acoustic pattern recognition, the classes of sounds that are to be discriminated between can be chosen arbitrarily, and thus the features must be able to adapt to the choice of sound classes. For example, in one problem the goal might be to distinguish between cars and planes, and in another problem the goal might be to distinguish between two different models of cars. It is unlikely that the features used in the first problem would be appropriate for the second problem. It is desirable to have an automated method that can choose features based on examples from each of the sound classes in the problem, which is the focus of this thesis.

The remainder of this section was written to give some practical examples of the process of selecting spectral features to discriminate between sounds which also helps to justify the focus in this thesis on spectral features rather than time domain or time-frequency domain features.

Consider the various sounds that are heard around the house. Without using any other sense except audition, it is usually possible to determine what type of

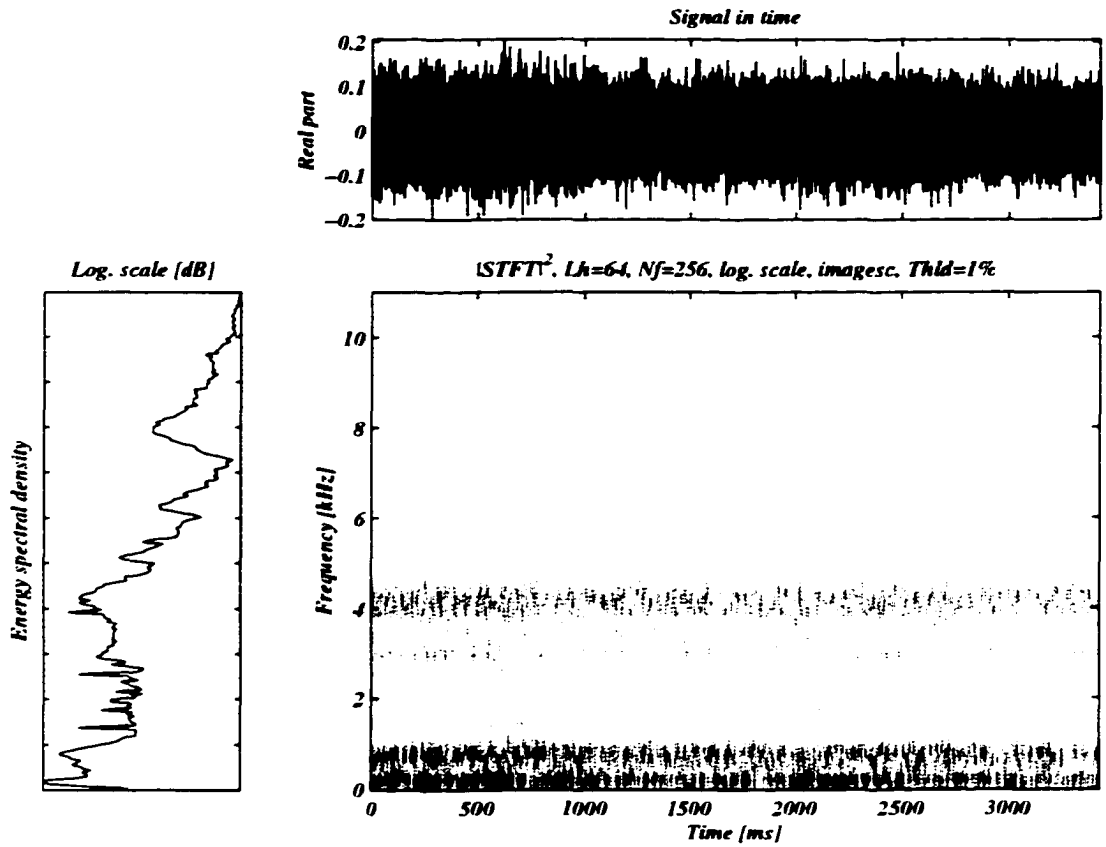

Figure 1.5. *Spectrum, time series and STFT spectrogram for a vacuum cleaner.*

activity is occurring. For instance, sitting in the office, it is easy to determine when someone is vacuuming or when wine glasses are clinking in the dining room because the sounds that distinguish these activities have been learned from past experiences. What features in the acoustic signal actually distinguish between the various sounds? The following examples suggest several possibilities.

Figures 1.5 – 1.8 show example recordings of a vacuum, a hair dryer, a glass clink, and a clap, all sampled at 22 kHz. The plot on the left shows the log spectrum, the plot on the top shows the time domain signal, and the other plot shows the time-frequency distribution computed using the short-time Fourier transform (STFT), where darker color indicates more energy in that time-frequency location.

It is clear that the spectra of the vacuum and hairdryer do not change very much

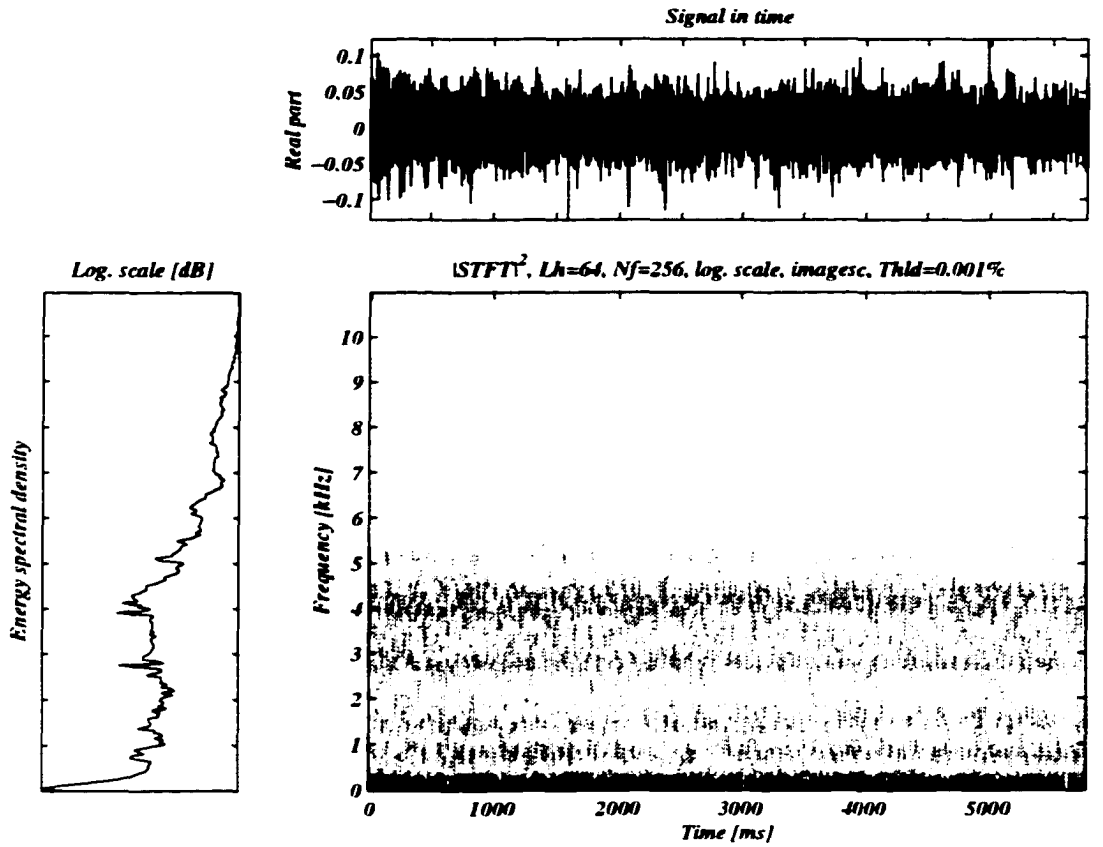

Figure 1.6. *Spectrum, time series and STFT spectrogram for a hair dryer.*

with time (*i.e.*, they are stationary[2]). The features that discriminate between stationary sounds must exist in the frequency domain (*i.e.*, the spectrum), since the time domain carries no information. Therefore looking at the spectra of these two sounds, can you find features that possibly discriminate between a vacuum and a hair dryer?

First of all, their are many similarities between the spectra. Both spectra are fairly flat out to $\approx$ 4 kHz and then show a gradual drop off to $\approx$ 7 kHz, and many of the broad spectral peaks are located in similar positions. The most obvious discriminating feature is that the hair dryer contains a very strong narrow peak at $\approx$ 100 Hz, whereas the vacuum does not contain such a strong peak there. Additionally, the vacuum shows a broad peak near 8 kHz which is not present in the hair dryer spectrum, and there are many more narrow band peaks in the vacuum spectrum than the hair dryer

[2]The word 'stationary' is used in a loose sense. For a rigorous definition, see Papoulis [94].

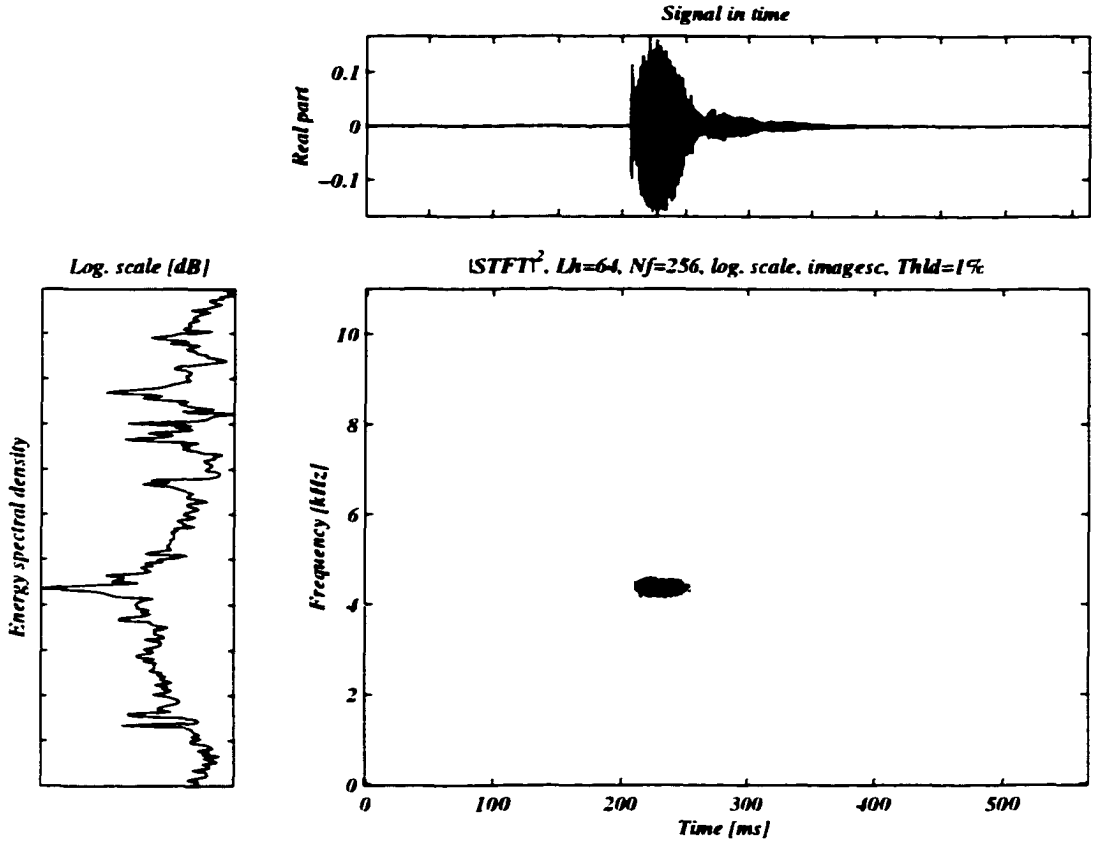

Figure 1.7. *Spectrum, time series and STFT spectrogram for a glass clink.*

spectrum.

In the case of the clap and glass clink shown in figures 1.7 and 1.8, the signals are obviously not stationary. There are clear transients at the onsets and offsets without a stationary period in between. These types of signals are called transients, and they contain information in the time-domain and time-frequency domain that can be used as features for pattern recognition. For example, it is clear that the rise time (*i.e.*, the time for the signal to reach maximum energy) is much shorter for a clap than a glass clink. Also, the decay curve (*i.e.*, the energy profile of the time domain signal in the offset phase) appears to be a uniform exponential for a clap, whereas the glass clink shows an extra energy rise about 100 ms after the maximum energy is reached. This type of information, while important in some applications, is not used in this thesis since the focus is purely on spectral feature extraction.

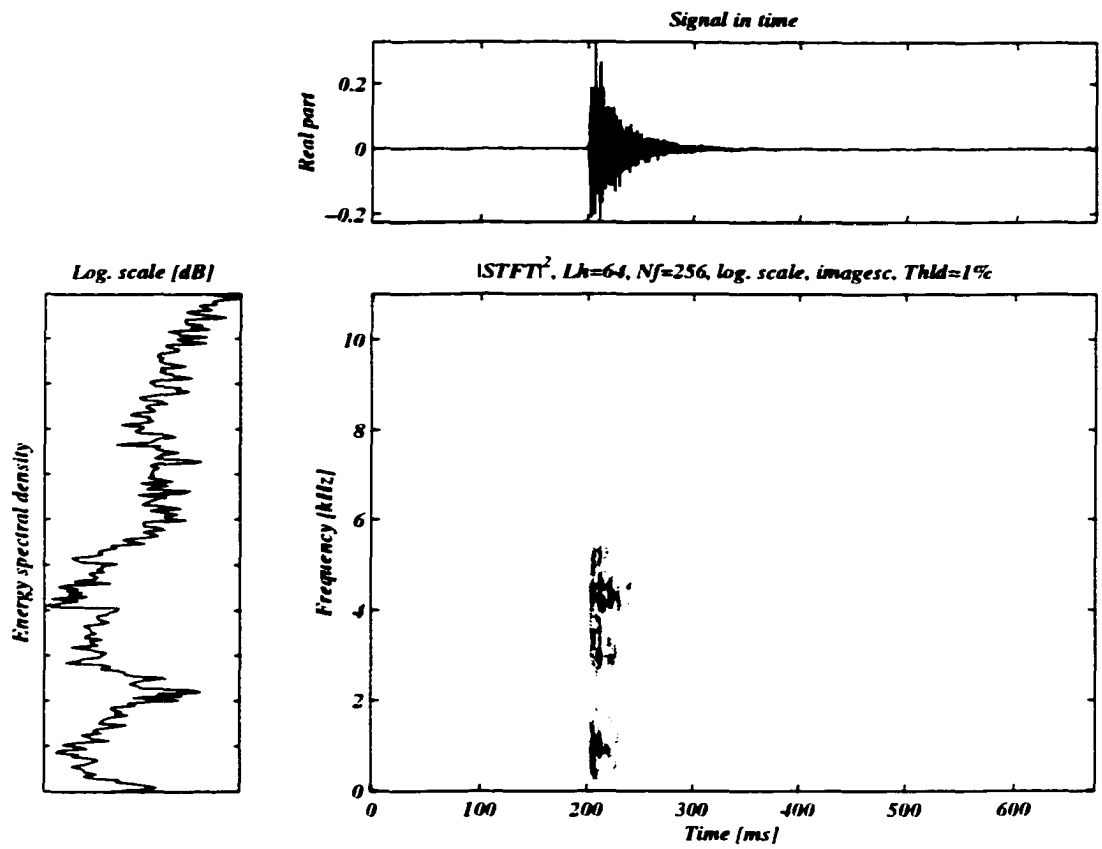

Figure 1.8. *Spectrum, time series and STFT spectrogram for a clap.*

It can be seen that the clap and glass clink can actually be more easily distinguished in the frequency domain. The spectrum of the glass clink shows a strong narrow peak at $\approx$ 4 kHz whereas the clap shows two very broad peaks at $\approx$ 1 kHz and $\approx$ 4 kHz. The obvious next step would be to extract features based on the distribution of energy in the time-frequency plane. This topic is not addressed in this thesis either, but instead is left for future research (see section 2.4.4 for more discussion on this topic).

The method used in this thesis to obtain a quantitative measure of these discriminating features was discussed in section 1.2, which simply amounts to correlating a template with the same shape as the discriminating feature with the signal. In the above example, single spectra from each class of sounds were examined which made it very easy to manually search for discriminating features, but as discussed

in section 1.2 a good feature extractor must also account for the internal variability within the classes which is very hard to do manually. This is the main motivation for researching automated methods of finding discriminant features between classes of sounds.

1.3.3 Motivation for using Wavelet Packets

Wavelet theory has many roots in diverse fields but the formalism of the continuous wavelet transform was developed in the field of geophysics by Morlet [87, 88], Grossmann and Morlet [56], and Goupillaud, Grossman, and Morlet [54]. Since that time, the foundation has been solidified by Meyer [85, 84] and Chui [17], and discrete versions of the transform were developed by Daubechies [29, 30] and Mallat [76, 77]. The wavelet packet transform, a generalization of the discrete wavelet transform, was later proposed by Coifman and Wickerhauser [21, 130]. Today, wavelets and wavelet packets are ubiquitous, being used in almost every field that uses signal processing in any way.

Wavelets and wavelet packets are basis functions (see figures 3.4 and 3.5 for example) which are localized in both time and frequency. This is in contrast to the basis functions of Fourier theory (*i.e.*, complex exponentials) which are perfectly localized in frequency but have no localization in time. Much like the Fourier basis functions, wavelets and wavelet packets can be viewed as an analysis tool (*i.e.*, as an integrating kernel that extracts information from the signal that is correlated with the basis function) or as a construction tool (*i.e.*, as a bases for representing a signal). In this thesis, wavelet packets are used as an analysis tool to extract features from acoustic spectra.

1.3.3.1 How can wavelet packets be used as a feature extraction tool and what is the advantage?

The approach taken in this thesis uses concepts similar to projection pursuit (see Huber [61] and references therein), which is a multivariate technique that uses numerical optimization to find 'interesting projections' of high dimensional datasets. By varying the criterion that is optimized (also called the projection index), different multivariate problems can be solved such as data visualization, regression, density

estimation, and discrimination. It is possible to express standard multivariate problems which are usually solved by eigenanalysis, such as finding the Karhunen Loève (KL) basis functions[3] or finding Fisher's linear discriminant functions in a projection pursuit form.

The advantage of doing this is that the criterion functions that are optimized by projection pursuit are not restricted in the same way as those that are optimized by eigenanalysis. Therefore, the criterion functions for the original problem can be modified to create robust versions, or modified in some other way to overcome shortcomings of the original formulation. In addition, new criteria can be developed that do not have an eigenanalysis equivalent, such as finding projections that maximize the non-Normality in the dataset distribution (since these projections are considered to have high information content [61]). The advantage comes with a price though since 'PP methods have one serious drawback: their high demand on computer time' [61, pg. 437].

In the original PP formulation, numerical optimization is used to find the direction in the multidimensional space that maximizes a criterion function. Therefore, these directions can take on an infinite number of values which accounts for the high computational cost of the algorithm. When the multidimensional vector comes from samples of an underlying continuous waveform, and in particular when they come from a continuous spectrum, the directions that are most likely to provide 'interesting projections' are those that show strong correlations in position and or frequency[4]. Since wavelet packet basis functions have both of these properties, it may be possible to speed up the PP algorithm by only looking at a finite number of projections in the directions defined by the wavelet packets without sacrificing the accuracy of the result too much.

In fact, since there are fast algorithms for computing the wavelet packet trans-

[3]This problem is called principal component analysis (PCA) by statisticians.

[4]The terminology becomes confusing here since the signal that is being analyzed is a spectrum, and thus the position coordinate is actually frequency in Hz, so it is unclear what the oscillations of the frequency should be called. One could use the terminology that developed around cepstral analysis and call it quefrency, but instead, the terms position and frequency are used, where it is understood that position refers to position along the x-axis (which happens to be frequency in Hz) and frequency refers to the rate at which the signal oscillates with respect to the x-axis.

form (*i.e.*, the projections onto a whole family of wavelet packet basis functions), the computational advantage that is gained is dramatic, making these 'fast approximate PP' algorithms significantly faster than both the PP algorithm *and* the eigenanalysis formulation of the problem (see chapter 4 for a comparison of computational complexity). Mallat and Zhang take advantage of this concept in their formulation of the 'Matching Pursuits' algorithm, which uses ideas related to PP to approximate a single signal by a sum of basis functions from a dictionary (see section 3.3.4.2).

The algorithm developed in this thesis (*i.e.*, dictionary projection pursuit in chapter 4) is more directly tied to the PP algorithm than Matching pursuits (MP) since an ensemble of signals is analyzed, but it uses the MP idea of using a dictionary of basis functions to speed up the optimization procedure. Like MP, dictionary projection pursuit does not require that a wavelet packet dictionary be used; any dictionary of waveforms that are thought to be highly probable interesting projections will do. However, the use of the wavelet packet dictionary allows the algorithm to be implemented very efficiently and thus is the only type of dictionary that is studied in this thesis.

While the dictionary projection pursuit algorithm could potentially be used for other multivariate problems in the same manner as the PP algorithm, only the problem of feature extraction is studied in this thesis, in which case the algorithm is called discriminant dictionary projection pursuit (DDPP).

1.3.4 Motivation for large-scale classification experiments

The motivation for doing large scale classification experiments in this thesis comes from the desire to have an unbiased and objective way of measuring the DDPP feature extraction algorithm developed in this thesis against other common feature extraction algorithms. The method in this thesis of using classification error rates with a common classifier to compare feature extraction algorithms is a reasonable and reproducible way of evaluating the DDPP algorithm. Simply comparing error rates of the DDPP algorithm to published error rates of other authors is difficult (since the exact dataset must be obtained) and error prone (due to small differences in the implementation or method of computing the error rates).

It is also important to study how an algorithm performs in a wide variety of sit-

uations. It is tempting (and sometimes carried out in the literature) to only publish the situations where the algorithm performs well, and ignore the others. This approach was not adopted here, and in fact the datasets and evaluation procedure were fixed before the DDPP algorithm was implemented. The importance of the synthetic datasets studied in chapter 5 lies in the fact that everything about the data is known, and thus it is easier to interpret the results. The importance of the recorded datasets studied in chapter 6 is to show how the algorithm works for real problems that might be encountered in a field application. To quote from Huber [61, pg. 525]

> "In order to understand what a technique can and cannot do, we must have some (mathematical or non-mathematical) theoretical understanding, and we must construct synthetical - and find actual - datasets where it works well, moderately well or not at all, and we must know why."

All of these objectives are satisfied in this thesis.

1.4 Thesis Organization and Original Contribution

The requisite background material on acoustic pattern recognition is given in chapter 2, and a review of wavelet packet techniques is given in chapter 3. These two chapters contain very little original contribution (except in the presentation of the material which is a culmination from numerous different sources) but are necessary to set the notation, formalism and concepts for later chapters. The reader who is familiar with either of these subject areas should still quickly peruse these chapter to understand the notation used later on. Chapter 2 does present some unique ideas regarding acoustic pattern recognition design, but the bulk of the original contribution in this thesis is found in chapter 4.

The development of the dictionary projection pursuit algorithm is presented in chapter 4, which includes examples of applying the algorithm to approximating the Karhunen-Loève (KL) basis functions with a comparison to Wickerhauser's approximate KL algorithm [130]. This chapter also presents discriminant dictionary projection pursuit (DDPP) which is simply dictionary projection pursuit with a discriminant criterion function. DDPP is applied to extracting features from waveforms and examples are given for each of the datasets studied in the last two chapters. This chapter

represents the main contribution of original material in this thesis.

Chapters 5 and 6 use synthetic data and recorded data respectively to assess the DDPP algorithm as a tool for extracting features for pattern recognition. The synthetic signal model and Monte Carlo method of computing the Bayes error rate in chapter 5 are original contributions. Comparisons are made with other popular methods of feature extraction such as the KL transform [34, 123] , Saito and Coifman's local discriminant bases [108, 109, 110] and Buckheit and Donoho's discriminant pursuit [13]. Chapter 7 summarizes the findings in this thesis.

Chapter 2

Acoustic Pattern Recognition and Feature Extraction

2.1 Introduction

This chapter provides the requisite background material on acoustic pattern recognition and feature extraction. It is typically easier to understand the concepts of feature extraction if the concepts of pattern recognition are first understood. Therefore, the ideas behind pattern recognition are presented first in section 2.2, followed by the concepts that are specific to acoustic pattern recognition in section 2.3, while the concluding section 2.4 gives a review of acoustic feature extraction.

2.2 Pattern Recognition

2.2.1 Introduction

Pattern recognition is a very broad field with fuzzy boundaries. Informally, it is the study of learning from past experiences. Humans use pattern recognition daily to recognize the faces and voices of their friends, to understand languages, to read books, and a plethora of other activities. The field of study that has come to be known as "pattern recognition" focuses on making machines perform these same operations without the intervention of humans.

The role that pattern recognition plays within the larger context of "understanding", is shown graphically in figure 2.1, where physical signals are converted to observations through some form of physical measurement, observations are converted to

symbols through some form of statistical reasoning, and symbols are converted to concepts through some form of symbolic reasoning. Analogously, three fields of scientific inquiry, "instrumentation and measurement", "pattern recognition" and "artificial intelligence" have developed relatively independently with their own sets of techniques and idiosyncratic terminology[1].

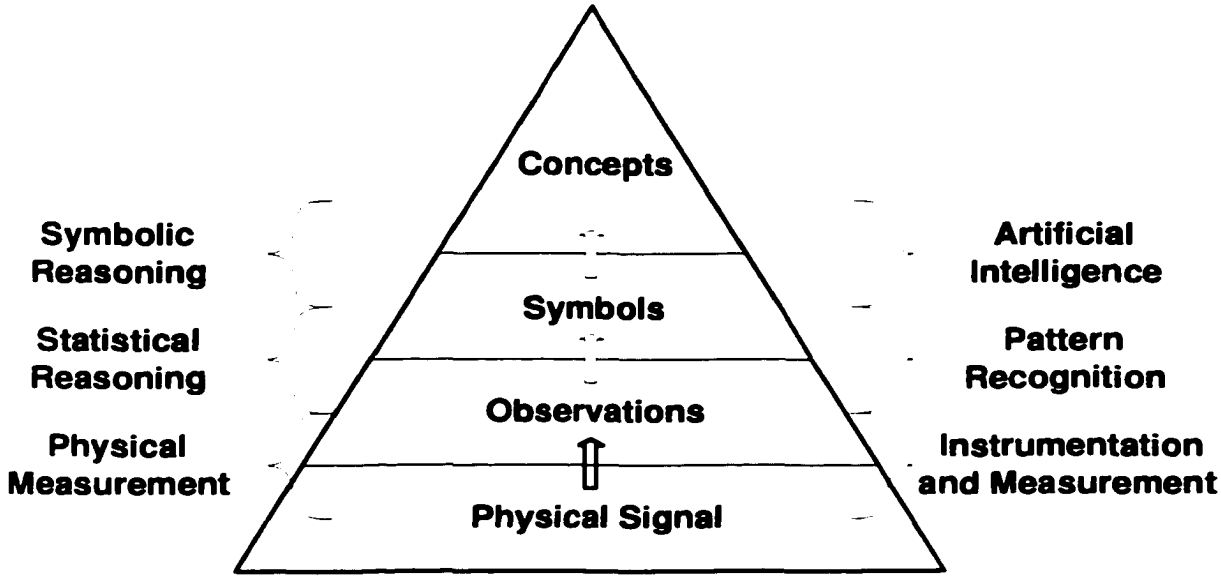

Figure 2.1. *The information hierarchy of forming concepts from measurements. A pyramid structure is used in this figure to show that generally many observations are required to form a symbol, and many symbols are required to form a concept.*

The work in this thesis falls firmly on the boundary between observations and symbols, which will be referred to as pattern recognition, but also goes by the names "pattern classification", "discriminant analysis", "machine learning", and others. Although the pattern recognition problem has been studied in statistics for more than sixty years, and today is a core part of any course on multivariate statistics (see Flury [43] for a recent multivariate statistics text, and McLachlan [82] for a recent review of the statistical literature and pattern recognition approaches from the statistical community), there was a lot of engineering research during the 60's and 70's, which some claim are the true roots of the field of pattern recognition (see Tou and Gonzalez

[1]Some authors prefer to think of pattern recognition as a subfield of artificial intelligence [113], but this is simply a matter of semantics.

[123], Duda and Hart [38] or Devijver and Kittler [34] for a review of the work done during that era[2]). Due to this dichotomous history, different terminology has been developed for this field, so this chapter serves to set the notation and symbols that will be used throughout this thesis. Footnotes will be used to notify the reader when there is distinct terminology from the different backgrounds.

2.2.2 Problem Formulation

The formal purpose of pattern recognition is to assign observations $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^M$ in the real input feature space to one of several classes $y \in \mathcal{Y} \equiv \{\boldsymbol{\omega}^{(1)}, \boldsymbol{\omega}^{(2)}, \ldots, \boldsymbol{\omega}^{(K)}\}$ in the categorical output decision space, where each $\boldsymbol{\omega}^{(k)}$ represents a class[3]. This operation defines a classifier, $\underline{\boldsymbol{D}}$, which can be viewed as a mapping (see figure 2.2)

$$\underline{\boldsymbol{D}} : \boldsymbol{x} \mapsto y, \tag{2.1}$$

or as a disjoint partitioning of the feature space with a class label attached to each partition (see figure 2.3)

$$\underline{\boldsymbol{D}} : \{(A_j, q_j)\}_{j=1}^{J} \text{ such that if } x \in A_j \text{ then } y = \boldsymbol{\omega}^{(q_j)}, \tag{2.2a}$$

where

$$A_j \cap A_i = \emptyset \quad \forall \quad i \neq j \tag{2.2b}$$

and

$$\cup_{j=1}^{J} A_j = \mathcal{X}. \tag{2.2c}$$

[2]During this time, two distinct forms of pattern recognition emerged. The first form, commonly called statistical pattern recognition, is the main concern of this thesis and is simply referred to as pattern recognition. The second form, commonly called structural or syntactic pattern recognition was developed mainly for image recognition where the relative positions of objects contain important information for classification. This type of pattern recognition is not studied in this thesis, but the interested reader is referred to Tou and Gonazalez [123, Ch. 8] or Fu [49].

[3]In the statistical literature, the feature space is referred to as the predictor space, and the decision space is referred to as the response space. Also, the vector $\boldsymbol{x}$ is sometimes referred to as the feature vector and other times simply as a pattern. We will use these two words inter-changeably in this thesis.

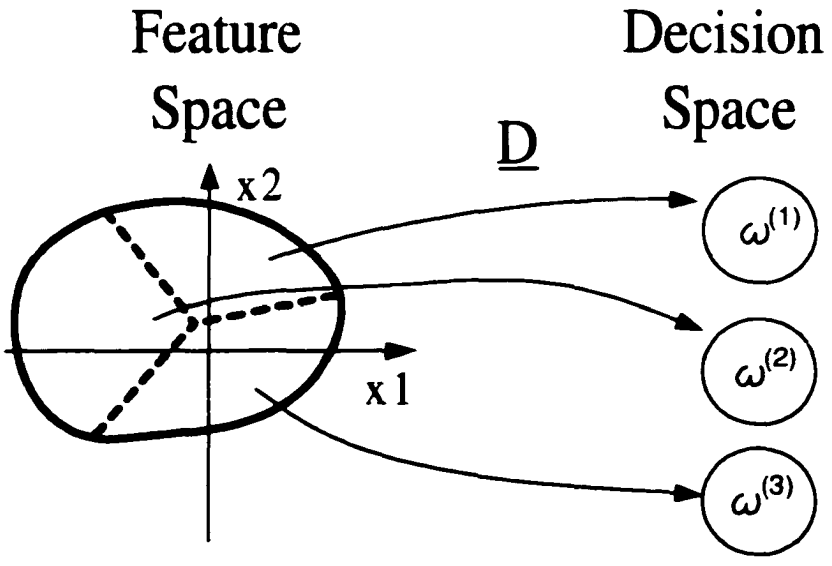

Figure 2.2. *A classifier $\underline{D}$ viewed as a mapping from the feature space $\mathcal{X} \subset \mathbb{R}^2$ to the decision space $\mathcal{Y} = \{\omega^{(1)}, \omega^{(2)}, \omega^{(3)}\}$. For every point in $\mathcal{X}$, there is an image point in the categorical decision space $\mathcal{Y}$.*

2.2.2.1 Example: Female-Male Data

To make this section more concrete, consider the hypothetical example of trying to classify a person as male or female, based only on measurements of their weight and height. Let the feature vector be arranged as

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \text{weight} \\ \text{height} \end{bmatrix}, \tag{2.3}$$

and let the decision space be defined as

$$\mathcal{Y} = \{\text{male}, \text{female}\}. \tag{2.4}$$

Assume that there are several 'examples' that can be used to create our classifier. That is, assume the weight and height of several males and females have been measured from some pre-defined population. Figure 2.4 shows this data plotted with a linear classifier that was chosen by eye. Rearranging the equation on the graph

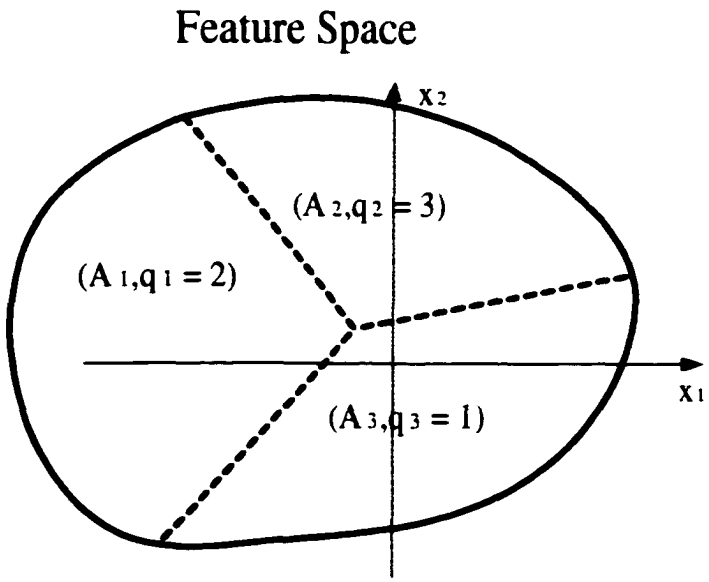

Figure 2.3. *A classifier $\underline{\boldsymbol{D}}$ viewed as a partition of the feature space $\mathcal{X} \subset \mathbb{R}^2$ into 3 disjoint regions A_1, A_2 and A_3, such that each region has a label $\boldsymbol{\omega}^{(q_1)}, \boldsymbol{\omega}^{(q_2)}$ and $\boldsymbol{\omega}^{(q_3)}$ corresponding to the categorical decision space $\mathcal{Y} = \{\boldsymbol{\omega}^{(1)}, \boldsymbol{\omega}^{(2)}, \boldsymbol{\omega}^{(3)}\}$.*

produces the discriminant function[4]

$$\underline{\boldsymbol{d}}(\boldsymbol{x}) = \underline{\boldsymbol{d}}(\text{height,weight}) = \text{height[cm]} + 0.34 \cdot \text{weight[kg]} - 190, \tag{2.5}$$

which can in turn be used to define a classifier

$$\underline{\boldsymbol{D}}(\boldsymbol{x}) = \begin{cases} \text{male} & \text{if } \underline{\boldsymbol{d}}(\boldsymbol{x}) > 0, \\ \text{female} & \text{if } \underline{\boldsymbol{d}}(\boldsymbol{x}) < 0, \\ \text{male or female} & \text{if } \underline{\boldsymbol{d}}(\boldsymbol{x}) = 0. \end{cases} \tag{2.6}$$

So now, if another person is drawn from the same population, and their weight and height are measured, then their gender can be guessed based on equations (2.5) and (2.6).

[4]The term discriminant function is used rather loosely in this thesis to refer to any function $\underline{\boldsymbol{d}}(\boldsymbol{x})$ defined on $\mathcal{X}$ that is used in the process of defining a classifier $\underline{\boldsymbol{D}}$. When a discriminant function is used to describe a given class $\boldsymbol{\omega}^{(k)}$, then the superscript notation will be used $\underline{\boldsymbol{d}}^{(k)}(\boldsymbol{x})$.

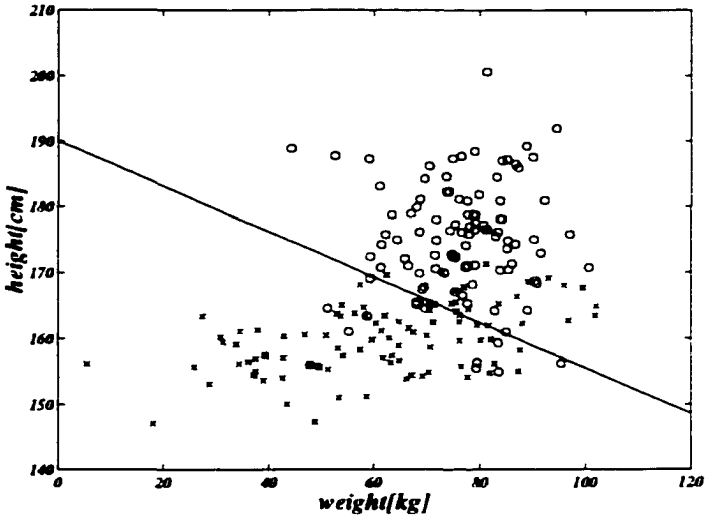

Figure 2.4. *The hypothetical distribution of weight and height for males and females. The 'o' represent males while the '×' represent females. The line drawn is given by the equation* $height[cm] = -0.34 \cdot weight[kg] + 190$, *which shows the decision boundary for classifying a person as male or female based on their height and weight.*

2.2.3 Bayes Classifier

The female-male classification example in section 2.2.2.1 raises the important question of whether the classifier that was designed by eye (see equations (2.5) and (2.6)) is the best possible classifier for the problem. Perhaps the slope should be slightly smaller, or the zero point should be larger. Perhaps a quadratic boundary should be used between the classes rather than a linear one. Perhaps more than one decision boundary should be used. This section summarizes the theoretical framework that allows these questions to be answered. A more thorough treatment can be found found in several textbooks [34, 38, 123].

Consider a K class pattern recognition problem with feature vectors $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^M$, where each of the classes are denoted by the symbol $\omega^{(k)}$ for $k = 1, \ldots, K$. Associated with the feature space $\mathcal{X}$, and each class $\omega^{(k)}$ are four fundamental quantities which are presented in the following definitions. A fundamental assumption in pattern recognition is that these four quantities are time invariant.

Definition 1 ***a priori probability*** $p(\omega^{(k)})$

The a priori probability $p(\omega^{(k)})$ *of a given class is a scalar quantity defining the probability of occurrence of class* $\omega^{(k)}$, *such that* $\sum_{k=1}^{K} p(\omega^{(k)}) = 1$.

Definition 2 ***conditional probability density function*** $p(\boldsymbol{x}|\omega^{(k)})$

The conditional probability density function $p(\boldsymbol{x}|\omega^{(k)})$ *of a given class is a scalar function defined on* $\mathcal{X}$, *which gives the probability that a random feature vector from class* $\omega^{(k)}$ *will have a value* $\boldsymbol{x}$. *As is mandatory for probability density functions,* $\int_{\mathcal{X}} p(\boldsymbol{x}|\omega^{(k)})d\boldsymbol{x} = 1$.

Definition 3 ***a posteriori probability*** $p(\omega^{(k)}|\boldsymbol{x})$

The a posteriori probability $p(\omega^{(k)}|\boldsymbol{x})$ *is a scalar function defined on* $\mathcal{X}$ *that gives the probability that a given feature vector* $\boldsymbol{x}$ *belongs to the class* $\omega^{(k)}$, *such that for any given* $\boldsymbol{x}$, $\sum_{k=1}^{K} p(\omega^{(k)}|\boldsymbol{x}) = 1$.

Definition 4 ***unconditional probability density function*** $p(\boldsymbol{x})$

The unconditional probability density function $p(\boldsymbol{x})$ *is a scalar function defined on* $\mathbb{R}^M$, *which gives the probability that a random feature vector from any class* $\omega^{(k)}$ *will have a value* $\boldsymbol{x}$. *This function can be computed as*

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} p(\omega^{(k)})p(\boldsymbol{x}|\omega^{(k)}). \tag{2.7}$$

As is mandatory for probability density functions, $\int_{\mathcal{X}} p(\boldsymbol{x})d\boldsymbol{x} = 1$, *which is enforced by equation* (2.7).

For the pattern recognition problem, a feature vector $\boldsymbol{x}$ is given as input, and a class $\omega^{(k)}$ is desired as an output. Clearly, the *a posteriori* probability $p(\omega^{(k)}|\boldsymbol{x})$ is the quantity that is desired to solve this problem, but sometimes, it is easier to estimate the other three quantities and compute $p(\omega^{(k)}|\boldsymbol{x})$ using Bayes rule

$$p(\omega^{(k)}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega^{(k)})p(\omega^{(k)})}{p(\boldsymbol{x})} = \frac{p(\boldsymbol{x}|\omega^{(k)})p(\omega^{(k)})}{\sum_{l=1}^{K} p(\omega^{(l)})p(\boldsymbol{x}|\omega^{(l)})}. \tag{2.8}$$

This relationship is used extensively in pattern recognition.

2.2.3.1 Bayes Classifier for Minimum Error

Using the *a posteriori* probability $p(\omega^{(k)}|\boldsymbol{x})$ defined in definition 3, or computed using equation (2.8), it is possible to define the Bayes classifier for minimum error,

$$\underline{\boldsymbol{D}}(\boldsymbol{x}) = \omega^{(k)} \text{ if } p(\omega^{(k)}|\boldsymbol{x}) > p(\omega^{(j)}|\boldsymbol{x}) \quad \forall \quad j \neq k, \tag{2.9}$$

where ties are resolved arbitrarily. If a feature vector $\boldsymbol{x}$ is classified using the Bayes classifier for minimum error, then the probability that it is assigned to the correct class is given by $\max_k p(\omega^{(k)}|\boldsymbol{x})$, and the probability of making an error is given by

$$e_B(\boldsymbol{x}) = 1 - \max_k p(\omega^{(k)}|\boldsymbol{x}). \tag{2.10}$$

Integrating over the feature space $\mathcal{X}$ defines the average Bayes error rate for the problem

$$E_B = \int_{\mathcal{X}} e_B(\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}. \tag{2.11}$$

It is important to recognize that this is the absolute lower limit on the average error rate, so no other classifier can do better.

2.2.3.2 Bayes Classifier for Minimum Risk

An important generalization of the Bayes classifier for minimum error is to assign a loss value L_{jk} when the classifier assigns a feature vector to $\omega^{(k)}$ while the correct class was $\omega^{(j)}$. L can therefore be represented as a $K \times K$ matrix, where in general, the diagonal (representing correct decisions by the classifier) contains zeros, and the off diagonals contain values ≥ 0. The risk associated with assigning $\boldsymbol{x}$ to $\omega^{(k)}$ is then given by

$$r_(\omega^{(k)}|\boldsymbol{x}) = \sum_{j=1}^{K} L_{jk} p(\omega^{(j)}|\boldsymbol{x}). \tag{2.12}$$

and the Bayes classifier for minimum risk is defined as

$$\underline{\boldsymbol{D}}(\boldsymbol{x}) = \omega^{(k)} \text{ if } r_(\omega^{(k)}|\boldsymbol{x}) < r_(\omega^{(j)}|\boldsymbol{x}) \quad \forall \quad j \neq k, \tag{2.13}$$

where ties are resolved arbitrarily. If a feature vector $\boldsymbol{x}$ is classified using the Bayes classifier for minimum risk, then the risk of assigning it to a wrong class is given by

$$r_B(\boldsymbol{x}) = \min_k r_(\boldsymbol{\omega}^{(k)}|\boldsymbol{x}). \tag{2.14}$$

Integrating over the feature space $\mathcal{X}$ defines the average Bayes risk for the problem,

$$R_B = \int_{\mathcal{X}} r_B(\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}. \tag{2.15}$$

This is the absolute lower limit on the average risk, so no other classifier can do better. Note that if $L_{jk} = 1 - \delta[j-k]$, then zero loss is incurred for correct classification, and a constant loss of one is incurred for any error that is committed. In this case, the risk is equivalent to the error rate, and the Bayes classifier for minimum risk is identical to the Bayes classifier for minimum error. This loss function will be referred to as the canonical loss function, and unless otherwise stated, it will be used exclusively in this thesis. Both classifiers will simply be referred to as the Bayes classifier.

2.2.3.3 Example: Female-Male Bayes Classifier

The following example should illuminate the main concepts that were introduced in this section. The data that was generated in section 2.2.2.1 was produced using equal *a priori* probabilities $p(\boldsymbol{\omega}^{(1)}) = p(\boldsymbol{\omega}^{(2)}) = 0.5$ and a Normal probability density function (*pdf*)

$$p(\boldsymbol{x}|\boldsymbol{\omega}^{(k)}) = (2\pi)^{-m/2} |\boldsymbol{\Sigma}^{(k)}|^{-1/2} \exp\left[-1/2(\boldsymbol{x} - \boldsymbol{\mu}^{(k)})^T (\boldsymbol{\Sigma}^{(k)})^{-1} (\boldsymbol{x} - \boldsymbol{\mu}^{(k)})\right], \tag{2.16}$$

where $m = 2$ is the dimensionality of the problem, the feature space has physical quantities defined in equation (2.3), and $\boldsymbol{\mu}^{(k)}$ and $\boldsymbol{\Sigma}^{(k)}$ are the mean vector and covariance matrix for the class $\boldsymbol{\omega}^{(k)}$ with numerical values

$$\boldsymbol{\mu}^{(1)} = \begin{bmatrix} 75 \\ 175 \end{bmatrix} \qquad \boldsymbol{\Sigma}^{(1)} = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix} \tag{2.17}$$

$$\boldsymbol{\mu}^{(2)} = \begin{bmatrix} 60 \\ 160 \end{bmatrix} \qquad \boldsymbol{\Sigma}^{(2)} = \begin{bmatrix} 400 & 50 \\ 50 & 25 \end{bmatrix}. \tag{2.18}$$

Therefore, it is possible to compute $p(\boldsymbol{x})$ using equation (2.7), the *a posteriori* probabilities $p(\boldsymbol{\omega}^{(k)}|\boldsymbol{x})$ using Bayes rule given in equation (2.8), and the Bayes error

rate e_B using equation (2.10). A graphical interpretation of these quantities is plotted in figure 2.5 which should provide a more intuitive feeling for the concepts behind Bayesian classifiers. The plots in the left column (*a*, *b*, and *c*) show the *pdfs* of the individual classes, and the combination of the classes, whereas the plots in the right column (*d*, *e*, and *f*) show probabilities that each $\boldsymbol{x}$ comes from each class, as well as the expected error for each $\boldsymbol{x}$. As expected, the highest probability of error occurs where the probability of belonging to each class is approximately equal. The average Bayes error rate for this problem can be computed by numerically integrating the surface plotted in figure 2.5(*f*) times the surface plotted in figure 2.5(*c*) according to equation (2.11), which gives $\boldsymbol{E_B} = 0.13$.

By taking $-\ln p(\boldsymbol{\omega}^{(k)}|\boldsymbol{x})$ with the Normal *pdf* given in equation (2.16) inserted into Bayes Rule, a discriminant function can be formed for each class[5]

$$\underline{\boldsymbol{d}}^{(k)}(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{\mu}^{(k)})^T(\boldsymbol{\Sigma}^{(k)})^{-1}(\boldsymbol{x} - \boldsymbol{\mu}^{(k)}) + \ln|\boldsymbol{\Sigma}^{(k)}| - 2\ln p(\boldsymbol{\omega}^{(k)}), \tag{2.19}$$

where the first term on the right is the familiar Mahalanobis distance between $\boldsymbol{x}$ and $\boldsymbol{\mu}^{(k)}$, the second term is an adjustment for the size of the covariance matrix, and the last term is an adjustment for the class *a priori* probability. Equation (2.19) is often referred to as the Bayes distance, but it should be remembered that it is actually a squared metric (*like* the Mahalanobis distance) that can take on negative values (*unlike* the Mahalanobis distance). Since the $-\ln$ operation is a strictly monotonically decreasing function, $\min_k \underline{\boldsymbol{d}}^{(k)}(\boldsymbol{x}) = \max_k p(\boldsymbol{\omega}^{(k)}|\boldsymbol{x})$ and the Bayes classifier for this problem (shown in the generalized form for more than two classes) can be defined as

$$\underline{\boldsymbol{D}}(\boldsymbol{x}) = \boldsymbol{\omega}^{(k)} \text{ if } \underline{\boldsymbol{d}}^{(k)}(\boldsymbol{x}) < \underline{\boldsymbol{d}}^{(j)}(\boldsymbol{x}) \quad \forall \quad j \neq k, \tag{2.20}$$

with ties resolved arbitrarily. The decision boundary between the classes is obtained by finding the quadratic solution of $\underline{\boldsymbol{d}}^{(1)}(\boldsymbol{x}) = \underline{\boldsymbol{d}}^{(2)}(\boldsymbol{x})$, which is plotted in figure 2.6 with the 1σ level of the *pdf* for each class, and the original 'by-eye' estimate given in section 2.2.2.1. So to answer the original question in this section, the 'by-eye' linear classifier is *not* the optimal classifier for this problem.

[5]The discriminant function could be defined to be the $p(\boldsymbol{\omega}^{(k)}|\boldsymbol{x})$ itself, but taking $-\ln p(\boldsymbol{\omega}^{(k)}|\boldsymbol{x})$ makes the math easier, and allows us to think in terms of Bayesian distances. Additionally, it avoids having to compute the exp function which is important in real-time applications. Notice that the quantities that are common to all classes ($(2\pi)^{-m/2}$ and $p(\boldsymbol{x})$) are dropped from the equation since they do not affect the discrimination between classes.

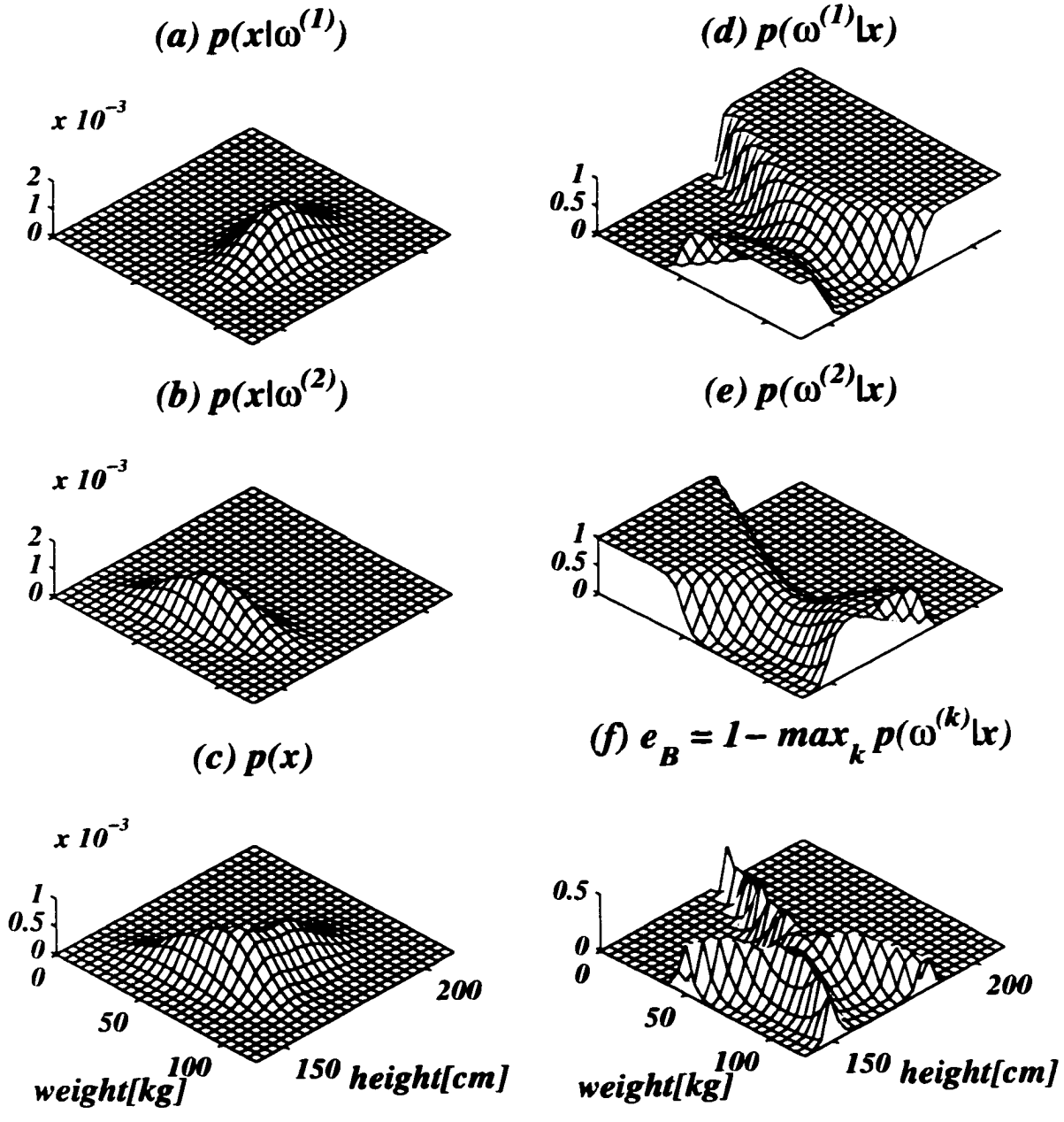

Figure 2.5. *The hypothetical distribution of weight and height for* $\omega^{(1)}$ = *male and* $\omega^{(2)}$ = *female. This plot shows the fundamental Bayesian quantities for this problem.*

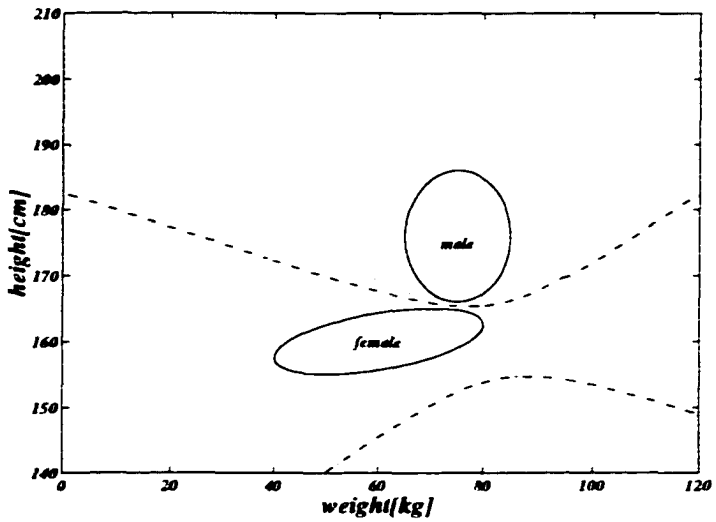

Figure 2.6. *The hypothetical distribution of weight and height for* $\omega^{(1)}$ *= males and* $\omega^{(2)}$ *= females. The solid lines show the* 1σ *level of the pdf for each class (i.e.,* $p(\boldsymbol{x}|\omega^{(1)})$ *and* $p(\boldsymbol{x}|\omega^{(2)})$*), the dot-dashed line shows the Bayes boundary between the classes that results in a minimal average error rate, and the dotted line shows the initial 'by-eye' guess which is also plotted in figure 2.4.*

2.2.4 Classifier Design

For most pattern recognition problems, the Bayes classifier is not known *a priori*, and the classifier must be designed using a finite set of labelled data $\mathcal{L}_t = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$. Designing a classifier is therefore a problem of 'learning from data' as described by Cherkassky and Mulier [16]. There are a variety of ways in which classifier design methods can be categorized (*e.g.* see Lippman [75]). The approach taken here is to categorize them by the property of the classifier that is being estimated in the design process, for which there are roughly three categories as described below.

2.2.4.1 Method 1: Estimate $p(\boldsymbol{x}|\omega^{(k)})$

In this approach, the conditional *pdfs* $p(\boldsymbol{x}|\omega^{(k)})$ are estimated for each class $\omega^{(k)}$ using either a parametric or a non-parametric approach. The most common example of using the parametric approach is to assume that $p(\boldsymbol{x}|\omega^{(k)})$ is Normally distributed, and to estimate the parameters using standard statistical estimation [43]. This ap-

proach results in quadratic decision surfaces, and the resulting classifiers are often called quadratic Gaussian classifiers. When there are not enough training samples to estimate the covariance matrices of each class individually, then it is often assumed that all the classes have the same covariance (so the covariance matrix can be estimated using the training samples from all the classes) and different means. This approach results in linear decision surfaces and the resulting classifiers are often called linear Gaussian classifiers. The issues behind matching the classifier complexity to the training set size are discussed more thoroughly in section 2.2.5.

Examples of the non-parametric approach of estimating $p(\boldsymbol{x}|\boldsymbol{\omega}^{(k)})$ are given by the method of potentials or Parzen windows [38, 123], which are re-emerging today under the new name of 'radial basis function neural networks' [75, 86]. In both cases, the *a posteriori* probabilities $p(\boldsymbol{\omega}^{(k)}|\boldsymbol{x})$ are then computed using Bayes rule (equation (2.8)), and the classifier is defined using the Bayes classifier (equation (2.9) or equation (2.13)).

2.2.4.2 Method 2: Estimate $p(\boldsymbol{\omega}^{(k)}|\boldsymbol{x})$

In this approach, the *a posteriori* probabilities $p(\boldsymbol{\omega}^{(k)}|\boldsymbol{x})$ are estimated directly from the data. This can be done using a nearest neighbour rule [28], where the intuitive concept that the *a posteriori* probability of a pattern $\boldsymbol{x}$ belonging to class $\boldsymbol{\omega}^{(k)}$ is proportional to the 'nearness' of $\boldsymbol{x}$ to previously observed patterns from class $\boldsymbol{\omega}^{(k)}$. In addition, a neural network pattern classifier can be viewed as a method for estimating the *a posteriori* probabilities of each class [105]. In both cases, the classifier is defined using the Bayes classifier (equation (2.9) or equation (2.13)).

2.2.4.3 Method 3: Estimate Decision Boundaries

In this approach, the decision boundaries between the classes are estimated directly from the learning set $\mathcal{L}_t$. This can be done in a heuristic manner as described in section 2.2.4.4, by using a recursive partition of the feature space as is done in CART [9], or through an optimization technique that minimizes the empirical risk of misclassification as is done for vector support machines (VSM) [16, 127] .

2.2.4.4 Fisher's LDA

Obviously, there are a vast number of different approaches to designing a classifier, but since the focus of this thesis is on the feature extraction part of designing an acoustic pattern recognition system, a single classifier was chosen to evaluate and compare the different feature extraction techniques developed in chapter 4. Fisher's linear discriminant analysis (LDA) [42] was chosen based on its intuitive and simple underlying principle, and because it is one of the most widely used classification methods with an excellent track record.

Fisher's LDA is a time-honored tool for both classification and dimensionality reduction which was designed for the two class discrimination problem and was later generalized to the multiple class discrimination problem by Bryan [12]. Although the approach is sometimes justified by assuming Normal distributions for the class conditional *pdfs* $p(\boldsymbol{x}|\boldsymbol{\omega}^{(k)})$, Fisher's motivation was heuristic in nature, and thus it is presented here as a heuristic method for finding the decision boundaries between classes.

Given a training set $\mathcal{L}_t = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ having K classes $\{\boldsymbol{\omega}^{(k)}\}_{k=1}^{K}$ with *a priori* probabilities $p(\boldsymbol{\omega}^{(k)})$, the patterns $\boldsymbol{x}_i$ belonging to class $y_i = \boldsymbol{\omega}^{(k)}$ will be denoted as $\mathcal{X}^{(k)} = \{\boldsymbol{x}_i^{(k)}\}_{i=1}^{N^{(k)}}$. Assume that the feature vectors are M-dimensional $\boldsymbol{x}_i \in \mathbb{R}^M$. For each class, the plug-in estimate for the mean is defined by

$$\boldsymbol{\mu}^{(k)} \stackrel{\text{def}}{=} \frac{1}{N^{(k)}} \sum_i \boldsymbol{x}_i^{(k)}, \tag{2.21}$$

the plug-in estimate for the correlation matrix is defined by[6]

$$\boldsymbol{Q}^{(k)} \stackrel{\text{def}}{=} \frac{1}{N^{(k)}} \sum_i \boldsymbol{x}_i^{(k)} \boldsymbol{x}_i^{(k)T}, \tag{2.22}$$

and the plug-in estimate for the covariance matrix is defined by[7]

$$\boldsymbol{\Sigma}^{(k)} \stackrel{\text{def}}{=} \boldsymbol{Q}^{(k)} - \boldsymbol{\mu}^{(k)} \boldsymbol{\mu}^{(k)T}. \tag{2.23}$$

[6]This definition is a deviation from the multivariate statistics literature, where the correlation matrix usually means the matrix of correlation coefficients.

[7]Often the covariance estimate is scaled by a factor $N/(N-1)$ to remove bias, but the importance of this correction from a statistical estimation perspective is often over emphasized [43].

For the whole training set, the mean is defined by

$$\boldsymbol{\mu} \stackrel{\text{def}}{=} \sum_k p(\omega^{(k)})\boldsymbol{\mu}^{(k)}, \tag{2.24}$$

the correlation matrix is defined by

$$\boldsymbol{Q} \stackrel{\text{def}}{=} \sum_k p(\omega^{(k)})\boldsymbol{Q}^{(k)}, \tag{2.25}$$

the within-class covariance is defined by

$$\boldsymbol{\Sigma}_W \stackrel{\text{def}}{=} \sum_k p(\omega^{(k)})\boldsymbol{\Sigma}^{(k)} \tag{2.26}$$

$$= \boldsymbol{Q} - \sum_k p(\omega^{(k)})\boldsymbol{\mu}^{(k)}\boldsymbol{\mu}^{(k)T}, \tag{2.27}$$

the between-class covariance is defined by

$$\boldsymbol{\Sigma}_B \stackrel{\text{def}}{=} \sum_k p(\omega^{(k)})(\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu})(\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu})^T \tag{2.28}$$

$$= \sum_k p(\omega^{(k)})\boldsymbol{\mu}^{(k)}\boldsymbol{\mu}^{(k)T} - \boldsymbol{\mu}\boldsymbol{\mu}^T, \tag{2.29}$$

and the total covariance is defined by

$$\boldsymbol{\Sigma}_T \stackrel{\text{def}}{=} \boldsymbol{\Sigma}_W + \boldsymbol{\Sigma}_B \tag{2.30}$$

$$= \boldsymbol{Q} - \boldsymbol{\mu}\boldsymbol{\mu}^T. \tag{2.31}$$

If the *a priori* probabilities are estimated as the fraction of samples from each class in the training set $p(\omega^{(k)}) = N^{(k)}/N$, then it is easy to show that $\boldsymbol{\mu}$, $\boldsymbol{Q}$ and $\boldsymbol{\Sigma}_T$ are the plug-in estimates for the mean, correlation matrix, and covariance matrix of the entire data set. This is significant since it implies that the total covariance of the data set can be divided into the covariance from the within-class scatter $\boldsymbol{\Sigma}_W$ and the covariance from the between class scatter $\boldsymbol{\Sigma}_B$.

Fisher's idea was to apply a transformation $\tilde{\boldsymbol{x}}_i = \boldsymbol{A}^T\boldsymbol{x}_i$ that maximizes the ratio of the between-class scatter to the within-class scatter. It is not hard to show that this transformation results in a new within-class covariance matrix given by

$$\tilde{\boldsymbol{\Sigma}}_W = \boldsymbol{A}^T\boldsymbol{\Sigma}_W\boldsymbol{A}, \tag{2.32}$$

and a new between-class covariance matrix given by

$$\tilde{\boldsymbol{\Sigma}}_B = \boldsymbol{A}^T \boldsymbol{\Sigma}_B \boldsymbol{A}. \tag{2.33}$$

Therefore, the criterion that Fisher maximized is given by

$$J(\boldsymbol{A}) = \frac{|\boldsymbol{A}^T \boldsymbol{\Sigma}_B \boldsymbol{A}|}{|\boldsymbol{A}^T \boldsymbol{\Sigma}_W \boldsymbol{A}|}, \tag{2.34}$$

where $|\boldsymbol{H}|$ indicates the determinant of the matrix $\boldsymbol{H}$. Recall that the determinant of a matrix is the product of the eigenvalues and hence for covariance matrices is the product of the variances in the principal directions. Therefore, if the determinant of the covariance matrix for a data set is increased, then the scatter of the data set must increase. Other measures of the scatter of the data could also have been used, such as the trace or 2-norm of the covariance matrix, but the use of the determinant results in an analytic solution that maximizes Fisher's criterion [38].

The columns of the matrix $\boldsymbol{A}$ that maximize equation (2.34) are given by the generalized eigenvectors that correspond to the largest eigenvalues of[8]

$$\boldsymbol{\Sigma}_B \boldsymbol{a}_i = \lambda_i \boldsymbol{\Sigma}_W \boldsymbol{a}_i. \tag{2.35}$$

If the within-class covariance is invertible, then the generalized eigenvalue problem can be converted to a regular eignenvalue problem by multiplying both sides by $\boldsymbol{\Sigma}_W^{-1}$, but this is unnecessary since techniques are readily available for solving the generalized eigenvalue problem without having to invert $\boldsymbol{\Sigma}_W$ [129].

The canonical variates of the discrimination problem are the $p = \min(M, K-1)$ eigenvectors of $\boldsymbol{A}$ corresponding to the largest eigenvalues which span a p-dimensional subspace of $\mathbb{R}^M$ containing the mean vectors from each class $\boldsymbol{\mu}^{(k)}$. If the covariance matrices of all the classes are identical, then this subspace contains all the discriminatory power of the discrimination problem and the remaining $M - p$ eigenvectors can be discarded. This is the feature reduction part of Fisher's LDA.

The transformation $\tilde{\boldsymbol{x}}_i = \boldsymbol{A}^T \boldsymbol{x}_i$ also re-scales the within-class covariance so that $\tilde{\boldsymbol{\Sigma}}_W$ is isotropic. This means that the squared Euclidean distance can be used to define a discriminant function for each class

$$\underline{d}^{(k)}(\tilde{\boldsymbol{x}}) = (\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{\mu}}^{(k)})^T (\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{\mu}}^{(k)}), \tag{2.36}$$

[8]The generalized eigenvalue problem is also called the pencil eigenvalue problem.

where $\tilde{\boldsymbol{\mu}}^{(k)} = \boldsymbol{A}^T \boldsymbol{\mu}^{(k)}$. A classifier can then be defined as

$$\underline{\boldsymbol{D}}(\tilde{\boldsymbol{x}}) = \boldsymbol{\omega}^{(k)} \text{ if } \underline{\boldsymbol{d}}^{(k)} < \underline{\boldsymbol{d}}^{(j)} \quad \forall \quad j \neq k, \tag{2.37}$$

where ties are resolved arbitrarily. The decision boundaries for this classifier are linear both in the original space and in the transformed space. This is the classification part of Fisher's LDA. When all the classes are Normally distributed with identical covariance matrices and different mean vectors, Fisher's heuristic classifier is also the Bayes classifier [16].

2.2.5 Learning From Data

A central issue when designing a classifier is that the properties of the classifier must in general be estimated from a finite set of labelled training data $\mathcal{L}_t = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$. In theory, the more flexible (*i.e.*, complex) a classifier is, the lower its classification error should be. This assumes that the properties for the classifier have been chosen optimally for the problem. Often the estimation errors that result from designing a classifier with finite data outweigh the potential advantage of the classifier's flexibility.

For example, it is known that the male-female dataset described in sections 2.2.2.1 and 2.2.3.3 was created using Normal distributions with different means and covariance matrices. However, in an experiment carried out 100 times, where a quadratic Gaussian classifier and linear Gaussian classifier were trained using 3 data samples from each class, the median error rate from an independent data set of 1000 samples from each class is 0.19 and 0.31 for the linear and quadratic Gaussian classifier respectively. The linear classifier outperformed the quadratic classifier in 71 % of the trials. This shows that even though it is known that the data have Normal distributions with *different* covariance matrices, lower classification errors are obtained by assuming that both classes have the *same* covariance matrix, since this reduces the estimation errors of the covariance matrices.

Matching the complexity of the classifier is thus very important but may seem trivial from the above example since there are almost always more than 3 samples from each class for training. However, as the dimensionality of the problem M increases the number of parameters that must be estimated for a covariance matrix increases as $M(M+1)/2$, so the sample size requirements quickly escalate. Other classifiers, such

as those that estimate the conditional *pdfs* $p(\boldsymbol{x}|\boldsymbol{\omega}^{(k)})$ of the classes non-parametrically are even more sensitive to dimensionality. In fact, the problem is so pervasive in all estimation tasks including classification, regression and density estimation, that it has earned the name 'curse of dimensionality', as coined by Bellman [5] and elaborated on by several authors [16, 34, 38, 75].

2.2.5.1 Curse of Dimensionality

The curse of dimensionality is a result of the geometry of high dimensional spaces. It is called a curse, because humans generally think in low dimensions (≤ 3 dimensions), and thus often have false intuition about how things should work in high dimensional spaces. The most illustrative example is given by looking at the density of samples in different dimensional spaces. If N samples are distributed uniformly on the interval $[0, 1]$ in $\mathbb{R}^1$, then N^M samples would be required to achieve the same density in $\mathbb{R}^M$ on the unit hypercube. In other words, sample sizes that are densely distributed in low dimensional space become sparsely distributed in high dimensional spaces. Friedman [46] gives an excellent summary of some other properties of distributions in high dimensional spaces. The effect that the curse of dimensionality has on classifier design depends on the type of classifier, but in general the designer can either control the complexity of the classifier, or control the dimensionality of the feature space to reduce the effects of the curse.

2.2.5.2 Control Classifier Complexity

This approach of suppressing the curse of dimensionality assumes that the dimensionality of the feature space is fixed, and thus tries to limit the complexity of the classifier so that estimation errors do not dominate the result. For some classifiers, their complexity can be defined formally in terms of the Vapnik-Chervonenkis (VC) dimension as defined in statistical learning theory [16, 127], but for our discussion classifier complexity will loosely describe the flexibility of the classifier to classify data. For parametric classifiers, controlling classifier complexity amounts to controlling the number of independent parameters that must be estimated, or controlling the range of values that the parameters can take on. For non-parametric classifiers, controlling complexity amounts to smoothing the non-parametric estimates (such as

densities or class boundaries).

An example of controlling complexity is given by Friedman's regularized discriminant analysis [45], which incorporates a Euclidean, linear Gaussian, and quadratic Gaussian classifier (listed in order of increasing complexity). The adoption of a linear classifier over a quadratic classifier allows the samples from all classes to be used to estimate a covariance matrix, and when the total number of samples is not even large enough to estimate a single covariance matrix, then the covariance matrix can be taken to be the identity matrix. A weighted average of the three covariance matrices is adopted, and cross-validation is used to choose the best weights.

Another example is given by Hastie et al.'s Penalized discriminant analysis (PDA) [58], which is a regularized version of Fisher's LDA that is applicable in high dimensional settings with small sample sizes. Their technique uses a penalized version of the within-class covariance matrix $\tilde{\mathbf{\Sigma}}_W = \mathbf{\Sigma}_W + \Omega$, where Ω is a penalty matrix that imposes smoothness on the resultant canonical variates. In other words, they control the complexity of the classifier by limiting the within-class covariance matrix to have values that they assume are more appropriate for the problem than the estimated values. They use a cross-validation method to choose the amount of smoothing that is required for a given problem.

The two methods described above are regularization methods that attempt to reduce the variance in the estimates of the classifier properties by biasing them away from the sample-based estimates in a direction that is assumed to be reasonable for the problem. If the assumption is valid, then the variance of the property estimates can be drastically reduced while keeping the bias low. However, if the assumption is not valid, then the bias introduced into the property estimates can far outweigh the decrease in variance. Other methods of controlling classifier complexity which can be thought of more generally as methods for model selection are Rissanen's minimum description length principle [106] and Vapnik's structural risk minimization [127].

2.2.5.3 Control Feature Space Dimensionality

The other method of combating the curse of dimensionality is to assume that the complexity of the classifier is fixed and attempt to reduce the dimensionality of the problem by using some kind of feature extraction algorithm. A classifier is a math-

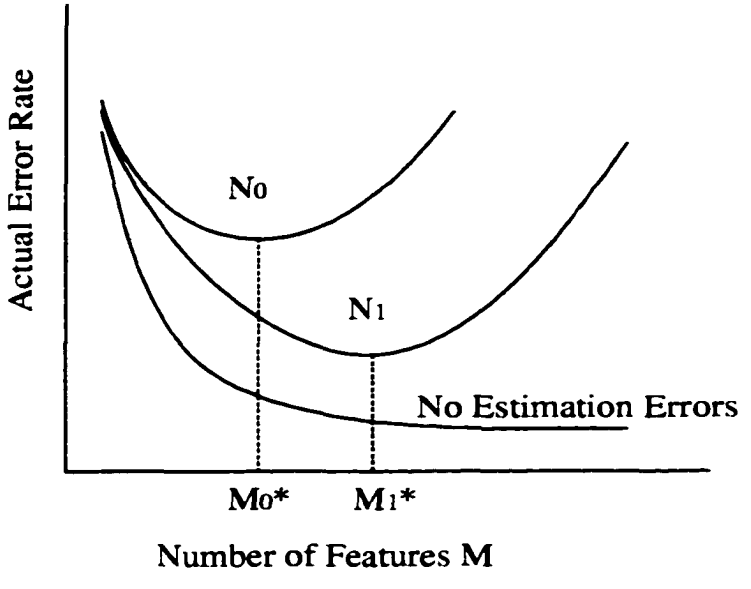

Figure 2.7. *The bottom line shows the actual error rate of a hypothetical classifer if there are no estimation errors. The top and middle lines show the actual error rates for a classifier that is trained with N_0 and N_1 samples respectively, where $N_1 > N_0$.*

ematical tool that has many internal parameters that must in general be estimated from a finite set of N training samples. When N is small, the estimation errors of the internal parameters increase the actual error rate of the classifier compared to the same classifier with no estimation errors[9]. As the number of features M increases, the number of internal parameters in the classifier typically increases at an exponential rate and the effect of estimation errors on the actual error rate is compounded. These effects are shown graphically in figure 2.7 for a hypothetical example.

As can be seen, when there are no estimation errors, the actual error rate of the classifier decreases monotonically as the number of features M increases. However, when the classifier is trained with a finite number of samples, there is a minimum

[9]The actual error rate is the expected value of the error rate of the classifier when used to classify data that is drawn from the same distribution as the training data yet independently of the training data.

actual error rate at M_0^* and M_1^*, after which the estimation errors dominate over the advantage of adding more features and the actual error rate increases. Note that the minimum M_1^* occurs at a higher value than M_0^* since $N_1 > N_0$ and thus has smaller estimation errors.

It is now easy to see why it is important to choose a good feature set for a given problem when there are a finite number of samples to use for training. Referring to figure 2.8, assume that two feature sets, $\mathcal{F}_0$ (a 'poor' feature set) and $\mathcal{F}_1$ (a 'good' feature set) are ordered such that the actual error rate of a classifier with no estimation errors decreases monotonically[10]. $\mathcal{F}_0$ should be thought of as the input features for a problem, and $\mathcal{F}_1$ should be thought of as linear transformation of the features in $\mathcal{F}_0$. Strictly speaking the hypothetical classifier should be affine invariant like Fisher's LDA[11].

When there are no estimation errors, the 'good' feature set shows a rapid decrease in actual error rate initially as M increases, and then levels off as it approaches M_{max}, while the 'poor' feature set shows a gradual decrease in the actual error rate over the whole range of M. Clearly, if there are no estimation errors, then by keeping M_{max} features, the same actual error rate can be achieved by both feature sets. However, if the classifier is trained with a finite number of samples, then the estimation errors cause the minimum actual error rate to occur at $M^* < M_{max}$, and a significant reduction in the actual error rate is obtained by using the 'good' feature set $\mathcal{F}_1$.

Controlling the dimensionality of the feature space is the approach taken in this thesis to reduce the effects of the curse of dimensionality; a detailed discussion of the different types of feature extraction methods is presented in section 4.3.

[10]This is an admittedly simplified view of a feature set since it does not describe the interactions between features, but for the purposes of this discussion, it will suffice. In any case, the patterns described here are observed in the synthetic and recorded experiments of chapters 5 and 6, so this presentation is valid in that regard.

[11]An affine invariant classifier will produce the same classification results if the input features are subject to a linear transformation without reducing the dimensionality. This accounts for the reason that the the two feature sets have the same actual error rate when $M = M_{max}$.

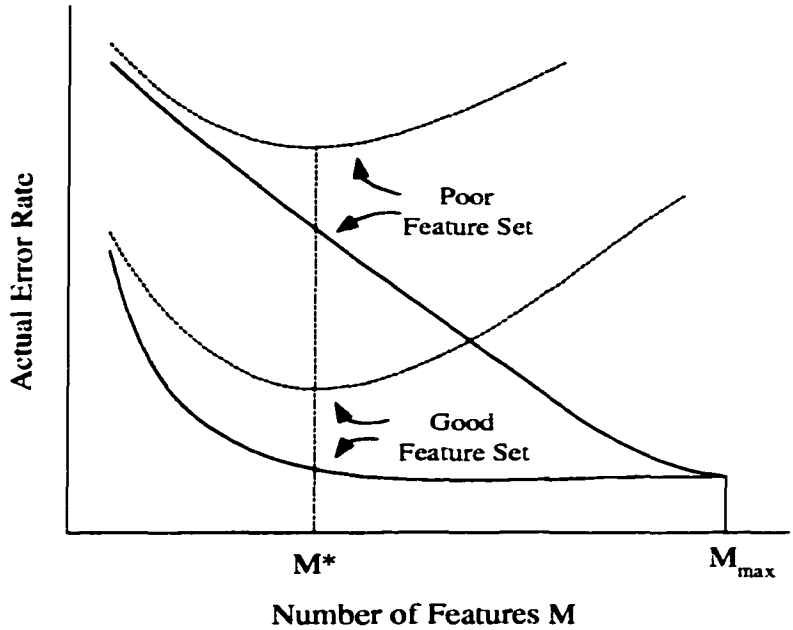

Figure 2.8. *The dotted and solid lines show the actual error rates for a hypothetical classifier with and without estimation errors respectively for a 'good' feature set* $\mathcal{F}_1$ *and a 'poor' feature set* $\mathcal{F}_0$. M_{max} *represents the maximum number of features.*

2.2.6 Performance Estimation

Estimating the performance of a classifier in an unbiased manner is important for both

Classifier Selection - Often, several classifiers or different variable settings in the same classifier need to be compared so that the 'best' classifier can be chosen for a given application.

Interpreting Classification Results - When a classifier is used for a given application, it is usually important to know the probability of the classifier making an error. This knowledge often affects how a system responds to a given classification result.

Classifiers are usually evaluated in terms of their expected average error rate as presented here, or in terms of the expected average risk of misclassification. A classifier $\underline{\boldsymbol{D}}$ is evaluated by classifying a labelled evaluation set of data $\mathcal{L}_e = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$

and computing the fraction of errors that were made

$$\boldsymbol{E}_{est} = \frac{\sum_{i=1}^{N} \mathcal{I}(y_i \neq \underline{\boldsymbol{D}}(\boldsymbol{x}_i))}{N}, \tag{2.38}$$

where $\mathcal{I}(y_i \neq \underline{\boldsymbol{D}}(\boldsymbol{x}_i))$ is an indicator function that returns a one if y_i and $\underline{\boldsymbol{D}}(x_i)$ are not equal, and a zero otherwise.

It is well known that if data that is used to train a classifier is also used to evaluate it, then the estimated average error rate will be biased lower than the true average error rate. Since there is often only a finite set of labelled data given by $\mathcal{L} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$, which must be used for both a training set $\mathcal{L}_t = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N_t} \subset \mathcal{L}$ and an evaluation set $\mathcal{L}_e = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N_e} \subset \mathcal{L}$, it is important to consider the relationship between $\mathcal{L}_t$ and $\mathcal{L}_e$, where ideally $\mathcal{L}_t \cap \mathcal{L}_e = \emptyset$.

It is desirable to use as many samples as possible for training and for evaluation since this lowers the bias and variance of the estimate of the average error rate respectively. The performance estimation methods for classifiers reviewed here are also described by Devijver and Kittler [34] and Breiman et al. [9].

2.2.6.1 Resubstitution

The entire labelled data set is used for both training and evaluation, $\mathcal{L}_t = \mathcal{L}_e = \mathcal{L}$, which allows the maximum number of samples to be used for both training and evaluation. However, due to the low bias on the estimated average error rate that results from using the same data to train and evaluate the classifier, this method of classifier performance is not highly recommended.

2.2.6.2 Holdout

The labelled data set $\mathcal{L}$ is divided into $\mathcal{L}_t$ for training and $\mathcal{L}_e$ for evaluation such that $\mathcal{L}_t \cap \mathcal{L}_e = \emptyset$. This method does not use the same data for training and evaluation so the estimated average error rate is not biased low. However, the number of samples used for training N_t and evaluation $N_e = N - N_t$ must be fixed, which for small N can be problematic. If N_t is chosen to be large then N_e must be small and the variance in the estimate of the average error rate will be large. On the other hand, if N_t is chosen to be small, then the classifier may not train very well, and the estimated average error rate would be biased high relative to a classifier that is trained with

all N samples. This technique is only recommended when there is an abundance of labelled data, such as when the data is created synthetically.

2.2.6.3 Cross Validation

The labelled data set $\mathcal{L}$ is partitioned into V disjoint sets $\mathcal{L}_v$. For $v = 1, 2, \ldots, V$, use an evaluation set given by $\mathcal{L}_e = \mathcal{L}_v$ and a training set given by $\mathcal{L}_t = \mathcal{L} - \mathcal{L}_v$. The estimated average error rate of the classifier is then taken to be the average of the estimates from the individual trials. In this way, each labelled data set is classified in the evaluation set once using a classifier that was trained without it. Some authors also refer to this method as 'rotation', and when the dataset is partitioned into N disjoint sets, this method is also called the 'Leave One Out' method.

The cross validation technique is an interesting way to use a large fraction of the labelled data both for training and evaluation while at the same time ensuring that data that is used to train the classifier is not used to evaluate the classifier. On the other hand, cross validation is computationally expensive. It is the recommended technique to use when there is only a small amount of labelled data.

2.3 Acoustic Pattern Recognition

So far, pattern recognition has been presented as a static process that operates on each feature vector independently. However, an acoustic signal is a continuous analog signal of pressure fluctuations as a function of time. Therefore, methods must be developed to convert the acoustic signal into a form that is more suitable for pattern recognition. At the highest level, an acoustic pattern recognition system can be viewed as a single operation which takes as input a continuous acoustic signal $s(t)$ and gives as output a continuous signal of class labels $y(t)$, as shown in figure 2.9. In practice, the acoustic signal is usually sampled at an interval $T = 1/f_s$ to produce $s(nT)$, where n is the sample number and f_s is the sampling rate of the signal in Hz. Similarly, class labels are produced at an interval $T' = 1/f_s'$ to produce $y(nT')$, where f_s' is the sampling rate of the class labels in Hz.

A more detailed look at an acoustic pattern recognition system is shown in figure 2.10 which divides the process into three main blocks. The *buffer* block groups

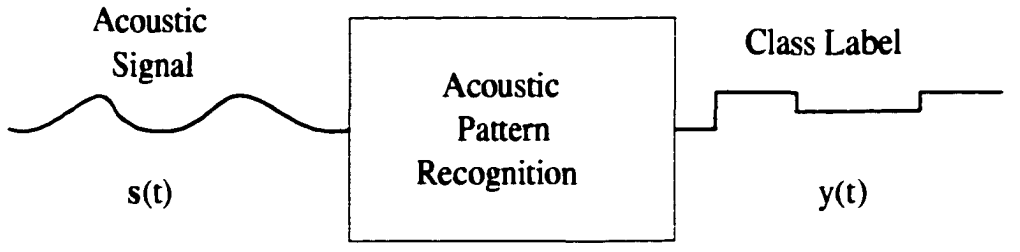

Figure 2.9. *A high level depiction of an acoustic pattern recognition system.*

the samples of $s(nT)$ into vectors $\boldsymbol{b}(nT')$ of length L with overlap d and sampling interval given by $T' = (L - d)T$. The *feature extractor* block and *classifier* block are dependent on the type of classifier that is used as discussed below.

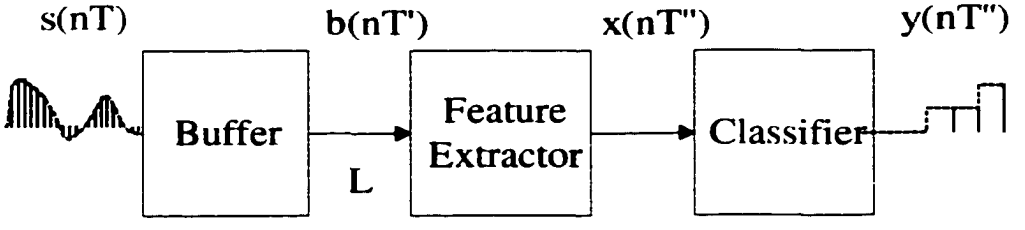

Figure 2.10. *A sampled acoustic pattern recognition system showing the three main processing blocks.*

2.3.1 Frame Classifiers

The *feature extraction* block for frame classifiers analyzes the vectors $\boldsymbol{b}(nT')$ and computes M features that are organized into vectors $\boldsymbol{x}(nT'')$, as shown in figure 2.11. The sampling interval is not changed so $T'' = T'$. Any method of classification, as discussed in section 2.2, can then be used to classify the frames individually. This is by far the most popular method of applying pattern recognition techniques to acoustic recognition and is the method adopted in this thesis. However, this method does not take into account the temporal evolution of the acoustic signal, which can be very valuable for recognition purposes.

2.3.2 Multi-Frame Classifiers

The *feature extraction* block for multi-frame classifiers analyzes the vectors $\boldsymbol{b}(nT')$ and computes M features that are organized into vectors $\tilde{\boldsymbol{x}}(nT')$. These vectors are then buffered into groups of length L' with overlap d', which gives a sampling interval of $T'' = (L' - d')T'$, as shown in figure 2.11. The feature vector $\boldsymbol{x}(nT'')$ in this case is a matrix with dimensions $M \times L'$ which can then be classified using any method of classification discussed in section 2.2. It is also possible to further process the block of $M \times L'$ features to produce a reduced set of features for the classifier as discussed in section 2.4.4.

2.3.3 Hidden Markov Models

The *feature extraction* block for hidden Markov Models (HMM) is identical to the multi-frame classifier, but the *classifier* block is significantly different. HMMs have been used very successfully for years as a technique to classify both single word [4] and continuous speech [72]. They have also been used to classify environmental acoustic transients [135]. HMMs are different from the classifiers presented in section 2.2 in that the temporal evolution of an observed sequence is modelled with state transition probabilities of an unobserved (*i.e.*, hidden) Markov chain state sequence. A review of HMMs will not be given here, but the interested reader is referred to excellent tutorials [98, 104, 103]. It should be emphasized that HMMs are just another technique for classifying a sequence of features.

2.4 Acoustic Features

2.4.1 Introduction

There has been a large variety of acoustic features used in the past for recognition purposes. Rather than try to list them all, this section provides a general outline of the signal processing techniques that have been used for extracting acoustic features and only a few references are given for typical features.

Almost all acoustic features use some sort of spectral representation of the acoustic

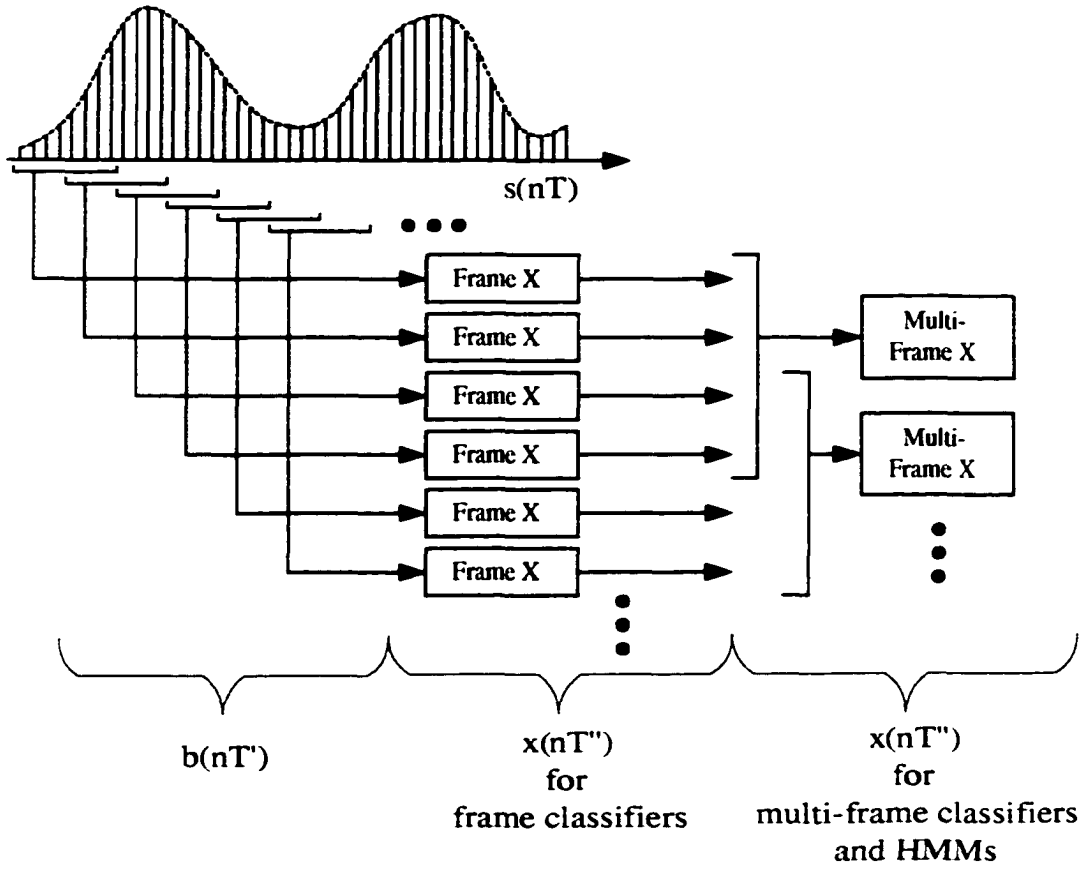

Figure 2.11. *Methods for extracting features from a sampled acoustic signal for different kinds of classifiers. The* Frame X *block is a feature extractor for a single frame or buffer* $\mathbf{b}(nT')$. *The* Multi-Frame X *block is a feature extractor for multiple frames.*

signal such as the discrete Fourier transform (DFT), filter banks or ARMA models[12]. Short time segments of the spectral representation are then analyzed to form frame level features. For multi-frame classifiers or HMMs, the frame level features are further analyzed to form multi-frame features. This section begins with a description of the most popular forms of spectral estimation (section 2.4.2), and then describes frame features (section 2.4.3), multi-frame features (section 2.4.4), and HMM features

[12]Except for perhaps the zero-crossing rate (*i.e.*, the number of times the acoustic signal crosses zero per second) which is measured in the time domain, but this too can be considered a crude form of spectral measurement since higher frequency signals have higher zero-crossing rate.

(section 2.4.5). Both the multi-frame features and HMM features require good frame level features.

2.4.2 Spectral Estimation

Although there are a variety of very sophisticated techniques for accurately estimating the spectra of a signal [50], the methods used for pattern recognition are generally straight-forward. There are three common methods for estimating spectra which are summarized below.

2.4.2.1 Discrete Fourier Transform (DFT - FFT)

The DFT of a discrete signal $x(nT)$ with length N and sampling interval T is given by

$$X(k\Delta\omega) = \sum_{n=0}^{N-1} x(nT)W^{-kn} \quad \text{for} \quad k = 0, 1, \ldots, N-1, \tag{2.39}$$

$$\text{where} \quad W = e^{j2\pi/N}, \quad \Delta\omega = \frac{\omega_s}{N}, \quad \text{and} \quad \omega_s = \frac{2\pi}{T} = 2\pi f_s$$

The basis functions $W^k(nT) = W^{kn} = e^{j2\pi kn/N}$ of the transform are complex functions representing evenly spaced vectors on the unit circle. Therefore, $X(k\Delta\omega)$ is also a complex sequence of numbers which are often written in polar form

$$X(k\Delta\omega) = A(k\Delta\omega)e^{j\phi(k\Delta\omega)}, \tag{2.40}$$

$$\text{where} \quad A(k\Delta\omega) = |X(k\Delta\omega)| = \sqrt{(\text{Re}\, X(k\Delta\omega))^2 + (\text{Im}\, X(k\Delta\omega))^2}, \tag{2.41}$$

$$\text{and} \quad \phi(k\Delta\omega) = \arg X(k\Delta\omega) = \tan^{-1} \frac{\text{Im}\, X(k\Delta\omega)}{\text{Re}\, X(k\Delta\omega)} \tag{2.42}$$

are called the amplitude spectrum and phase spectrum respectively. This transformation has complexity $O(N^2)$ but it has a fast numerical implementation called the fast Fourier transform FFT [23] which has complexity $O(N \log N)$; in order to use the fast algorithm, the signal length must be a power of two.

If $x(nT)$ is real, then $X(k\Delta\omega)$ contains $N/2$ complex conjugate pairs, which forces $A(k\Delta\omega)$ and $\phi(k\Delta\omega)$ to be symmetrical and anti-symmetrical about $\omega = \omega_s/2$ respectively. The phase is rarely used for pattern recognition purposes since the phase

of an acoustic signal changes depending on how the signal was buffered. Therefore, only the first $N/2$ components of the amplitude spectrum are usually kept for pattern recognition. This sequence will be referred to as the FFT spectrum in this thesis. See Bracewell [7] for general properties of the continuous Fourier transform and Antoniou [1] for properties of the discrete Fourier transform.

When the finite length signal $x(nT)$ is a frame from a longer signal, as is the case for acoustic pattern recognition, then $x(nT)$ is often multiplied by a window function $w(nT)$, such as the Hamming window, to reduce edge effects. Due to the product theorem of the Fourier transform [7], multiplication in the time domain by $w(nT)$ corresponds to convolution in the frequency domain by $W(k\Delta\omega)$ which both smoothes the spectral estimate of $x(nT)$ and modifies the spectral leakage caused by side lobes in $W(k\Delta\omega)$. See Antoniou [1] for a description of the various window functions that are commonly used. When a window function is applied to frames of a buffered time series, the sequence of spectral estimates is often called the short-time Fourier transform (STFT) of the signal. However, in this thesis, the spectral estimate of a frame given by the first $N/2$ samples of $A(k\Delta\omega)$ will still be referred to as the FFT spectrum. In chapter 6, the FFT spectrum is referred to as the periodogram to be consistent with the literature.

2.4.2.2 Filter Banks

A digital filter bank is nothing more than a set of digital filters, each of which is defined by its transfer function in the z-domain [1]

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_N z^{-N}}{a_0 + a_1 z^{-1} + \cdots + a_N z^{-N}}, \tag{2.43}$$

where N is the order of the filter. The filter is called a finite impulse response (FIR) filter if $a_i = 0 \quad \forall \quad i \neq 0$. Otherwise the filter is called an infinite impulse response (IIR) filter. The frequency response of a digital filter is given by the transfer function evaluated on the unit circle

$$H(e^{j\omega T}) = \frac{B(e^{j\omega T})}{A(e^{j\omega T})} = \frac{b_0 + b_1 e^{-j\omega T} + \cdots + b_N e^{-jN\omega T}}{a_0 + a_1 e^{-j\omega T} + \cdots + a_N e^{-jN\omega T}}, \tag{2.44}$$

The frequency response is a complex function that can be expressed in polar form as

$$H(e^{j\omega T}) = M(\omega) e^{j\theta(\omega)} \tag{2.45}$$

$$\text{where} \quad M(\omega) = |H(e^{j\omega T})|, \quad \text{and} \quad \theta(\omega) = \arg H(e^{j\omega T}), \tag{2.46}$$

are called the amplitude and phase response respectively. Most filters are designed to have a single passband in the amplitude response with a given center frequency and bandwidth. Other factors that affect the quality of the filter are passband ripple, stopband attenuation, settling time, and phase distortion. See Antoniou [1] or Oppenheim and Schafer [93] for more details on designing and evaluating digital filters.

Constant-Q filter banks are most commonly used for acoustic pattern recognition and analysis [25] (also see chapter 6), which means that the bandwidth of the filter is proportional to the center frequency of the passband. This means that the filters at low frequencies have very narrow bandwidth (*i.e.*, high spectral resolution) and filters at high frequencies have large bandwidth (*i.e.*, low spectral resolution). The rationale for this is that the human ear has roughly constant-Q frequency resolution. In fact, the critical bands of the human ear are not constant-Q, but instead have roughly constant bandwidth up to $f \approx 1$ kHz and then show constant-Q behaviour above 1 kHz [41, 57]. For this reason, many authors who are concerned with matching the psychophysical characteristics of human hearing use this model to build 'cochlear' filter banks [39, 96, 97, 115].

Filter banks are not the most efficient way to produce a spectral estimate of a signal for pattern recognition purposes. This is because the output sampling rate of the filter bank is the same as the input sampling rate, which in general is much too high (and noisy) for the pattern recognition system to handle. For this reason, some form of smoothing is usually required after the filter bank, such as an RMS integrator [25], or a lowpass decimator [39]. Therefore, much of the computationally expensive time resolution that a filter bank offers is not used. For this reason, an FFT with window functions in the frequency domain to represent the passbands of the filters is usually a more efficient approach. Multirate filter banks [125, 126] provide another alternative since the output sampling rates can be significantly lower than the input sampling rate, which leads to much more efficient implementations. For example, Desai and Shazeer [33] use a wavelet packet filter bank (which is multirate) to decompose underwater acoustic transients efficiently.

2.4.2.3 ARMA Modelling

Auto-regressive Moving Average (ARMA) modelling is another popular way of obtaining spectral estimates of a time domain signal. ARMA models approximate the spectrum of a signal as the amplitude response of a digital filter (see equations (2.44) and (2.46)). Although full ARMA modelling is employed in some situations [124], the vast majority of authors use linear predictive coding (LPC) to predict the coefficients of a pure AR model [102]. That is, LPC analysis gives the coefficients $a_0, a_1, \ldots, a_N$, from which an LPC spectrum can be produced using equations (2.44) and (2.46). This has been a popular method of spectral estimation in the speech recognition community since this smooth spectral estimate tends to emphasize the important features for phoneme classification [102] and speaker identification [35, 80, 132]. LPC spectra have also been used for the clustering of acoustic transients [99]. One of the disadvantages of this technique is that the order of the model N must be chosen *a priori*, which is problematic if different classes in the pattern recognition problem are best represented with different orders.

2.4.3 Frame Features

Features at the frame level can be divided into the following categories.

Spectral Coefficients - In some cases, the spectral coefficients are used directly as features. This is most common when the spectral estimate is produced by a filter bank with relatively few passbands [25] since FFT spectra often have a very large number of coefficients, which generally makes it difficult to train the classifier due to the curse of dimensionality (see section 2.2.5). However, if the proper measures are taken to ensure accurate training, then raw FFT spectral coefficients can be used for features as well [58].

Spectral Moments - One of the most common methods of reducing the information of the spectral estimate down to a few features is to compute moments [44]. For example, Pinkowski [99] compares FFT and LPC spectral moments as features for classifying bioacoustic signals. The company Musclefish[13] use the first and second moments of the FFT spectrum as features for their audio

[13]http://www.musclefish.com

database indexing products [133]. Scheirer and Slaney [111] also introduced a speech/music discrimination system for monitoring radio broadcasts which uses spectral moments for features.

Transformations - It is possible to change the spectral resolution of a spectrum by taking inner products of the spectrum with frequency domain window functions of varying centroid and bandwidth. In this way, an FFT spectrum can be converted into a constant-Q spectrum, or as is common for speech processing, a mel-frequency spectrum, which is constant-Q for higher frequency, but almost constant bandwidth for lower frequencies [134]. In some applications, such as voiced/unvoiced detection, the ratio of energy at high and low frequencies is an important discriminant feature which can also be easily implemented using frequency domain window functions and the FFT spectrum [20]. It is also popular to perform an FFT analysis on the log FFT spectrum $\log A(k\Delta\omega)$ or on the log mel-frequency spectrum, which produces cepstral coefficients and mel-frequency cepstral coefficients (MFCC) respectively [80, 102, 134]. The cepstrum of a signal decomposes the log spectrum into a sum of cosines. The first few cepstral coefficients describe the envelope of the log spectrum which are usually the important features for pattern recognition.

Pitch and Harmonicity - Most periodic sounds produced in nature are not perfect sinusoids so their spectra display a peak at the fundamental frequency f_0 and other peaks at the harmonics $f_n = (n+1)f_0$, with n a positive integer. The pitch of a sound is the perceptual quality associated with the fundamental frequency of the sound [59]. The harmonicity of a sound describes the distribution of energy in the harmonic peaks relative to the energy in the rest of the spectrum. For music recognition purposes pitch and harmonicity are extremely important perceptual qualities, which have been implemented in Musclefish's products [133, 134] and in Scheirer and Slaney's music/speech discriminator [111]. For voiced/unvoiced detection, the important feature for discrimination is whether or not the signal is pitched or not; the actual pitch of the signal is not important [20]. For these cases, the degree of pitch in the signal is often measured as the correlation of the signal at one pitch period lag, or more simply as the maximum value of the short time autocorrelation function of the signal.

2.4.4 Multi-frame Features

Features at the multi-frame level can be divided into several categories.

Feature Blocks - The frame level features can simply be grouped into blocks of length L' to give a $M \times L'$ feature block. In the case where the frame level features are spectral coefficients, the feature block corresponds to a time-frequency image of the signal. Feature blocks represent a large number of features, and due to the curse of dimensionality, they are rarely used in their raw form as features.

Frame Statistics and Descriptors - Statistics such as the mean, variance, autocorrelation and derivatives of the frame level features are often used as multi-frame features [111, 133, 134]. Moukas et al. [89] use peaks in the FFT spectrum of the frame energies to look for the impulsive sound of a helicopter rotor. Pinkowski [100] uses Fourier descriptors of the acoustic signal's time domain amplitude envelope to analyze and classify amplitude modulated bioacoustic waveforms. Statistics and descriptors of the frame level features provide an effective means of reducing the dimensionality of the feature set, while maintaining most of the discriminant information.

It should be emphasized that multi-frame features depend on having good features at the frame level.

2.4.5 HMM Features

Hidden Markov models require a sequence of features, so usually an $M \times L'$ feature block is passed to the HMM for classification. In this case the HMM is called a continuous HMM because the features that it is processing are continuous. However, in many cases, the multi-dimensional continuous features are converted into a finite set of discrete features through the use of a vector quantizer [51]. In this case, the HMM is called a discrete HMM. Like multi-frame features, HMM features depend on having good features at the frame level.

2.5 Summary

This chapter introduced the background material on acoustic pattern recognition and feature extraction which is required to understand the criteria that shaped the development of the discriminant dictionary projection pursuit feature extraction algorithm presented in chapter 4, and to understand the methodology used in chapter 5 and 6 to evaluate the various feature extraction algorithms. The following chapter takes a close look at the mathematics of wavelet packet signal processing which is at the core of the discriminant dictionary projection pursuit algorithm.

Chapter 3

Wavelet Packet Essentials

3.1 Introduction

This chapter introduces the mathematical background behind wavelet packets and their transforms. It is not intended to be a rigorous treatment of the material, but rather a concise synopsis that introduces the required notation and formalism that is required in later chapters. References to more thorough descriptions are given where appropriate. Good introductory sources on wavelets and wavelet packets can be found in Strang [119], Strang and Nguyen [121], Cohen [19], Graps [55], and in Vetterli and Kovačević [128]. Wavelet and wavelet packet theory are strongly tied to linear algebra concepts. Good introductory material for this subject area is given by Strang [118] and Strang and Borre [120], while more advanced material is given by Watkins [129] and Golub and Van Loan [53].

The descriptions that follow assume that a time domain signal is being analyzed and that frequency is expressed in samples per second (*i.e.*, Hz). This is done simply for familiarity reasons. The theory does not require this of course, and in fact the signals being analyzed in later chapters of this thesis are frequency domain signals. Also, while wavelets and wavelet packets are generally considered to be continuous time signals, a large class of them exist only in the asymptotic limit of a sequence of discrete time signals. In this thesis, the wavelet packet basis functions are defined to be the discrete time signals that are formed as part of the asymptotic sequence.

3.2 Notation and Terminology

3.2.1 Signal Space

The signals in this work are finite dimensional real-valued discrete signals $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^M$, where M is assumed to have a length that is a dyadic power given by 2^{m0}. Different vectors are indexed as $\boldsymbol{x}_i$, and the individual elements of the vectors are indicated using square brackets such as $\boldsymbol{x}_i[m]$. Unless otherwise stated, $\boldsymbol{x}$ should be thought of as a column vector. The dimensionality of a vector is referred to as $\dim\{\boldsymbol{x}\} = M$ where $\boldsymbol{x} \in \mathbf{R}^M$. The set of all $\boldsymbol{x} \in \mathcal{X}$ represents a vector space (see Strang [118] for a definition) with an inner product defined by

$$\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle = \boldsymbol{x}_i^T \boldsymbol{x}_j, \tag{3.1}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are both in $\mathcal{X}$, and a norm defined by

$$\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^T \boldsymbol{x}}, \tag{3.2}$$

which is the classical 2-norm.

The theory behind wavelets and wavelet packets was developed with respect to infinite dimensional Hilbert spaces such as $\boldsymbol{\ell}_2(\mathbf{Z})$ (*i.e.*, the space of complex valued functions defined on the integer number line that are square summable, that is, $\sum_{n\in\mathbf{Z}}|x[n]|^2 < \infty$) and $\boldsymbol{L}_2(\mathbf{R})$ (*i.e.*, the space of complex valued functions defined on the real number line that are square integrable, that is, $\int_{-\infty}^{\infty}|x(t)|^2 < \infty$). In practice however, signals that are manipulated by computers are finite dimensional discrete sequences, so the application of wavelet packet theory is usually applied to finite dimensional Hilbert spaces which can simply be thought of as vector spaces with the added constraint that the signals must have finite energy (always true for real-world signals).

The connection between finite dimensional discrete wavelet theory and infinite dimensional continuous wavelet theory is similar to the connection between discrete Fourier theory and continuous Fourier theory. In both cases, the inner product operator in the continuous space is replaced by a circular inner product operator in the finite discrete space which imposes a periodicity on the signals being analyzed. Also, the requirement of having dyadic power signal lengths (*i.e.*, signal lengths must be a

power of 2) in the finite dimensional discrete versions of wavelet and Fourier theory is only necessary for the implementation of fast transforms.

3.2.2 Basis Functions φ

A basis function φ is a real-valued[1] discrete sequence defined on $\mathbb{R}^M$, which is restricted to have unit norm $\|\varphi\| = 1$. In other words, φ defines a direction in M dimensional space. For example, in 3-dimensional space, φ defines a vector from the origin to a point on the unit sphere.

Different basis functions are indexed as φ_i, and the individual elements of the basis function are indicated using square brackets such as $\varphi_i[m]$. Unless otherwise stated, φ should be thought of as a column vector. The dimensionality of a basis function is referred to as $\dim\{\varphi\} = M$ where $\varphi \in \mathbb{R}^M$.

3.2.3 Subspaces Ω and Bases Φ

A subspace Ω of $\mathbb{R}^M$ is any subset of $\mathbb{R}^M$ that is closed under addition and scalar multiplication. That is, Ω is a subspace of $\mathbb{R}^M$ if and only if whenever $\varphi_i, \varphi_j \in \Omega$, and $\alpha \in \mathbb{R}$, then $\varphi_i + \varphi_j \in \Omega$ and $\alpha\varphi_i \in \Omega$.

Given linearly independent basis functions $\varphi_1, \varphi_2, \ldots, \varphi_r \in \mathbb{R}^M$, a linear combination of $\varphi_1, \varphi_2, \ldots, \varphi_r$ is a vector of the form $\alpha_1\varphi_1 + \alpha_2\varphi_2 + \cdots + \alpha_r\varphi_r$, where $\alpha_1, \alpha_2, \ldots, \alpha_r \in \mathbb{R}$. The numbers $\alpha_1, \alpha_2, \ldots, \alpha_r$ are called the coefficients of the linear combination. In matrix form, with the basis functions φ_i as columns of a bases matrix

$$\Phi = \begin{bmatrix} \varphi_1 & \varphi_2 & \cdots & \varphi_r \end{bmatrix} \tag{3.3}$$

and the coefficients α_i as elements of the column vector

$$\alpha = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_r \end{bmatrix}^T, \tag{3.4}$$

then the linear combination looks like $\Phi\alpha$.

The following properties can be defined for a matrix of basis functions,

[1] A more general definition of a basis function would include complex-valued sequences, such as the Fourier basis functions, but since all the basis functions in this work are real-valued, complex functions are excluded from our definition for simplicity's sake.

span$\{\Phi\}$ - is the set of all linear combinations of the columns of Φ, which can also be denoted as span$\{\varphi_1, \varphi_2, \ldots, \varphi_n\}$.

card$\{\Phi\}$ - is the total number of basis functions in the matrix which is referred to as the cardinality of the matrix and is equivalent to the number of columns in the matrix Φ.

dim$\{\Phi\}$ - is the dimensionality of the basis functions in the matrix which is equivalent to the number of rows in the matrix Φ.

rank$\{\Phi\}$ - is the number of linearly independent basis functions (or columns) in the matrix which is referred to as the rank of the matrix.

Since the basis functions in Φ are restricted to be linearly independent, card$\{\Phi\}$ = rank$\{\Phi\}$.

It is clear that span$\{\Phi\}$ is closed under scalar multiplication and addition, so Ω = span$\{\Phi\}$ is a subspace of $\mathbf{R}^M$. The basis functions in Φ are said to form a bases for the subspace Ω. Note that the actual basis functions used to define the subspace are not unique. Any set of rank$\{\Phi\}$ linearly independent vectors in the subspace could be used to define the subspace. The properties of the subspace that are independent of the basis functions used to describe it are

dim$\{\Omega\}$ - is the dimensionality of the subspace which is equal to the dimensionality of the basis functions used to define the subspace.

rank$\{\Omega\}$ - is the number of linearly independent basis functions required to describe the subspace, which is referred to as the rank of the subspace.

It should be clear that if Ω = span$\{\Phi\}$, then dim$\{\Omega\}$ = dim$\{\Phi\}$, and rank$\{\Omega\}$ = rank$\{\Phi\}$.

3.2.4 Orthogonal Bases

Two vectors, and in particular, two basis functions are said to be orthogonal if

$$\langle \varphi_i, \varphi_j \rangle = 0. \tag{3.5}$$

A set of basis functions $\{\varphi_1, \varphi_2, \ldots, \varphi_r\}$ is said to be orthogonal if

$$\langle \varphi_i, \varphi_j \rangle = 0 \quad \forall \quad i \neq j, \quad i, j \in \{1, 2, \ldots, r\}, \tag{3.6}$$

which can be succinctly expressed in matrix notation as $\boldsymbol{\Phi}^T\boldsymbol{\Phi} = \boldsymbol{I}$, where $\boldsymbol{\varphi}_1$, $\boldsymbol{\varphi}_2$, $\ldots$, $\boldsymbol{\varphi}_r$ form the columns of the bases matrix $\boldsymbol{\Phi}$ and $\boldsymbol{I}$ is the identity matrix. In this case, the bases matrix $\boldsymbol{\Phi}$ is said to be orthogonal, and thus defines an orthogonal bases for the subspace $\boldsymbol{\Omega} = \text{span}\{\boldsymbol{\Phi}\}$. Since the basis functions have unit norm, $\boldsymbol{\Phi}$ defines an orthonormal bases, but the term orthogonal is used throughout this thesis with normality assumed.

3.2.5 Orthogonal Complement $\boldsymbol{\Omega}^\perp$ and Direct Sum $\oplus$

The orthogonal complement of a subspace $\boldsymbol{\Omega} = \text{span}\{\boldsymbol{\Phi}\} \subset \mathbb{R}^M$ with $\text{rank}\{\boldsymbol{\Omega}\} = r$, is given by $\boldsymbol{\Omega}^\perp = \text{span}\{\boldsymbol{\Phi}^\perp\} \subset \mathbb{R}^M$ with $\text{rank}\{\boldsymbol{\Omega}^\perp\} = M - r$, where the matrix $\boldsymbol{\Phi}^\perp$ contains $M - r$ basis functions that are all orthogonal to the basis functions in $\boldsymbol{\Phi}$. The vector space $\mathbb{R}^M$ can then be expressed as the direct sum of $\boldsymbol{\Omega}$, and its orthogonal complement

$$\mathbb{R}^M = \boldsymbol{\Omega} \oplus \boldsymbol{\Omega}^\perp. \tag{3.7}$$

More generally, if $\hat{\boldsymbol{\Omega}} = \text{span}\{\hat{\boldsymbol{\Phi}}\} \subset \boldsymbol{\Omega} \subset \mathbb{R}^M$, then the orthogonal complement of $\hat{\boldsymbol{\Omega}}$ is given by $\hat{\boldsymbol{\Omega}}^\perp = \check{\boldsymbol{\Omega}} = \text{span}\{\check{\boldsymbol{\Phi}}\} \subset \boldsymbol{\Omega} \subset \mathbb{R}^M$, where $\check{\boldsymbol{\Phi}}$ contains the basis functions in $\boldsymbol{\Omega}$ that are orthogonal to the basis functions in $\hat{\boldsymbol{\Phi}}$. The subspace $\boldsymbol{\Omega}$ can then be expressed as the direct sum

$$\boldsymbol{\Omega} = \hat{\boldsymbol{\Omega}} \oplus \check{\boldsymbol{\Omega}}. \tag{3.8}$$

The direct sum implies that if $\hat{\boldsymbol{x}} \in \hat{\boldsymbol{\Omega}}$ and $\check{\boldsymbol{x}} \in \check{\boldsymbol{\Omega}}$, then $\boldsymbol{x} = \hat{\boldsymbol{x}} + \check{\boldsymbol{x}}$ is uniquely defined in $\boldsymbol{\Omega}$. Two subspaces can be expressed as a direct sum if and only if the intersection between the subspaces is empty [129].

3.2.6 Projection and Coefficient Operators $\boldsymbol{P}$ and $\boldsymbol{H}$

The orthogonal projection of a vector $\boldsymbol{x} \in \boldsymbol{\Omega} \subset \mathbb{R}^M$ onto a subspace $\hat{\boldsymbol{\Omega}} \subset \boldsymbol{\Omega}$ with $\dim\{\boldsymbol{\Omega}\} = M$ is given by

$$\begin{aligned} \hat{\boldsymbol{x}} &= \boldsymbol{P}\boldsymbol{x} &&= \hat{\boldsymbol{\Phi}}(\hat{\boldsymbol{\Phi}}^T\hat{\boldsymbol{\Phi}})^{-1}\hat{\boldsymbol{\Phi}}^T\boldsymbol{x} \\ &= \hat{\boldsymbol{\Phi}}\boldsymbol{H}\boldsymbol{x} &&= \hat{\boldsymbol{\Phi}}\hat{\boldsymbol{\alpha}}, \end{aligned} \tag{3.9}$$

where

- $\hat{\boldsymbol{\Phi}} \in \mathbb{R}^{M \times \hat{r}}$ is a bases matrix for the subspace $\hat{\boldsymbol{\Omega}} = \text{span}\{\hat{\boldsymbol{\Phi}}\}$ with $\text{rank}\{\hat{\boldsymbol{\Omega}}\} = \hat{r}$,
- $\hat{\boldsymbol{x}} \in \hat{\boldsymbol{\Omega}}$ is the orthogonal projection (often shortened to just projection) of $\boldsymbol{x}$ on $\hat{\boldsymbol{\Omega}}$,
- $\boldsymbol{P} = \hat{\boldsymbol{\Phi}}(\hat{\boldsymbol{\Phi}}^T\hat{\boldsymbol{\Phi}})^{-1}\hat{\boldsymbol{\Phi}}^T \in \mathbb{R}^{M \times M}$ is called the projection matrix or projection operator for the subspace $\hat{\boldsymbol{\Omega}}$ with respect to the bases $\hat{\boldsymbol{\Phi}}$,
- $\boldsymbol{H} = (\hat{\boldsymbol{\Phi}}^T\hat{\boldsymbol{\Phi}})^{-1}\hat{\boldsymbol{\Phi}}^T \in \mathbb{R}^{\hat{r} \times M}$ is called the coefficient matrix or coefficient operator for the subspace $\hat{\boldsymbol{\Omega}}$ with respect to the bases $\hat{\boldsymbol{\Phi}}$, and
- $\hat{\boldsymbol{\alpha}} = (\hat{\boldsymbol{\Phi}}^T\hat{\boldsymbol{\Phi}})^{-1}\hat{\boldsymbol{\Phi}}^T\boldsymbol{x} \in \mathbb{R}^{\hat{r}}$ is called the coefficient vector of $\hat{\boldsymbol{x}}$.

The component of $\boldsymbol{x}$ that is not projected onto $\hat{\boldsymbol{\Omega}}$ is given by

$$\check{\boldsymbol{x}} = \boldsymbol{x} - \hat{\boldsymbol{x}}, \tag{3.10}$$

where $\check{\boldsymbol{x}}$ is called the residual of $\hat{\boldsymbol{x}} = \boldsymbol{P}\boldsymbol{x}$. It follows that, as shown in figure 3.1, any vector $\boldsymbol{x}$ can be orthogonally decomposed into $\boldsymbol{x} = \hat{\boldsymbol{x}} + \check{\boldsymbol{x}}$, which is a central concept in wavelet and wavelet packet theory. Although figure 3.1 shows one and two rank subspaces, the same conceptual idea holds for any rank subspaces, so it is a good picture to keep in mind.

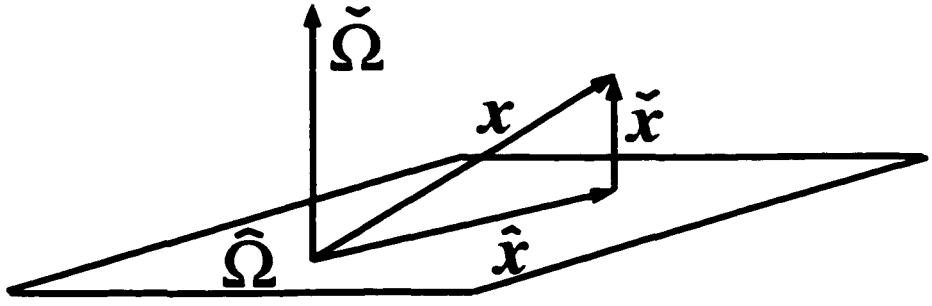

Figure 3.1. *The orthogonal decomposition of* $\boldsymbol{x} \in \boldsymbol{\Omega}$ *with* $\text{rank}\{\boldsymbol{\Omega}\} = 3$ *into* $\hat{\boldsymbol{x}} \in \hat{\boldsymbol{\Omega}}$ *with* $\text{rank}\{\hat{\boldsymbol{\Omega}}\} = 2$ *and* $\check{\boldsymbol{x}} \in \check{\boldsymbol{\Omega}}$ *with* $\text{rank}\{\check{\boldsymbol{\Omega}}\} = 1$.

The projection $\hat{\boldsymbol{x}}$ and the projection matrix $\boldsymbol{P}$ are mostly of theoretical value and are rarely computed since the coefficient vector $\hat{\boldsymbol{\alpha}}$ and coefficient matrix $\boldsymbol{H}$ contain all the information of the projection in a condensed form (*i.e.*, $\hat{r}$ elements rather than M for the vector, and $\hat{r} \times M$ elements rather than $M \times M$ elements for the matrices).

When $\hat{\boldsymbol{\Phi}}$ is an orthogonal bases for $\hat{\boldsymbol{\Omega}}$, $\hat{\boldsymbol{\Phi}}^T\hat{\boldsymbol{\Phi}} = \boldsymbol{I}$ (section 3.2.4), the projection matrix simplifies to $\boldsymbol{P} = \hat{\boldsymbol{\Phi}}\hat{\boldsymbol{\Phi}}^T$, and the coefficient matrix simplifies to $\boldsymbol{H} = \hat{\boldsymbol{\Phi}}^T$.

This is the fundamental reason for choosing orthogonal bases. For the case where $\text{rank}\{\hat{\boldsymbol{\Omega}}\} = 1$, the projection is onto a line defined by a single basis function, so $\hat{\boldsymbol{\varphi}}^T\hat{\boldsymbol{\varphi}} = 1$ (by definition in section 3.2.2), the projection matrix becomes $\boldsymbol{P} = \hat{\boldsymbol{\varphi}}\hat{\boldsymbol{\varphi}}^T$, and the coefficient matrix becomes $\boldsymbol{H} = \hat{\boldsymbol{\varphi}}^T$.

3.2.7 Dictionary $\mathcal{D}$

A dictionary of basis functions $\boldsymbol{\mathcal{D}} = \{\boldsymbol{\varphi}_\gamma | \gamma \in \Lambda\}$ can be formed where the parameter γ can index

1. time, in which case the dictionary is a time dictionary (*e.g.* the standard dictionary of discrete time impulse functions),
2. frequency, in which case the dictionary is a frequency dictionary (*e.g.* the Fourier dictionary of complex exponentials),
3. time and scale jointly, in which case the dictionary is a time-scale dictionary (*e.g.* wavelet dictionaries),
4. time, scale and frequency jointly, in which case the dictionary is a time-scale-frequency dictionary (*e.g.* wavelet packet and cosine packet dictionaries [130]),
5. polynomial order, in which case the dictionary is a polynomial dictionary (*e.g.* Legendre polynomials, Hermite polynomials etc. [2]),

or some other property of the basis functions in the dictionary. Since a dictionary is just a set of basis functions, the terminology $\text{span}\{\boldsymbol{\mathcal{D}}\}$, $\text{card}\{\boldsymbol{\mathcal{D}}\}$, $\dim\{\boldsymbol{\mathcal{D}}\}$, $\text{rank}\{\boldsymbol{\mathcal{D}}\}$ will be used to refer to the properties defined in section 3.2.3 with respect to the basis functions in the dictionary.

A dictionary is said to be

under-complete - if $\text{rank}\{\boldsymbol{\mathcal{D}}\} < \dim\{\boldsymbol{\mathcal{D}}\}$,

complete - if $\text{card}\{\boldsymbol{\mathcal{D}}\} = \text{rank}\{\boldsymbol{\mathcal{D}}\} = \dim\{\boldsymbol{\mathcal{D}}\}$,

over-complete - if $\text{card}\{\boldsymbol{\mathcal{D}}\} > \dim\{\boldsymbol{\mathcal{D}}\}$.

Under-complete and complete dictionaries are said to be orthogonal if the basis functions of the dictionary form an orthogonal set (section 3.2.4). Technically, an over-complete dictionary cannot be orthogonal since the cardinality is higher than the

dimensionality, but Wickerhauser [130] defines the over-complete wavelet packet dictionary to be orthogonal if it is created using orthogonal wavelets (see section 3.3 for more details), so this convention will be maintained here.

3.3 Wavelet Packets

Wavelet packets are basis functions that come in dictionaries, as described in section 3.3.1. The dictionary is defined in terms of a recursive subspace decomposition as described in section 3.3.2. Wavelet packets have several desirable properties that make them ideal for analyzing signals, including

time-frequency localization - Time localization refers to the interval in time (*i.e.*, the duration) over which the basis function is supported. If the function is identically zero outside a given interval, it is said to have compact support. Frequency localization refers the interval in frequency, given by the Fourier transform of the basis function (*i.e.*, the bandwidth), over which the basis function is supported. Whereas the standard bases of delta functions are perfectly localized in time but have no localization in frequency, and the Fourier basis functions are perfectly localized in frequency but have no localization in time, wavelet packets provide a compromise and are localized in both time and frequency. The localization in the two domains is however constrained by the Heisenberg uncertainty principle which states that $\Delta f \Delta t \geq 1/4\pi$. This means that as the wavelet packet becomes more localized in one domain, it must become less localized in the other.

fast transform - Since wavelet packets are defined through a multirate binary tree algorithm as described in section 3.3.2, the coefficient matrix $\boldsymbol{H}$ (defined in section 3.2.6) can be factored into several sub-matrices operating at progressively lower sampling rates. This allows the $M \log_2 M$ coefficients of a wavelet packet tree to be computed in $O(M \log M)$ operations [130]. The number of computations is on the same order as that required for the FFT operation, except that the FFT only produces $M/2$ distinct spectral coefficients for real signals.

3.3.1 Wavelet Packet Dictionary

A wavelet packet dictionary for a signal of length $M = 2^{m0}$ contains $M * (m0 + 1)$ basis functions $\varphi_{s,f,p}$ which are indexed by three integer parameters,

scale index $s = \{0, 1, \ldots, m0\}$ - The scale or extent of the basis function in the time domain is proportional to 2^s. Therefore, the effective time support of the basis functions double for every integer increase in s. Due to the similarity theorem in Fourier theory [7], if the scale doubles in the time domain, then the scale must half in the frequency domain. So the effective frequency support, or bandwidth of the basis functions halves for every integer increase in s.

frequency index $f = \{0, 1, \ldots, 2^s - 1\}$ - The Fourier transform of a basis function shows localized support in the frequency domain; the position of this support is proportional to $2^{-s}f$.

position index $p = \{0, 1, \ldots, 2^{m0-s} - 1\}$ - The position of the basis function in the time domain is proportional to $2^s p$.

The limits for each of these parameters may seem complicated and hard to remember, but they become very logical when you look at them in a map as shown in figure 3.2 for a small example with $m0 = 3$. On the top level ($s = 0$), there is only one frequency bin with M basis functions having no frequency localization (*i.e.*, delta functions), so the position index ranges from 0 to $2^{m0} - 1$. For every integer increase in s, the number of frequency bins double, which linearly segment the frequency axis from zero to the Nyquist frequency. On each level, the total number of basis functions must remain the same (*i.e.*, equal to 2^{m0}), so the number of position indexes in each frequency bin must half. When $s = m0$, the frequency axis is split into M bins (similar to a Fourier decomposition), so the basis functions on this level give the maximum frequency localization but have lost all of their time localization since there is only one position index in each frequency bin. This map is a convenient way to remember the limits and organization of the wavelet packet basis functions. Also, when analyzing a signal, it is used to display the coefficients $\alpha_{s,f,p}$ of the basis functions.

s = 0	0,0,0	0,0,1	0,0,2	0,0,3	0,0,4	0,0,5	0,0,6	0,0,7
s = 1	1,0,0	1,0,1	1,0,2	1,0,3	1,1,0	1,1,1	1,1,2	1,1,3
s = 2	2,0,0	2,0,1	2,1,0	2,1,1	2,2,0	2,2,1	2,3,0	2,3,1
s = 3	3,0,0	3,1,0	3,2,0	3,3,0	3,4,0	3,5,0	3,6,0	3,7,0

Figure 3.2. *Organization of the wavelet packet coefficients for* $M = 2^{m0} = 2^3$. *The same pattern persists for larger values of* $m0$. *The bold solid lines represent frequency bins* f *for each level* s, *while the dotted lines show the partition of each frequency bin into positions* p.

The orthogonal wavelet packet basis functions[2] are defined in terms of a recursive supspace decomposition [130] of $\mathbf{R}^M$ (where $M = 2^{m0}$) as shown in figure 3.3. Each subspace at scale s and frequency f is spanned by 2^{m0-s} identical waveforms $\{\varphi_{s,f,p} | p \in \{0, 1, \ldots, 2^{m0-s} - 1\}\}$ (*e.g.* see figure 3.4) which are circularly shifted relative to one another by $2^s p$. All of the basis functions at a given scale s are orthogonal to one another as given by

$$\langle \varphi_{s,f_i,p_j}, \varphi_{s,f_m,p_k} \rangle = \delta[f_i - f_j]\delta[p_m - p_k], \tag{3.11}$$

where $\delta[i - j]$ is the Kronecker delta function.

To help visualize the change in scale and frequency of the wavelet packet basis functions, several examples of the Coiflet wavelet packets[3], as defined by Daubechies [30], are plotted in figure 3.4 for the time domain and in figure 3.5 for the frequency domain. In both cases, the signal length was set to $M = 2^{m0} = 2^5$, but wavelet packets are only displayed down to $s = 3$.

[2]Biorthogonal wavelet packets are also possible as discussed in Strang and Nguyen [121], but only the orthogonal case is considered here.

[3]The order was set to 2, which gives a filter length of 12.

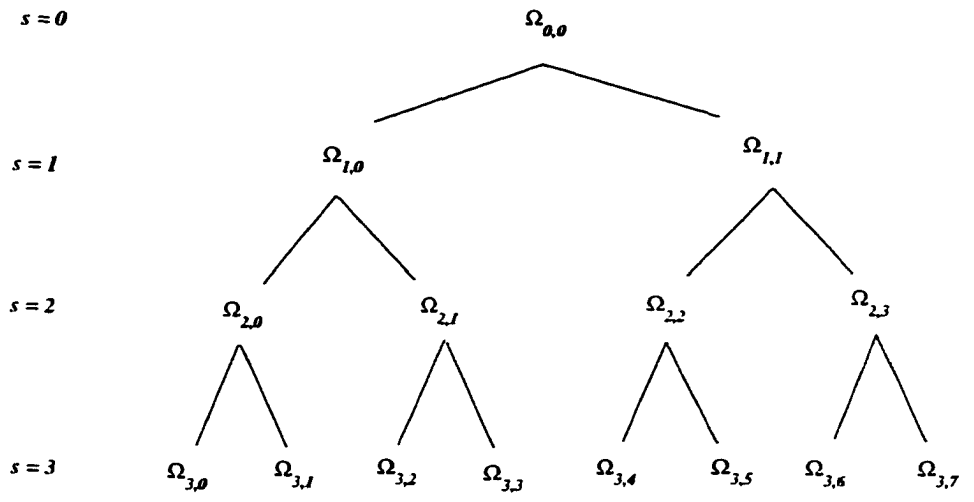

Figure 3.3. *The recursive binary tree subspace partition of* $\Omega_{0,0} = R^M$ *by the wavelet packet basis functions.*

3.3.2 Wavelet Packet Transform

Let the subspace decomposition shown in figure 3.6 represent any of the subspace splits shown in figure 3.3. The individual basis functions in a wavelet packet dictionary are defined according to the following equations[4]

$$\varphi_{s+1,2f,p} = \sum_{j=0}^{L-1} f_0[j]\varphi_{s,f,2p+j} \tag{3.12}$$

$$\varphi_{s+1,2f+1,p} = \sum_{j=0}^{L-1} f_1[j]\varphi_{s,f,2p+j} \tag{3.13}$$

where f_0 is a lowpass FIR (finite impulse response) filter, and f_1 represents a highpass FIR filter. Since the basis functions are defined to be delta functions for the top level $\Omega_{0,0}$, the filters f_0 and f_1 together with these formulae define the entire wavelet packet dictionary. Since the basis functions in $\Omega_{s+1,2f}$ and $\Omega_{s+1,2f+1}$ are linear combinations of basis functions in $\Omega_{s,f}$, this ensures that they are subspaces of $\Omega_{s,f}$. The issues involved in choosing "good" filters f_0 and f_1 are discussed in section 3.3.3. In this

[4]Actually, in some cases, the sequences f_0 and f_1 are switched to account for aliasing (see section 3.3.2.5).

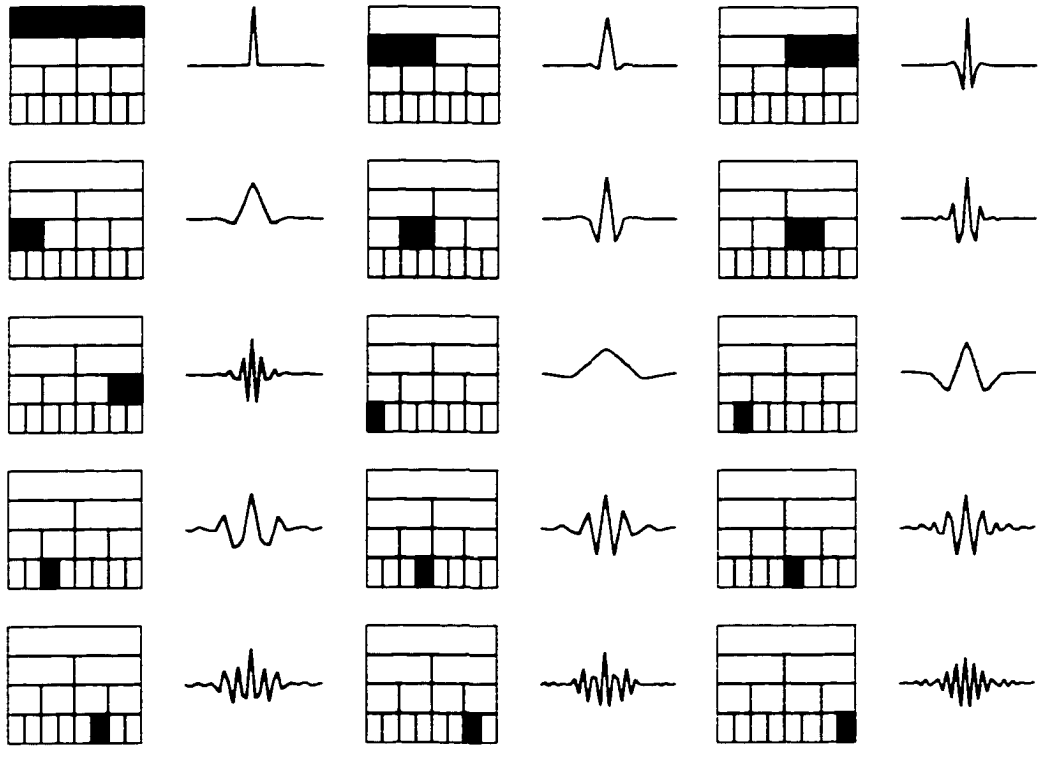

Figure 3.4. *The map to the left of each waveform indicates the scale s and frequency f of the wavelet packet plotted in the time domain. The position of the wavelet packet was shifted so that the max energy is roughly centered.*

section it is assumed that the sequences have been chosen to make each of the basis functions at a given level s orthogonal to one another.

The slow way of computing the wavelet packet transform of a signal $\boldsymbol{x}$ is to precompute each basis function using equations (3.12) and (3.13), and then compute the coefficient of the projection of $\boldsymbol{x}$ onto each basis function individually using

$$\alpha_{s,f,p} = \boldsymbol{\varphi}_{s,f,p}^T \boldsymbol{x}. \tag{3.14}$$

However, for a signal of length M, there are $M \log_2 M$ wavelet packet basis functions, and each coefficient requires M multiply-adds, so the complexity of computing the coefficients in this way is $O(M^2 \log M)$ which becomes formidable for large M. The following sections describe how to compute all of the wavelet packet coefficients with complexity $O(M \log M)$. The inverse wavelet packet transform has the same order of complexity. The fast algorithm described was first proposed by Mallat [77] for the

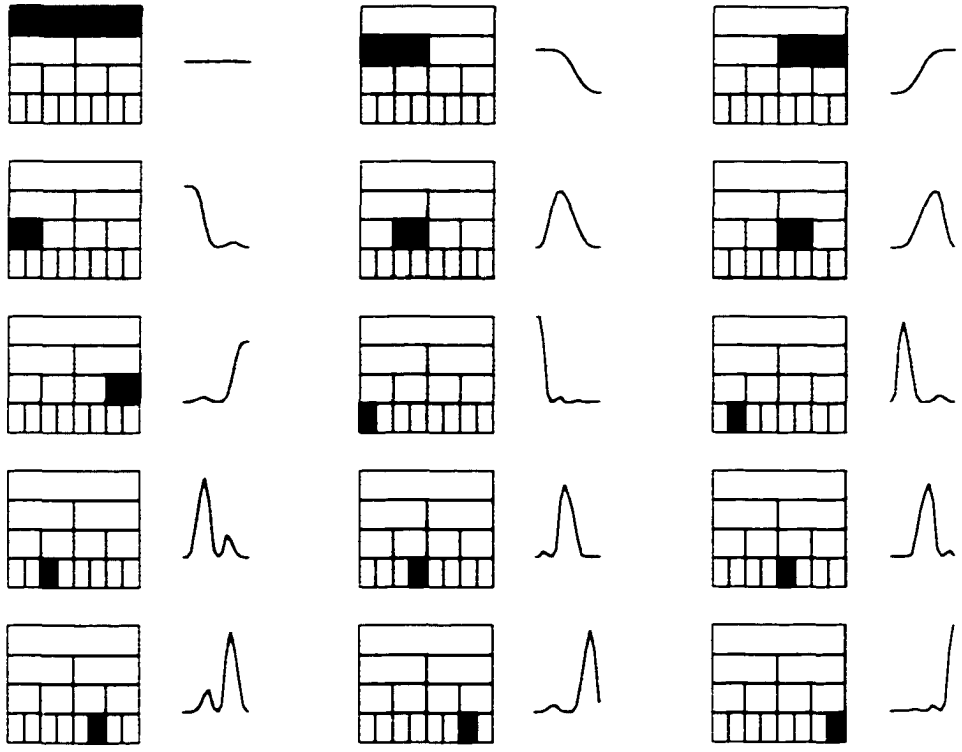

Figure 3.5. *The map to the left of each waveform indicates the scale s and frequency f of the wavelet packet plotted in the frequency domain. The waveforms plotted are the absolute value of the FFT of the time domain wavelet packets plotted from a frequency of 0 to the Nyquist frequency.*

wavelet transform, and was generalized to wavelet packets by Coifman and Wickerhauser [21, 130]. The following sections provide an illustrative description of the algorithm; for rigorous proofs, see Wickerhauser [130].

3.3.2.1 Single Level Forward Transform

The key to the single level forward transform is to relate the coefficients in $\Omega_{s+1,2f}$ and $\Omega_{s+1,2f+1}$ to the coefficients in $\Omega_{s,f}$. Taking the inner product of equations (3.12) and (3.13) with $\boldsymbol{x}$ and making the change of variables $k = 2p + j$ results directly in the equations

$$\alpha_{s+1,2f,p} = \sum_k f_0[k - 2p]\alpha_{s,f,k} \tag{3.15}$$

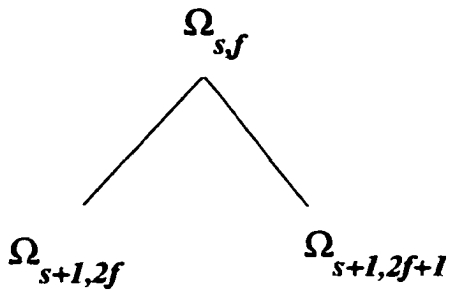

Figure 3.6. *A generic subspace split in the wavelet packet decomposition.*

$$\alpha_{s+1,2f+1,p} = \sum_k f_1[k-2p]\alpha_{s,f,k} \tag{3.16}$$

If the sequences f_0 and f_1 are replaced by $h_0[j] = f_0[-j]$ and $h_1[j] = f_1[-j]$, then the two equations can be interpreted as a convolution followed by downsampling[5] (*i.e.*, throwing away every other sample), as shown in figure 3.7, where H_0 and H_1 are linear FIR filters with impulse responses given by h_0 and h_1 respectively.

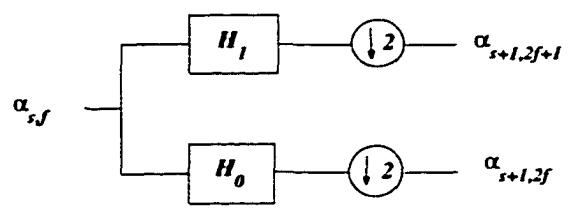

Figure 3.7. *Single level forward transform.*

3.3.2.2 Single Level Inverse Transform

The key to the single level inverse transform is to relate the coefficients in $\Omega_{s,f}$ to the coefficients in $\Omega_{s+1,2f}$ and $\Omega_{s+1,2f+1}$. Given a signal $\boldsymbol{x}$ of length $M = 2^{m0}$, the

[5]This operation makes h_0 and h_1 non-causal, but since the applications in this thesis do not involve real-time, this is irrelevant. For real time applications, a delay of $L-1$ must be incurred by defining $h_0[j] = f_0[L-1-j]$ and $h_1[j] = f_1[L-1-j]$.

projection in each subspace (as defined in section 3.2.6) is given by

$$\boldsymbol{x}_{s,f} = \sum_{p=0}^{2^{m0-s}-1} \alpha_{s,f,p}\boldsymbol{\varphi}_{s,f,p} \tag{3.17}$$

$$\boldsymbol{x}_{s+1,2f} = \sum_{p=0}^{2^{m0-s-1}-1} \alpha_{s+1,2f,p}\boldsymbol{\varphi}_{s+1,2f,p} \tag{3.18}$$

$$\boldsymbol{x}_{s+1,2f+1} = \sum_{p=0}^{2^{m0-s-1}-1} \alpha_{s+1,2f+1,p}\boldsymbol{\varphi}_{s,f,p} \tag{3.19}$$

where

$$\alpha_{i,j,k} = \langle \boldsymbol{\varphi}_{i,j,k}, \boldsymbol{x} \rangle = \boldsymbol{\varphi}_{i,j,k}^T \boldsymbol{x}. \tag{3.20}$$

Since the basis functions on any level s are chosen to be orthogonal, the expression

$$\boldsymbol{x}_{s,f} = \boldsymbol{x}_{s+1,2f} + \boldsymbol{x}_{s+1,2f+1} \tag{3.21}$$

is an orthogonal decomposition of $\boldsymbol{x}_{s,f}$, as described in section 3.2.6 and is uniquely defined. Substituting equation (3.12) in equation (3.18), equation (3.13) in equation (3.19), and equations (3.17), equation (3.18) and equation (3.19) in equation (3.21), and finally taking inner products on both sides with $\boldsymbol{\varphi}_{s,f,p}$ yields with some algebraic manipulation

$$\alpha_{s,f,p} = \sum_k f_0[p-2k]\alpha_{s+1,2f,k} + \sum_k f_1[p-2k]\alpha_{s+1,2f+1,k}. \tag{3.22}$$

This equation can be interpreted as upsampling of the two coefficient sequences (*i.e.*, inserting zeros between each of the samples) followed by convolution and summation, as shown in figure 3.8, where F_0 and F_1 are linear FIR filters with impulse responses given by f_0 and f_1 respectively.

3.3.2.3 Finite Length Signals

The forward transform shown in figure 3.7, and the inverse transform shown in figure 3.8 are valid for both infinite length signals and finite length signals. For finite length signals, the convolution operation is simply replaced with circular convolution, which imposes a periodicity on the signal. There are of course other ways to deal with finite length signals, such as

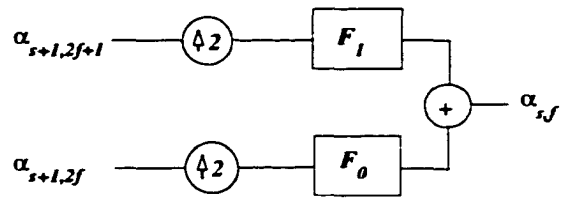

Figure 3.8. *Single level inverse transform.*

zero padding - the ends of the signal are padded with zeros, and the convolution operator extends past the edge of the signal. In this case, the number of coefficients at a given scale of the transform becomes larger as the scale increases. Wickerhauser calls this the aperiodic wavelet packet transform [130].

symmetric extension - the ends of the signal are extended symmetrically, so the transform can be computed simply by using data addressing rather than physically extending the signal. Strang describes this type of extension in detail [121].

filter modification - regular convolution filtering is replaced with special edge-filter matrices near the edges of the signal which define wavelets on the interval. This algorithm was developed by Cohen, Daubechies and Vial [18] for the wavelet transform, and could be extended to the wavelet packet transform.

Although there may be advantages to some of these other methods of dealing with finite length signals, the periodic extension is by far the most common technique used in the literature, and thus was adopted for the work in this thesis.

3.3.2.4 Multi-Level Wavelet Packet Transform

Armed with the knowledge of how to compute the coefficients at a larger scale given coefficients at a smaller scale (*i.e.*, the single level forward transform in section 3.3.2.1), and the fact that on the first level, the basis functions are delta functions (*i.e.*, the coefficients $\alpha_{0,0,p}$ are simply given by the signal values $\boldsymbol{x}$), the multi-level wavelet packet transform can be computed by iterating the single level transform in a binary tree, as shown in figure 3.9. The wavelet packet transform is very general,

as the wavelet transform and the cosine-like transform are special cases of it. The inverse multi-level wavelet packet transform is simply the mirror image of the forward transform (with the H filters replaced by F filters).

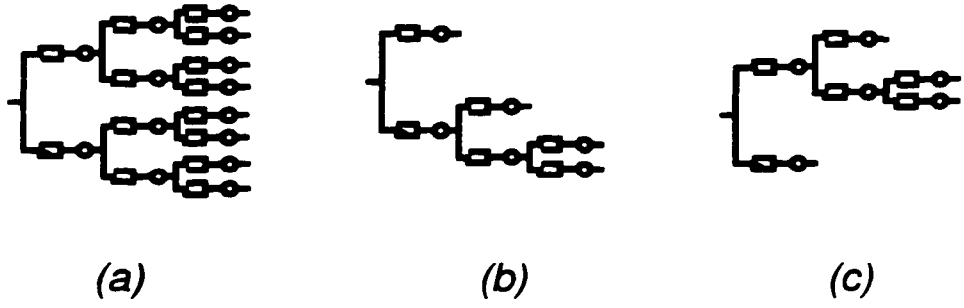

Figure 3.9. *Various three level wavelet packet transforms: (a) the full tree which is similar to a windowed cosine transform, (b) the wavelet transform, which iterates on the lowpass pass filter only, and (c) an arbitrary wavelet packet transform.*

3.3.2.5 Accounting for Aliasing

The wavelet transform iteratively splits the lowpass subspace in half and leaves the highpass subspace, which defines an octave resolution frequency decomposition. The wavelet packet transform iteratively splits the frequency domain of both the lowpass and highpass subspaces in half. On the first level of the decomposition, the signal $\boldsymbol{x}$ is split into a lowpass channel and highpass channel. They are both downsampled, which poses no problem for the lowpass channel, but causes the highpass channel to be aliased into the baseband. Therefore, the highest frequency components in the signal appear as the lowest frequency components of the downsampled highpass channel. When the highpass channel is split again at the second level of the decomposition, the frequency components in the lowpass channel actually have higher frequency components of $\boldsymbol{x}$ than the highpass channel!

If the single level forward transform shown in figure 3.7 is cascaded into the multi-level wavelet packet transform with H_0 always on the bottom, then this is called the 'natural' order. Let the bin number indicate the position in the tree at a given level, which increases from bottom to top; so the bin that has iteratively been operated on by H_0 is bin 0. The center frequency of the basis functions corresponding to

each of the bins does not increase monotonically with bin number. Wickerhauser [130] proved that the permutations of the bins that gives a monotonic increase of the center frequency of the basis functions is given by the inverse gray code. Therefore, the multi-level wavelet packet transform consists of first filtering using a binary tree with identical single level forward transforms, as shown in figure 3.7, followed by an inverse gray code permutation applied to each level of the binary tree. The inverse multi-level wavelet packet transform consists of first applying a gray code permutation to each level of the binary tree, followed by inverse filtering in a binary tree with identical single level inverse transforms as shown in figure 3.8.

3.3.3 Choosing Filter Coefficients

The filters and the filter structure that are used to define wavelets and wavelet packets have actually been around for more than two decades as they were first used by the sub-band coding community to define perfect reconstruction (PR) filterbanks [40, 116, 117]. A lot of excitement about wavelets in this past decade was due to the connection that was discovered by Daubechies [30] between PR filter banks and the continuous time basis functions that are a result of iterative applications of the filters [30] (also see Daubechies [31] for an interesting perspective on the origins of wavelets). This fueled an incredible amount of research which extended the concepts of wavelets in many directions, but the fundamental research is always focussed on the definition of good filters.

The filters contain all the information about the wavelet and wavelet packet basis functions they produce. For example, if the filters h_0 and h_1 are orthogonal to one another, then this automatically implies that the subspaces on any level of the wavelet packet decomposition are orthogonal to one another (see Wickerhauser [130] for a proof). Also, the more zeros that the frequency response of the lowpass filter has at $\omega = \pi$, the smoother the resulting wavelet packets will be. The multiplicity of the zeros at π is referred to as the number of vanishing moments of the wavelet.

PR filters were originally termed quadrature mirror filters (QMFs) since the frequency response of the lowpass filter h_0 is the mirror image about $\pi/2$ of the frequency response of the highpass filter h_1. Today, PR filters are defined according to many criteria such as orthogonal, biorthogonal, FIR, IIR, the number of vanishing moments

etc. Rather than go into the design criteria for PR filters (which are already well described by Strang and Nguyen [121]) or try to justify one set of PR filters over another, the Coiflet filters were adopted for the applications in this thesis since they have been used by other authors for wavelet packet feature extraction [108].

The Coiflet filters and the corresponding wavelet and scaling function are defined by Daubechies [30]. They are designed to have the maximum number of vanishing moments for a given support width in both the lowpass and highpass filter. The Coiflet filters are defined for several different orders N, for which the number of vanishing moments is given by $2N$ and the filter length is given by $6N$. The order used for the applications in this thesis was $N = 2$. As seen in figure 3.4, the Coiflet wavelet packets show approximate symmetry. Perfect symmetry is not possible, since Daubechies [30] proved that wavelets cannot be both perfectly symmetrical and orthogonal (with the exception of the Haar wavelet which also goes by the name Daubechies 2).

3.3.4 Optimization of Over-Complete Dictionaries

Signal compression has been one of the most common applications of wavelets and wavelet packets. The algorithms presented in this section represent dictionary optimization techniques that have been used for many signal compression problems. The transformation of these algorithms to a form that is suitable for discriminant feature extraction is presented in chapter 4.

In general, the goal of signal compression is to represent a signal $\boldsymbol{x}$ of length M with an approximation $\hat{\boldsymbol{x}}$ which is coded with as few bits as possible, while maintaining the fidelity of the signal. The fidelity of the approximation is usually measured with respect to the 2-norm $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|$, where the fidelity of $\hat{\boldsymbol{x}}$ is considered higher for lower values of $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|$. Other fidelity measures are also possible such as the 1-norm [14], or measures based on perceptual quality [128].

The transform coding approach to this problem is to express the signal as a linear combination of basis functions

$$\boldsymbol{x} = \boldsymbol{\Phi}\boldsymbol{\alpha}, \tag{3.23}$$

where $\boldsymbol{\Phi}$ contains the basis functions as columns, and $\boldsymbol{\alpha}$ is the coefficient vector of the expansion. The basis functions should be chosen so that most of the energy of the

signal $\boldsymbol{x}$ is contained in relatively few coefficients. In this way, all coefficients below a given threshold can be discarded and the signal representation is compressed. Given an over-complete dictionary $\boldsymbol{\mathcal{D}}$ of basis functions, the solution to equation (3.23) is not unique, so some method of selecting the "best" basis functions from the dictionary must be developed. Two such methods are described here[6], both of which have been generalized to solve the feature extraction problem for classification purposes, as described in chapter 4.

3.3.4.1 Best Basis Algorithm

This algorithm was first presented by Coifman and Wickerhauser [21], but a more detailed description is given by Wickerhauser [130]. Given a *single* signal of length $M = 2^{m0}$, the best basis algorithm selects a complete orthogonal bases from an over-complete wavelet packet dictionary that optimizes a criterion J. For signal compression, the criterion J should measure concentration of energy in the coefficients of expansion given in equation (3.23). An example is given by the Shannon entropy criterion[7]

$$J_S(\boldsymbol{\alpha}) \stackrel{\text{def}}{=} \sum_{i=1}^{M} |\alpha_i|^2 \log_2 \frac{1}{|\alpha_i|^2}. \tag{3.24}$$

This criterion is minimized in order to maximize the concentration of energy in the coefficients $\boldsymbol{\alpha}$. Other criteria are also possible [130], but are not discussed here.

Let $B_{s,f}$ denote a matrix of orthogonal basis functions (arranged as columns) belonging to the subspace $\boldsymbol{\Omega}_{s,f}$, such that the orthogonal projection of a signal $\boldsymbol{x}$ onto these basis functions is given by $\boldsymbol{\alpha}_{s,f} = B_{s,f}^T \boldsymbol{x}$. Let $A_{s,f}$ represent the best basis for the signal $\boldsymbol{x}$ restricted to the span of $B_{s,f}$. The best basis algorithm prunes the full wavelet packet tree to select the best basis by comparing the criterion function of each parent node to its two children as shown in algorithm 1.

---

[6] A third method called Basis Pursuit [14, 15] has also been developed, but is not discussed here since this method has not yet been generalized from the signal compression problem to the discrimination problem.

[7] This criterion is not identical to Shannon's entropy since the elements $|\alpha_i|^2$ do not form a valid *pdf*. However, normalizing each element by $\|\boldsymbol{\alpha}\|^2$ causes the criterion to be non-additive, which destroys the efficiency of the algorithm. Therefore, J_S is adopted as an approximation of Shannon's entropy [130].

Algorithm 1 Best Basis [Coifman and Wickerhauser]

Given: a signal $\boldsymbol{x}$, a dictionary $\mathcal{D}$ of wavelet packet basis functions down to scale S, and a criterion function J,

step 1: Begin at scale S and set $A_{S,f} = B_{S,f}$ for $f = 0, 1, \ldots, 2^S - 1$.

step 2: Determine the best subspace $A_{s,f}$ iteratively for $s = S-1, S-2, \ldots, 0$ and $f = 0, 1, \ldots, 2^s - 1$ using

$$A_{s,f} = \begin{cases} B_{s,f} & \text{if } J(B_{s,f}^T \boldsymbol{x}) \leq J(A_{s+1,2f}^T \boldsymbol{x} \cup A_{s+1,2f+1}^T \boldsymbol{x}), \\ A_{s+1,2f} \oplus A_{s+1,2f+1} & \text{otherwise,} \end{cases} \tag{3.25}$$

to minimize the criterion J, and

$$A_{s,f} = \begin{cases} B_{s,f} & \text{if } J(B_{s,f}^T \boldsymbol{x}) \geq J(A_{s+1,2f}^T \boldsymbol{x} \cup A_{s+1,2f+1}^T \boldsymbol{x}), \\ A_{s+1,2f} \oplus A_{s+1,2f+1} & \text{otherwise,} \end{cases} \tag{3.26}$$

to maximize the criterion J.

Assuming that the full wavelet packet tree of coefficients $\alpha_{s,f,p}$ have been precomputed using the fast transform described in section 3.3.2, the criterion J has been computed for each subspace $\Omega_{s,f}$ (this is usually called the statistic tree), and the criterion is additive (*i.e.*, $J(\alpha_1 + \alpha_2) = J(\alpha_1) + J(\alpha_2)$) then the complexity of this search algorithm is very low (*i.e.*, $O(M)$). The computational advantage of using an additive criterion is evident since

$$J(A^T_{s+1,2f}\boldsymbol{x} \cup A^T_{s+1,2f+1}\boldsymbol{x}) = J(A^T_{s+1,2f}\boldsymbol{x}) + J(A^T_{s+1,2f+1}\boldsymbol{x}), \tag{3.27}$$

so the computation of the criterion of the union of the coefficients from two disjoint subspaces can be computed by simple addition of the criterion of the coefficients from the individual subspaces.

When the algorithm is complete, $A_{0,0}$ contains the best basis of the signal $\boldsymbol{x}$ restricted to the span of $B_{0,0} = \mathbf{R}^M$. The algorithm always chooses a complete orthogonal set of M basis functions defined by a set of disjoint subspaces $\Omega_{s,f}$ spanned by 2^{m0-s} basis functions each. The criterion J is globally optimized over the subspaces of the wavelet packet decomposition. This algorithm can also be applied to cosine packet decompositions [130].

3.3.4.2 Matching Pursuits

The matching pursuits algorithm was first proposed by Mallat and Zhang [79]. It is a greedy algorithm that chooses at each iteration a waveform from a dictionary $\mathcal{D}$ that is best adapted to approximate part of the signal, as shown in algorithm 2.

The matching pursuits algorithm returns a structure book $\{\alpha_j, \gamma_j\}_{j=1}^k$ which contains the coefficient of projection and the index of the basis function for each stage of the algorithm. An approximation of the signal can be obtained through the linear expansion $\hat{\boldsymbol{x}} = \boldsymbol{\Phi}\boldsymbol{\alpha}$, where $\boldsymbol{\Phi} = [\boldsymbol{\varphi}_{\gamma_1}\boldsymbol{\varphi}_{\gamma_2}\ldots\boldsymbol{\varphi}_{\gamma_k}]$, and $\boldsymbol{\alpha} = [\alpha_1\alpha_2\ldots\alpha_k]^T$. The matching pursuits algorithm described in algorithm 2 does not return the optimal coefficient vector $\boldsymbol{\alpha}$ in the least squares sense (*i.e.*, $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|$ is not minimized). The optimal coefficient vector is given by $\boldsymbol{\alpha}_{opt} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\boldsymbol{x}$, which is only equal to $\boldsymbol{\alpha}$ if the the basis functions in $\boldsymbol{\Phi}$ are orthogonal, which is rarely the case. The algorithm that minimizes $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|$ is called matching pursuits with backfitting [79], since the coefficients are recomputed at each stage of the algorithm (*i.e.*, the signal is backfitted to the collection of basis functions). This algorithm is described in algorithm 3.

Algorithm 2 Matching Pursuits [Mallat and Zhang]

Given: the following quantities

- a *single* signal $\boldsymbol{x}$.
- a dictionary $\mathcal{D} = \{\boldsymbol{\varphi}_\gamma | \gamma \in \Lambda\}$.

step 1: initialize the following variables

- set the signal approximation to zero $\boldsymbol{s}_0 = 0$.
- set the residual to be equal to the signal $\boldsymbol{r}_0 = \boldsymbol{x}$.
- set the counter $k = 1$.

step 2: select an index $\gamma_k = \underset{\gamma_k \in \Lambda}{\operatorname{argmax}} \langle \boldsymbol{r}_{k-1}, \boldsymbol{\varphi}_{\gamma_k} \rangle$

step 3: perform the following operations

- set the projection coefficient $\alpha_k = \langle \boldsymbol{r}_{k-1}, \boldsymbol{\varphi}_{\gamma_k} \rangle$.
- set the signal approximation $\boldsymbol{s}_k = \boldsymbol{s}_{k-1} + \alpha_k \boldsymbol{\varphi}_{\gamma_k}$.
- set the residual $\boldsymbol{r}_k = \boldsymbol{x} - \boldsymbol{s}_k$.

step 4: If the stop condition is reached then quit and return the structure book $\{\alpha_j, \gamma_j\}_{j=1}^{k}$, else set $k = k + 1$ and goto step 2.

Algorithm 3 Matching Pursuits with Backfitting [Mallat and Zhang]

Given: the following quantities

- a *single* signal $\boldsymbol{x}$.
- a dictionary $\mathcal{D} = \{\boldsymbol{\varphi}_\gamma | \gamma \in \Lambda\}$.

step 1: initialize the following variables

- set the signal approximation to zero $\boldsymbol{s}_0 = 0$.
- set the residual to be equal to the signal $\boldsymbol{r}_0 = \boldsymbol{x}$.
- set the bases matrix to be empty $\boldsymbol{\Phi}_0 = [\emptyset]$.
- set the counter $k = 1$.

step 2: select an index $\gamma_k = \underset{\gamma_k \in \Lambda}{\operatorname{argmax}} \langle \boldsymbol{r}_{k-1}, \boldsymbol{\varphi}_{\gamma_k} \rangle$

step 3: perform the following operations

- update the bases matrix $\boldsymbol{\Phi}_k = [\boldsymbol{\Phi}_{k-1} \boldsymbol{\varphi}_k]$.
- update the coefficient matrix

$$\boldsymbol{H}_k = (\boldsymbol{\Phi}_k^T \boldsymbol{\Phi}_k)^{-1} \boldsymbol{\Phi}_k^T. \tag{3.28}$$

- compute all the projection coefficients $\boldsymbol{\alpha} = \boldsymbol{H}_k \boldsymbol{x}$.
- set the signal approximation $\boldsymbol{s}_k = \boldsymbol{\Phi}_k \boldsymbol{\alpha}$.
- set the residual $\boldsymbol{r}_k = \boldsymbol{x} - \boldsymbol{s}_k$.

step 4: If the stop condition is reached then quit and return the structure book $\{\alpha_j, \gamma_j\}_{j=1}^k$, else set $k = k + 1$ and goto step 2.

The algorithm is very general in that $\mathcal{D}$ can contain any kind of basis functions, and the algorithm stop condition can be set so that the algorithm terminates when a fixed number of basis functions are chosen, a fixed percentage of the total energy is contained in the signal approximation, or some other criterion. Mallat and Zhang focus on basis functions that have time-frequency localization; they term these basis functions time-frequency atoms. For a wavelet packet dictionary, the algorithm is very efficient since the correlation of the residual with each of the basis functions in the dictionary can be computed using the fast transform described in section 3.3.2. Mallat and Zhang compare this algorithm to the best basis algorithm of Coifman and Wickerhauser (see section 3.3.4.1); they find that matching pursuits can represent the significant parts of a signal with fewer basis functions when the signal is highly non-stationary. This is to be expected since the best basis algorithm does a global optimization over the whole signal, and is thus best suited for stationary signals.

3.4 Summary

This chapter presented the required background material on wavelet packet signal processing which is used in the discriminant dictionary projection algorithm developed in the next chapter. Although the details of the wavelet packet transform where presented in this chapter, it is not necessary to understand all these details to understand the discriminant dictionary projection pursuit algorithm. The important concept to take away is that the wavelet packet basis functions are localized in time and frequency, and the projections on the whole dictionary can be computed with complexity $O(M \log M)$ where M is the dimensionality of the signal. In a similar manner, it is not necessary to understand the details of the FFT algorithm which computes the Fourier coefficients with complexity $O(M \log M)$ (actually few people do) in order to take advantage of the FFT.

The optimization algorithms presented in the last section are the core algorithms that are used in many signal compression applications of wavelet packets. The transformation of these algorithms into a form that is suitable for the discriminant feature exctraction problem is discussed in the next chapter. With this background information in hand, it is now time to get to the core contribtuion of this thesis - the

dictionary projection pursuit algorithm.

Chapter 4

Adapted Wavelet Packet Feature Extraction

4.1 Introduction

Most applications of wavelets and wavelet packets such as signal compression [114], denoising [36], and singularity detection [78] are designed to work with a *single* signal $\boldsymbol{x}$. This differs significantly from the wavelet packet algorithms discussed in this chapter, which deal with an *ensemble* of signals $\mathcal{X}^{(k)} = \{\boldsymbol{x}_i^{(k)}\}_{i=1}^{N^{(k)}}$ from a multitude of classes $\{\omega^{(k)}\}_{k=1}^{K}$, where the goal is to find the features in the signals that best discriminate between the classes.

The main emphasis of this chapter is to develop the dictionary projection pursuit algorithm which is a powerful and efficient technique for solving multivariate estimation problems. This algorithm can be thought of as a fast approximate version of the projection pursuit algorithm [61] which is applicable when the vectors are samples from an underlying continuous waveform. The discriminant form of dictionary projection pursuit can be used to extract features for pattern recognition purposes.

This chapter starts with a review of notation and terminology (section 4.2), and then gives a review of waveform feature extraction methods (section 4.3). The next section introduces the dictionary projection pursuit algorithm (section 4.4), which is then applied to the problem of finding the maximum directions of variance in a dataset (*i.e.*, approximation of the Karhunen-Loève Transform) in section 4.5, and as a feature extractor for the discrimination problem in section 4.6. This chapter concludes with a description of the feature extraction algorithms that are used in the classification experiments of chapters 5 and 6 (section 4.7).

4.2 Notation and Terminology

4.2.1 Class Notation $\omega^{(k)}$

Given a set of labelled data $\mathcal{L} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ having K classes $y_i \in \{\omega^{(k)}\}_{k=1}^{K}$, the patterns $\boldsymbol{x}_i \in \mathbb{R}^M$ belonging to class $y_i = \omega^{(k)}$ will be denoted as $\mathcal{X}^{(k)} = \{\boldsymbol{x}_i^{(k)}\}_{i=1}^{N^{(k)}}$. When any of the quantities defined below apply to the vectors $\boldsymbol{x}^{(k)}$ from class $\omega^{(k)}$, then the corresponding symbol is superscripted with (k).

4.2.2 Statistics

Let an ensemble of signals $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^{N} \in \Omega \subset \mathbb{R}^M$ be represented in matrix form as

$$\boldsymbol{X} = [\boldsymbol{x}_1 \boldsymbol{x}_2 \ldots \boldsymbol{x}_N] \in \mathbb{R}^{M \times N}. \tag{4.1}$$

Let $\mathbf{1}_N$ be a column vector of length N, such that the mean of the ensemble is defined by

$$\boldsymbol{\mu} \stackrel{\text{def}}{=} N^{-1} \boldsymbol{X} \mathbf{1}_N = N^{-1} \sum_{i=1}^{N} x_i, \tag{4.2}$$

the correlation matrix of the ensemble is defined as[1]

$$\boldsymbol{Q} \stackrel{\text{def}}{=} N^{-1} \boldsymbol{X} \boldsymbol{X}^T = N^{-1} \sum_{i=1}^{N} x_i x_i^T. \tag{4.3}$$

and the covariance matrix of the ensemble is defined as[2]

$$\boldsymbol{\Sigma} \stackrel{\text{def}}{=} \boldsymbol{Q} - \boldsymbol{\mu}\boldsymbol{\mu}^T = N^{-1} \sum_{i=1}^{N} (x_i - \boldsymbol{\mu})(x_i - \boldsymbol{\mu})^T. \tag{4.4}$$

These all represent the plug-in estimates (which, assuming a Normal distribution, are maximum likelihood estimates) for the corresponding theoretical variables, $E\left\{\tilde{\boldsymbol{X}}\right\} \sim \boldsymbol{\mu}$, $E\left\{\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^T\right\} \sim \boldsymbol{Q}$, $E\left\{\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^T\right\} - E\left\{\tilde{\boldsymbol{X}}\right\} E\left\{\tilde{\boldsymbol{X}}\right\}^T \sim \boldsymbol{\Sigma}$, where $\tilde{\boldsymbol{X}}$ within the expectation is a random vector. The statistics corresponding to individual classes will be denoted as $\boldsymbol{\mu}^{(k)}$, $\boldsymbol{Q}^{(k)}$, and $\boldsymbol{\Sigma}^{(k)}$.

[1] This definition is a deviation from the multivariate statistics literature, where the correlation matrix usually means the matrix of correlation coefficients.

[2] Often the covariance estimate is scaled by a factor $N/(N-1)$ to remove bias.

4.2.3 Maps $\Gamma(\gamma)$

Given a dictionary $\mathcal{D} = \{\varphi_\gamma | \gamma \in \Lambda\}$, the coefficient of projection α_γ for each member of the ensemble $\mathcal{X}$ can be computed as $\varphi_\gamma^T \boldsymbol{x}_i$. Let a map element $\Gamma(\gamma)$ be defined to be a scalar function of the projection coefficients $\varphi_\gamma^T \boldsymbol{x}_i$ such that the map Γ represents the value of the scalar function for *all* basis functions in the dictionary. It is also possible to define a vector map over the basis functions which will be denoted as $\boldsymbol{\Gamma}(\gamma)$, which defines a vector of values associated with a given basis function φ_γ.

The coefficient map for a vector $\boldsymbol{x}_i$ is defined as

$$\Gamma_{\alpha_i}(\gamma) \stackrel{\text{def}}{=} \varphi_\gamma^T \boldsymbol{x}_i, \tag{4.5}$$

the mean map is defined as

$$\Gamma_{\boldsymbol{\mu}}(\gamma) \stackrel{\text{def}}{=} \varphi_\gamma^T \boldsymbol{\mu} = N^{-1} \sum_{i=1}^{N} \varphi_\gamma^T \boldsymbol{x}_i, \tag{4.6}$$

the energy map[3]

$$\Gamma_{\boldsymbol{Q}}(\gamma) \stackrel{\text{def}}{=} \varphi_\gamma^T \boldsymbol{Q} \varphi_\gamma = N^{-1} \sum_{i=1}^{N} (\varphi_\gamma^T \boldsymbol{x}_i)^2, \tag{4.7}$$

the variance map is defined as

$$\Gamma_{\boldsymbol{\Sigma}}(\gamma) \stackrel{\text{def}}{=} \varphi_\gamma^T \boldsymbol{\Sigma} \varphi_\gamma = \Gamma_{\boldsymbol{Q}}(\gamma) - \Gamma_{\boldsymbol{\mu}}^2(\gamma), \tag{4.8}$$

and the criterion map is defined as

$$\Gamma_J(\gamma) \stackrel{\text{def}}{=} J(\varphi_\gamma^T \boldsymbol{X}), \tag{4.9}$$

where $\boldsymbol{X}$ contains the vectors from an ensemble $\mathcal{X}$ as columns, and J is a criterion function (see section 4.6.1). Other maps such as the weight map Γ_w will be defined with appropriate subscripts later in this chapter. As usual, maps corresponding to a specific class are superscripted as $\Gamma^{(k)}$.

Maps are a convenient way to summarize operations that have been performed for each basis function. They can also be used to represent operations that are to

[3]It seems more reasonable to call this the correlation map, but since it was called the energy map by both Saito [108] and Learned and Willsky [71], this convention will be retained.

be performed identically for all basis functions in the dictionary $\mathcal{D}$. For example, the statement $\Gamma_{\mu}^{2}(\gamma)$ means that the mean value corresponding to each basis function should be squared, and the statement $\Gamma_{w}(\gamma)\Gamma_{J}(\gamma)$ indicates the element-wise multiplication of the weights for each basis function with the criterion function for each basis function.

Although, a map can be defined for any dictionary of basis functions, this thesis will only deal with wavelet packet dictionaries with the Coiflet order 2 filtes, which for a signal of length $M = 2^{m0}$ are indexed by

$$\mathcal{D} = \{\varphi_{\gamma} | \gamma \in \Lambda\}, \tag{4.10}$$

where $\Lambda = (s, f, p), s \in \{0, 1, \ldots, S\}, f \in \{0, 1, \ldots, 2^{s}-1\}$, and $p \in \{0, 1, \ldots, 2^{m0-s}-1\}$. The coefficient of projection α_{γ} for each member of the ensemble $\mathcal{X}$ can be computed for each basis in the dictionary using the fast transform described in section 3.3.2. However, the projection coefficients will still be written as $\varphi_{\gamma}^{T}\boldsymbol{x}_{i}$. It is useful, but not essential to organize the elements of a wavelet packet map according to the scheme defined in section 3.3.1, which allows each map to be displayed as an image map with familiar structure.

4.3 Waveform Feature Extraction

The advantages of reducing the dimensionality of the feature space for pattern recognition purposes was discussed in section 2.2.5. The general problem of choosing a subset of features from a larger set of possible features has been well documented by Devijver and Kittler [34]. They divide the problem into two groups which they term feature selection and feature extraction[4]. Feature selection is the process of choosing a subset of features from the non-transformed feature space. This is the appropriate technique to use when the cost of acquiring each measurement in the feature vector is high. Feature extraction is the process of choosing a subset of features from a transformed feature space. This is the appropriate technique to use if the information in the feature space is distributed among the features in a correlated manner and the cost of acquiring each feature is insignificant. The transformation is usually

[4]Feature extraction is also called feature reduction by some authors [123].

taken to be linear, but non-linear techniques have also been proposed such as the self-organizing maps [68] and neural networks [62, 63].

For sampled waveform datasets (*e.g.* acoustic spectra), linear feature extraction is an effective method of selecting features. It consists of finding basis functions Φ such that the coefficients of projection $\boldsymbol{\alpha} = \boldsymbol{\Phi}^T \boldsymbol{x}$ can be used as features for a pattern recognition system. Ideally, basis functions should be chosen to minimize the classification error of a pattern recognition system, but since this is a hard criterion to optimize due to its non-linearity, basis functions are generally chosen to optimize some discriminant criterion J (see section 4.6.1).

The following two sections describe methods for finding basis functions for waveform feature extraction by optimizing a criterion function. The first section focusses on continuous optimization techniques where the directions or basis functions of the projection can take on values from a continuum. The second section focusses on much more efficient dictionary optimization techniques which only use a finite number of directions or basis functions in the pursuit of good projections.

4.3.1 Continuous Optimization

One of the most popular methods of finding basis functions for linear feature extraction is to find the directions of maximum variance in the dataset, subject to orthogonality constraints. The criterion function to be maximized is then given as $J(\boldsymbol{\varphi}) = \boldsymbol{\varphi}^T \boldsymbol{\Sigma} \boldsymbol{\varphi}$, subject to the constraint that every basis function chosen must be orthogonal to the previously selected basis functions. The solution to this particular problem is given by the eigenvectors of the covariance matrix $\boldsymbol{\Sigma}$ corresponding to the largest eigenvalues, which is known as the Karhunen-Loève transform (KLT) in the pattern recognition community [34], and as principal component analysis (PCA) in the statistical community [43]. This method of selecting features for a dataset is probably the most commonly used method in pattern recognition [34, 123], which works very well if the variance in the dataset is due to between-class scatter rather than within-class scatter and the number of classes is small. See algorithm 5 for a more thorough description of the KLT algorithm.

Another popular criterion to maximize is Fisher's criterion [38]

$$J(\mathbf{\Phi}) = \frac{|\mathbf{\Phi}^T \mathbf{\Sigma}_B \mathbf{\Phi}|}{|\mathbf{\Phi}^T \mathbf{\Sigma}_W \mathbf{\Phi}|}, \tag{4.11}$$

where $\mathbf{\Sigma}_B$ and $\mathbf{\Sigma}_W$ are the between-class and within-class covariance of the data set respectively (see section 2.2.4.4 for more details), and $\mathbf{\Phi}$ contains the basis functions of the projection subspace as columns. The solution to this problem is given by the generalized eigenvectors of the ordered pair $(\mathbf{\Sigma}_B, \mathbf{\Sigma}_W)$ corresponding to the largest generalized eigenvalues. This method of selecting features for a dataset works well in general, but low variance noisy subspaces $\tilde{\Omega} = \text{span}\{\tilde{\mathbf{\Phi}}\}$ can sometimes foil this method when the sample size N is small since $|\tilde{\mathbf{\Phi}}^T \mathbf{\Sigma}_W \tilde{\mathbf{\Phi}}|$ can be smaller than $|\tilde{\mathbf{\Phi}}^T \mathbf{\Sigma}_B \tilde{\mathbf{\Phi}}|$ by random chance giving the noisy subspace a high criterion value.

Subspace methods of pattern recognition [91] offer another alternative for extracting waveform features. In this method, features are extracted for each class individually by maximizing the criterion function $J^{(k)} = \boldsymbol{\varphi}^T \mathbf{Q}^{(k)} \boldsymbol{\varphi}$ subject to the constraint that every basis function chosen must be orthogonal to the previously selected basis functions; this method finds the bases that contain the maximum energy for a given class. The solution to this problem is given by the eigenvectors of the class correlation matrix $\mathbf{Q}^{(k)}$ that correspond to the largest eigenvalues.

Projection pursuit (PP) is yet another technique for linear feature extraction. The concept was first proposed by Kruskal [69], but was named and elaborated on by Friedman and Tukey [48] as a method for exploratory data analysis. The concepts were extended to encompass regression problems [47] and eventually discrimination problems [61]. The idea behind the method is to find low dimensional projections of high dimensional multivariate data that optimize a projection criterion via numerical optimization. By varying the projection criterion, the same algorithm can be used for exploratory data analysis, regression, and feature extraction [61]. Although, the algorithm can be applied to data with dimension M, to find subspaces of arbitrary dimension $p < M$, the optimization process becomes prohibitively expensive when $p > 2$ and M is large. For this reason, a greedy approach is often used where good one dimensional projections are found in an iterative fashion. After a given projection is selected, then the structure found in that projection is removed from the dataset before the next projection is searched for.

The only draw back of this algorithm is its computational complexity, which becomes formidable for large M since numerical optimization must be used. All of the eigenanalysis optimization problems discussed above can be solved by PP, albeit with higher computational cost. The advantage of this algorithm is that almost any criterion function can be optimized whereas eigenanalysis is restricted to optimizing a small class of criterion functions. Therefore robust versions of finding the KL basis functions can be formulated [61], and criterion functions such as Fisher's criterion can be modified to overcome their shortcomings in the small sample size domain (see section 4.6.1.2). The dictionary projection pursuit algorithm developed in section 4.4 can be thought of as a fast approximate form of PP which is applicabale when the feature vector consists of samples of an underlying continuous waveform or image.

The first three criteria mentioned in this section all have closed form solutions based on eigenanalysis of a correlation matrix or covariance matrix. Other criterion functions such as those proposed by Kullback [70] and those compiled by Devijver and Kittler [34] must be optimized using a continuous optimization procedure such as the projection pursuit algorithm, which is very computationally expensive. In all cases, the basis functions are chosen continuously from all possible basis functions in the input space. An alternative to this continuous optimization is to use dictionary optimization which is computationally efficient and returns easily interpretable results.

4.3.2 Dictionary Optimization

Dictionary methods of selecting basis functions for waveform feature extraction dismiss the utopian goal of finding the best basis functions from the infinitely many possible basis functions, and instead try to find the best basis functions from a finite dictionary $\mathcal{D} = \{\boldsymbol{\varphi}_\gamma | \gamma \in \Lambda\}$. The dictionary is generally chosen to be over-complete with basis functions that are interpretable (*i.e.*, have specific well defined characteristics) and are well adapted to the problem. For waveform feature extraction and in particular for spectral feature extraction, time-frequency dictionaries are especially useful, such as the Gabor dictionary [79], wavelet packet dictionaries [21, 130], and cosine packet dictionaries [130].

Learned and Willsky [71] developed a methodology for classifying transient sonar

signals using a wavelet packet dictionary to select features. They use the energy map (equation (4.7)) of each class averaged over a bin (*i.e.*, each bin represents a subspace $\Omega_{s,f}$ that has fixed scale s and frequency f and a range of positions p) to look for the basis functions that focus the energy of the class the most. They use the singular value decomposition (SVD) of the bin energy map to show that each class is well represented by a single average bin energy map. This is done by showing that the ratio of the second largest and largest singular values is very small. They compensate for ambient ocean noise and select bin energies by eye that separate the classes. Using a nearest neighbour and neural network classifier, they obtain classification results with less than 5 % error rate. While this approach is interesting and demonstrates the power of a wavelet packet dictionary, it requires a lot of manual analysis of the data in order to choose the best features. Ultimately, a more automated approach is desired.

Saito and Coifman [108, 109, 110] have developed an algorithm for extracting waveform features which they call local discriminant bases (LDB). They use the power of the best basis algorithm (see section 3.3.4.1) to automatically select basis functions from a wavelet packet or cosine packet dictionary. They use the energy map (equation (4.7)) of each class $\Gamma_Q^{(k)}(\gamma)$ to compute a discriminant value for each basis function (see section 4.6.1). The sum of the discriminant value over a subspace $\Omega_{s,f}$ is used to produce a statistic tree, and the best basis algorithm is used to maximize the discriminant measure over all possible subspaces. This algorithm works very well in general, but since it uses subspace optimization, if a good feature exists in a parent subspace and a child subspace, then the algorithm must choose one subspace over the other even if the two features are orthogonal to one another. The best basis algorithm forms an orthogonal bases for the signal which is important for signal compression (which the algorithm was originally designed for) but has questionable significance when trying to choose only a few features for discrimination.

Buckheit and Donoho [13] have developed a dictionary method for extracting waveform features that they call discriminant pursuit (DP). Their algorithm generalizes the matching pursuits algorithm [79] and is specifically designed to be used with Fisher's LDA classifier (see section 2.2.4.4). They suggest that the main reason that Fisher's LDA fails in the neo-classical setting (*i.e.*, small sample size N and

large dimension M) is due to noise in the empirical mean differences. They apply the matching pursuits (MP) algorithm (see section 3.3.4.2) to the difference between class means (contrasts) in order to search for discriminant features.

However, they essentially ignore the estimate of the within-class covariance $\mathbf{\Sigma}_W$ in their analysis except in a weighted version of their algorithm that performs a rescaling on the contrasts by $\mathbf{\Sigma}_W^{-1}$ before applying the MP algorithm. This is a questionable operation since in the neo-classical setting, where N is often less than M, the estimated within-class covariance is guaranteed to be singular since the N samples can span at most a subspace of dimension $N-1$ [43]. The poor performance of the weighted version of their algorithm in the neo-classical setting is demonstrated with synthetic data in chapter 5 and with recorded data in chapter 6.

In summary, the best basis algorithm is able to optimize different criterion function over a wavelet packet or cosine packet dictionary, which allowed Saito and Coifman to extract features by optimizing discriminant criterion functions. The matching pursuits algorithm is focussed on optimizing correlations with a *single* signal, so Buckheit and Donoho used this algorithm to extract features by optimizing correlations with the class mean differences. Since the best basis algorithm is only subspace optimal (*i.e.*, it optimizes over subspaces rather than individual basis functions) and is limited to optimizing a restricted class of criterion functions (see section 4.6.1), it would be advantageous to develop a dictionary algorithm for optimizing a general criterion function over individual basis functions as described in the next section.

4.4 Dictionary Projection Pursuit

Dictionary projection pursuit is a fast approximate projection pursuit algorithm that iteratively finds basis functions from a dictionary $\mathcal{D}$ that optimizes a projection criterion J subject to approximate orthogonality conditions. The algorithm chooses bases from the dictionary, and then uses a Gram-Schmidt like orthogonalization process [91, 118] to ensure that the basis functions returned by the algorithm are orthogonal to one another. These basis functions, when aligned as columns of a matrix $\boldsymbol{A}$, form an orthogonal bases for a subspace $\mathbf{\Omega} = \text{span}\{\boldsymbol{A}\}$ which is said to be the criterion optimized subspace. For wavelet packet or cosine packet dictionaries, the algorithm

is fast with complexity $O(\hat{M}M \log M)$, where M is the dimensionality of the input space and $\hat{M}$ is the number of basis functions in the selected subspace $\boldsymbol{A}$. The algorithm can be applied to problems such as finding approximate Karhunen-Loève basis functions (section 4.5), finding a discriminant subspace for pattern recognition tasks (section 4.6), and for many other tasks simply by changing the projection criterion index.

4.4.1 Dictionary Projection Pursuit Algorithm

Given a dictionary $\mathcal{D} = \{\boldsymbol{\varphi}_\gamma | \gamma \in \Lambda\}$, and an ensemble of vectors $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ with $\boldsymbol{x}_i \in \mathbb{R}^M$, dictionary projection pursuit iteratively selects basis functions from $\mathcal{D}$ which optimize a projection criterion function $J(\boldsymbol{\varphi}_\gamma^T \boldsymbol{X})$ subject to approximate orthogonality constraints, where $\boldsymbol{X}$ contains the vectors in $\mathcal{X}$ as columns. The criterion function for a particular basis $\boldsymbol{\varphi}_\gamma$ is a scalar function of the projection coefficients of the vectors in $\mathcal{X}$ on that basis function. Evaluating the criterion function for each basis in the dictionary results in the criterion map $\Gamma_J(\gamma)$, as defined in equation (4.9) which remains unchanged throughout the algorithm.

Thinking of each basis function as a direction vector in $\mathbb{R}^M$, then it is assumed that the criterion function varies smoothly as a function of direction. That is, if a basis function is given a small perturbation, $\bar{\boldsymbol{\varphi}}_\gamma = \boldsymbol{\varphi}_\gamma + \delta\boldsymbol{\varphi}_\gamma$, then the criterion function should also experience a small perturbation $J(\bar{\boldsymbol{\varphi}}_\gamma^T \boldsymbol{X}) = J(\boldsymbol{\varphi}_\gamma^T \boldsymbol{X}) + \delta J(\boldsymbol{\varphi}_\gamma^T \boldsymbol{X})$. This is a valid assumption for almost all useful criterion functions which helps justify the weighting scheme that is used in this algorithm.

The dictionary projection pursuit algorithm is described formally in algorithm 4. It is an iterative greedy optimization algorithm such that at stage k, it chooses the optimal basis function index γ_k based on maximizing or minimizing the product of the criterion map $\Gamma_J(\gamma)$ and a weight map from the last iteration $\Gamma_w(\gamma)$ (step 2). The weight map is a measure of the linear independence of the dictionary basis function from the subspace of previously selected basis functions $\boldsymbol{\Omega} = \text{span}\{\boldsymbol{A}\}$. One of the factors that makes this algorithm fast is that the criterion map $\Gamma_J(\gamma)$ is not recomputed at each stage of the algorithm; it is simply modified by the weight map. The criterion map $\Gamma_J(\gamma)$ is assumed to be pre-computed and is only modified in step 1.1 and 1.2 where it is inverted for minimization and made strictly positive so that

the weight map properly scales the criterion map in step 2.

Let $\boldsymbol{P}$ and $\boldsymbol{P}^{\perp}$ be the projection matrices for $\boldsymbol{\Omega} = \text{span}\{\boldsymbol{A}\}$ and its residual orthogonal complement $\boldsymbol{\Omega}^{\perp}$ respectively. Then $\hat{\boldsymbol{\varphi}}_{\gamma_k} = \boldsymbol{P}\boldsymbol{\varphi}_{\gamma_k}$ is the projection of $\boldsymbol{\varphi}_{\gamma_k}$ (*i.e.*, the dictionary basis function selected at the k^{th} step) on $\boldsymbol{\Omega}$, which can be computed efficiently as $\hat{\boldsymbol{\varphi}}_{\gamma_k} = \boldsymbol{A}\boldsymbol{\Gamma}_{\boldsymbol{\alpha}}(\gamma_k)$ (step 3) since $\boldsymbol{\Gamma}_{\boldsymbol{\alpha}}(\gamma_k)$ stores as a column vector the coefficients of projection for $\boldsymbol{\varphi}_{\gamma_k}$ on the previously selected basis functions in $\boldsymbol{\Omega} = \text{span}\{\boldsymbol{A}\}$. The residual orthogonal projection $\check{\boldsymbol{\varphi}}_{\gamma_k} = \boldsymbol{P}^{\perp}\boldsymbol{\varphi}_{\gamma_k}$ is then easily computed as $\check{\boldsymbol{\varphi}}_{\gamma_k} = \boldsymbol{\varphi}_{\gamma_k} - \hat{\boldsymbol{\varphi}}_{\gamma_k}$ (step 3). The residual projection is normalized $\boldsymbol{a}_k = \check{\boldsymbol{\varphi}}_{\gamma_k}/\|\check{\boldsymbol{\varphi}}_{\gamma_k}\|$ and added to the set of previously selected basis functions $\boldsymbol{A} = [\boldsymbol{A}, \boldsymbol{a}_k]$ (step 4), which ensures that the bases describing the selected subspace are always orthogonal and normalized. This step is similar to applying Gram-Schmidt orthogonalization to a set of basis functions [91, 118].

Due to the orthogonality of the vectors in $\boldsymbol{A}$, the coefficients of projection of the dictionary basis functions on the subspace $\boldsymbol{\Omega} = \text{span}\{\boldsymbol{A}\}$ can be computed independently for each vector in $\boldsymbol{A}$. So the next element in $\boldsymbol{\Gamma}_{\boldsymbol{\alpha}}(\gamma)$ can be computed as $\boldsymbol{\varphi}_{\gamma}^{T}\boldsymbol{a}_k$ (step 6) for each basis function in the dictionary. This is the most computationally intensive step in the algorithm, but if the dictionary is a wavelet packet dictionary (section 3.3.2) or a cosine packet dictionary [130], then a fast algorithm can be employed that requires $O(M \log M)$ operations. Therefore, if $\hat{M}$ basis functions are selected, then the entire search algorithm has complexity of roughly $O(\hat{M}M \log M)$.

Each basis function in the dictionary can be orthogonally decomposed as $\boldsymbol{\varphi}_{\gamma} = \hat{\boldsymbol{\varphi}}_{\gamma} + \check{\boldsymbol{\varphi}}_{\gamma}$, which splits $\boldsymbol{\varphi}_{\gamma}$ into a piece $\hat{\boldsymbol{\varphi}}_{\gamma}$ which is in the subspace $\boldsymbol{\Omega} = \text{span}\{\boldsymbol{A}\}$ defined by the previously selected basis functions and $\check{\boldsymbol{\varphi}}_{\gamma}$ which is in the residual subspace $\boldsymbol{\Omega}^{\perp}$. The norm map $\Gamma_{\|\hat{\boldsymbol{\varphi}}\|^2}(\gamma)$ stores the squared norm of $\hat{\boldsymbol{\varphi}}_{\gamma}$ for each basis function in the dictionary. Since the vectors in $\boldsymbol{A}$ are orthonormal, the squared norm of $\hat{\boldsymbol{\varphi}}_{\gamma}$ is given by $\|\hat{\boldsymbol{\varphi}}_{\gamma}\|^2 = \|\sum_{i=1}^{k}(\boldsymbol{\varphi}_{\gamma}^{T}\boldsymbol{a}_i)\boldsymbol{a}_i\|^2 = \sum_{i=1}^{k}(\boldsymbol{\varphi}_{\gamma}^{T}\boldsymbol{a}_i)^2$. Therefore, the norm map can be updated as $\Gamma_{\|\hat{\boldsymbol{\varphi}}\|^2}(\gamma) = \Gamma_{\|\hat{\boldsymbol{\varphi}}\|^2}(\gamma) + (\boldsymbol{\varphi}_{\gamma}^{T}\boldsymbol{a}_k)^2$ (step 7), where $(\boldsymbol{\varphi}_{\gamma}^{T}a_k)^2$ was previously computed for the coefficient vector map. Due to Pythagoras' theorem, the norm in the selected and residual subspace satisfy $\|\boldsymbol{\varphi}_{\gamma}\|^2 = \|\hat{\boldsymbol{\varphi}}_{\gamma}\|^2 + \|\check{\boldsymbol{\varphi}}_{\gamma}\|^2 = 1$, so the norm map for the residual subspace can easily be computed as $\Gamma_{\|\check{\boldsymbol{\varphi}}\|^2}(\gamma) = 1 - \Gamma_{\|\hat{\boldsymbol{\varphi}}\|^2}(\gamma)$.

The weight for each basis φ_γ at stage k is computed as

$$\Gamma_w(\gamma) = f(\Gamma_{\|\tilde{\varphi}\|^2}(\gamma)) \tag{4.12}$$

$$= f\left(1 - \Gamma_{\|\tilde{\varphi}\|^2}(\gamma)\right). \tag{4.13}$$

That is, the weight for a basis φ_γ is a function of the norm of its projection in the residual subspace. For our purposes, the function was defined as $f(x) = x$, but using other functional forms such as $f(x) = x^2$ or $f(x) = x^{1/2}$ causes the algorithm to penalize basis functions more and less respectively for their lack of linear independence from $\mathbf{\Omega}$. Notice that if a basis function φ_γ is orthogonal to $\mathbf{\Omega}$, then $\|\tilde{\varphi}_\gamma\| = 1$, and the weight is given as one. When φ_γ is in the subspace $\mathbf{\Omega}$, then $\|\tilde{\varphi}_\gamma\| = 0$, and the weight is zero. For intermediate positions, the weight takes on values between zero and one. In this way, the next basis function is always chosen to be linearly independent of $\mathbf{\Omega}$. It should be clear that the weight function for a given basis function is a monotonically decreasing function of k.

All eigenanalysis continuous optimization problems have a criterion function that must be optimized subject to orthogonality constraints (see section 4.3.1). Dictionary projection pursuit offers an alternative dictionary optimization algorithm for these problems which is computationally more efficient. Since there are only a finite number of basis functions in the dictionary, strict orthogonality is not imposed at each stage; instead basis functions are penalized for their lack of independence from the subspace of previously selected basis functions. However, the basis functions returned by the algorithm $\boldsymbol{A}$ which define the selected subspace $\mathbf{\Omega}$ *are* orthogonal. In addition, this algorithm can be used to optimize criterion functions that previously could only be optimized by the very computationally intensive PP algorithm. However, in both cases, this algorithm is best suited to problems where the vectors are defined as samples of a continuous waveform or image. Examples of applying this algorithm to different problems are studied in the following sections.

Remark: It should be noted that there is a subtle but important difference between the types of criterion functions that the dictionary projection pursuit and best basis algorithm (section 3.3.4.1) can optimize. The best basis algorithm is a global optimization technique that seeks to extremize a criterion function over a complete bases. The type of criterion functions that are optimized are those that describe the distribution of criterion values (*i.e.*, the criterion function evaluated for

Algorithm 4 Dictionary Projection Pursuit

Given: the following quantities

- a dictionary $\mathcal{D} = \{\boldsymbol{\varphi}_\gamma | \gamma \in \Lambda\}$ with $\boldsymbol{\varphi}_\gamma \in \mathbb{R}^M$.
- a criterion map $\Gamma_J(\gamma) = J(\boldsymbol{\varphi}_\gamma^T \boldsymbol{X})$.
- a scalar function f for the weight map.

step 1: perform the following initialization steps

1. If 'maximize' set $\Gamma_J(\gamma) = \Gamma_J(\gamma)$
   If 'minimize' set $\Gamma_J(\gamma) = -\Gamma_J(\gamma)$
2. set $\Gamma_J(\gamma) = \Gamma_J(\gamma) - \min_\gamma \Gamma_J(\gamma)$.
3. initialize the subspace bases to be empty $\boldsymbol{A} = [\emptyset]$.
4. initialize the norm map to be zero $\Gamma_{\|\hat{\boldsymbol{\varphi}}\|^2}(\gamma) = 0$.
5. initialize the coefficient vector map to be empty $\boldsymbol{\Gamma}_{\boldsymbol{\alpha}}(\gamma) = [\emptyset]$.
6. initialize the weight map to be all ones $\Gamma_w(\gamma) = 1$.
7. initialize the counter $k = 1$.

step 2: select an index $\gamma_k = \underset{\gamma \in \Lambda}{\operatorname{argmax}} \Gamma_J(\gamma)\Gamma_w(\gamma)$.

step 3: Compute the projection of the selected basis function on the previously selected subspace $\boldsymbol{\Omega} = \operatorname{span}\{\boldsymbol{A}\}$ using $\hat{\boldsymbol{\varphi}}_{\gamma_k} = \boldsymbol{P}\boldsymbol{\varphi}_{\gamma_k} = \boldsymbol{A}\boldsymbol{\Gamma}_{\boldsymbol{\alpha}}(\gamma_k)$ and on the residual subspace $\boldsymbol{\Omega}^\perp$ using $\check{\boldsymbol{\varphi}}_{\gamma_k} = \boldsymbol{P}^\perp \boldsymbol{\varphi}_{\gamma_k} = \boldsymbol{\varphi}_{\gamma_k} - \hat{\boldsymbol{\varphi}}_{\gamma_k}$.

step 4: Compute the normalized residual $\boldsymbol{a}_k = \check{\boldsymbol{\varphi}}_{\gamma_k} / \|\check{\boldsymbol{\varphi}}_{\gamma_k}\|$, update the selected subspace bases $\boldsymbol{A} = [\boldsymbol{A}, \boldsymbol{a}_k]$ and store the structure book elements $\mathcal{B}_k = \{\gamma_k, \Gamma_J(\gamma_k), \Gamma_w(\gamma_k)\}$.

step 5: If the stop conditions are satisfied, then quit and return the selected subspace bases $\boldsymbol{A}$ and structure book $\mathcal{B}$.

step 6: Update the coefficient vector map $\boldsymbol{\Gamma}_{\boldsymbol{\alpha}}(\gamma) = [\boldsymbol{\Gamma}_{\boldsymbol{\alpha}}(\gamma), \boldsymbol{\varphi}_\gamma^T \boldsymbol{a}_k]$.

step 7: Update the norm map $\Gamma_{\|\hat{\boldsymbol{\varphi}}\|^2}(\gamma) = \Gamma_{\|\hat{\boldsymbol{\varphi}}\|^2}(\gamma) + (\boldsymbol{\varphi}_\gamma^T \boldsymbol{a}_k)^2$.

step 8: Update the weight map $\Gamma_w(\gamma) = f(1 - \Gamma_{\|\hat{\boldsymbol{\varphi}}\|^2}(\gamma))$.

step 9: Set $k = k + 1$ and goto step 2.

each basis function in the selected bases) such as entropy measures. In addition, to make the algorithm efficient, the criterion function must be additive. The dictionary projection pursuit algorithm on the other hand is a greedy optimization technique that does not consider how the criterion function is distributed over a complete bases. The projection criterion is a function of the projection coefficients of each vector in the dataset on a *single* basis. Therefore, the dictionary projection pursuit algorithm is more flexible than the best basis algorithm in terms of the types of criterion functions that can be optimized.

4.5 Approximate Karhunen-Loève Transform

4.5.1 Karhunen-Loève Transform

The Karhunen-Loève (KL) transform is probably one of the best known and extensively used statistical orthogonal transforms that enjoys many optimality properties as described by Oja [91], Devijver and Kittler [34], and Wickerhauser [130]. The transform was originally conceived in terms of continuous second-order random processes independently by Karhunen and Loève (see Oja [91] for details). However, when presented in terms of a discrete time process, the mathematics is identical to principal component analysis (PCA), which was introduced to the statistics community by Hotelling much earlier [60], and is closely related to the well known mathematical technique of singular value decomposition (SVD) [129, 130].

Given an ensemble of vectors $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ with $\boldsymbol{x}_i \in \Omega \subset \mathbb{R}^M$, the KL transform is given by

$$\hat{\alpha} = \Phi^T \boldsymbol{x}, \tag{4.14}$$

where $\Phi \in \mathbb{R}^{M \times \hat{M}}$ contains the orthogonal basis functions φ_i of the KL transform as columns. The number of basis functions $\hat{M} \leq M$ is usually taken to be less than M, in which case the KL transform represents a compression of data. The signal can be reconstructed using the KL linear expansion

$$\hat{\boldsymbol{x}} = \Phi\hat{\alpha} = \sum_{j=1}^{\hat{M}} (\varphi_j^T \boldsymbol{x})\varphi_j. \tag{4.15}$$

The basis functions φ_j are selected to maximize the criterion function $J(\varphi) = \varphi^T \Sigma \varphi$, subject to the orthogonality constraint $\varphi_j^T \varphi_k = \delta[j-k] \ \forall \ j,k \in \{1,2,\ldots,\hat{M}\}$, where Σ is the theoretical covariance matrix of the ensemble $\mathcal{X}$ which is usually approximated using equation (4.4)[5]. Since $\varphi_j^T \Sigma \varphi_j$ gives the variance of the projection coefficients on the basis function φ_j, the KL basis functions find the directions of maximum variance in the dataset. The solution is given by the eigenvectors of Σ corresponding to the largest eigenvalues[6], which also ensures that the covariance matrix of the projection coefficients is diagonal (*i.e.*, the KL coefficients are mutually uncorrelated). The algorithm for computing the basis functions is described more formally in algorithm 5 and a diagram showing the KL basis functions for an example with $M = 2$ is given in figure 4.1.

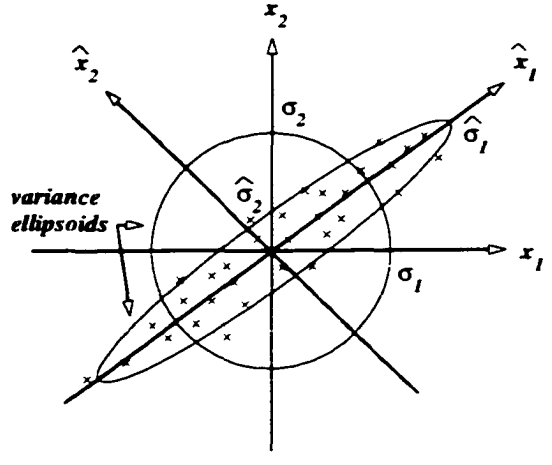

Figure 4.1. *The KL Basis functions φ_1 and φ_2 for this dataset are aligned with $\hat{x}_1$ and $\hat{x}_2$ respectively. If Σ represents the covariance matrix for the dataset, then $\sigma_1^2 = [1\ 0]\Sigma[1\ 0]^T = \Sigma_{11}$, $\sigma_2^2 = [0\ 1]\Sigma[0\ 1]^T = \Sigma_{22}$, $\hat{\sigma}_1^2 = \varphi_1^T \Sigma \varphi_1$, and $\hat{\sigma}_2^2 = \varphi_2^T \Sigma \varphi_2$. The KL transform maximizes the projected variance of the dataset along the KL basis functions. It also minimizes the volume of the variance ellipsoid.*

The idea behind using the KL transform for feature extraction in pattern recog-

[5]Theoretically, the ensemble $\mathcal{X}$ should be thought of as a finite realization of a multivariate random variable $\boldsymbol{X}$. The theoretical covariance matrix is then computed in terms of expectation values $E\{(\boldsymbol{X} - E\{\boldsymbol{X}\})(\boldsymbol{X} - E\{\boldsymbol{X}\})^T\}$.

[6]It is well known that the singular value decomposition (SVD) of the dataset $\mathcal{X}$ with its mean subtracted off will also give the KL basis functions [130].

nition is to reduce the dimensionality of the feature space by discarding linear combinations of variables that have low variance and using those that have large variance for classification. This idea works well when there are many low variance subspaces that contain only noise. However, this method of feature extraction can be foiled if the high variance directions in the data set are due to within-class variance, and the differences between the classes are highest in projection on the low variance subspaces. This effect is demonstrated in section 5.5. Additional problems arise when there are too many isotropically distributed clusters (for example when clusters are at the corners of a regular simplex [48]).

Another drawback of the KL transform is its computational complexity which is $O(NM^2 + M^3)$ including the computation of the covariance matrix from N samples. This limits applications of the KL transform to problems with dimensionality $M \leq 10^3$ or perhaps $M \leq 10^4$ for the fastest computers, which prevents the KL transform from being applied in most image processing problems and multi-frame acoustic processing problems (see section 2.4.4). For this reason, the approximate KL transform was proposed by Wickerhauser [130] which is discussed in section 4.5.2. The dictionary projection pursuit algorithm developed in this thesis (section 4.4) can also be used to find approximate KL basis functions and is described in section 4.5.3. Numerical comparisons between the three algorithms are performed in section 4.5.4.

Remark: The KL transform method of feature extraction does not suffer from the same problems as Fisher's method of finding canonical variates (see section 2.2.4.4) in the neo-classical setting (*i.e.*, high dimensionality M and low sample size N) as proposed by Buckheit and Donoho [13]. When N is less than M, then the covariance matrix of $\mathcal{X}$ is guaranteed to *not* be positive definite since the N samples can span at most a plane with dimension $N-1$, but it remains non-negative definite [43]. In this case, the the first N KL basis functions still correspond to eigenvalues that are greater than zero (assuming that the points do not lie in a plane of dimensionality smaller than $N-1$ which is extremely unlikely with random noise involved), and give reasonable estimates for directions of high variance in the dataset. This remark is supported by the experimental classification results given in chapters 5 and 6 where the KL method of feature extraction performs very well even in the neo-classical setting.

Algorithm 5 Basis Functions for Karhunen-Loève Transform

Given: the following quantities

- an ensemble of signals $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^{N}$ organized into a data matrix $\boldsymbol{X} = [\boldsymbol{x}_1 \boldsymbol{x}_2 \ldots \boldsymbol{x}_N]$, where $\boldsymbol{x}_i \in \boldsymbol{\Omega} \subset \mathbf{R}^M$.
- the number of basis functions to keep $\hat{M} \leq M$.

step 1: compute the covariance matrix $\boldsymbol{\Sigma}$ for the data set using equation (4.4).

step 2: perform an eigenanalysis on $\boldsymbol{\Sigma}$ to the obtain eigenpairs $\{(\boldsymbol{\varphi}_i, \lambda_i)\}_{i=1}^{M}$.

step 3: return the $\hat{M}$ eigenvectors $\{\boldsymbol{\varphi}_i\}_{i=1}^{\hat{M}}$ corresponding to the largest eigenvalues λ_i.

4.5.2 Best Basis Approximate KL Transform

Wickerhauser introduced the best basis approximate KL transform to overcome the complexity issues of the KL transform in high dimensional spaces [130]. The best basis approximate KL transform has complexity $O(NM \log M + \hat{M}^3)$, where N is the number of samples in the dataset, M is the dimensionality of the vectors, and $\hat{M}$ is the number of basis functions that are kept. This is considerably better than the KL transform which has complexity $O(NM^2 + M^3)$ since $\hat{M}$ is usually much less than M.

Wickerhauser uses an alternative but equally valid criterion to the one given in section 4.5.1 in order to define the KL basis functions. The basis functions of the KL transform minimize the volume of the variance ellipsoid as shown in figure 4.1 which is proportional to $(\prod_{i=1}^{M} \sigma_i)^{\beta}$, where $\sigma_i^2 = \boldsymbol{\varphi}_i^T \boldsymbol{\Sigma} \boldsymbol{\varphi}_i$ and β is a constant that is dependent on the dimensionality M. It may seem counter-intuitive to minimize the volume, but Wickerhauser [130] proved that the basis functions that satisfy this criteria also maximize the projected variance. Since the log operation is a monotonic function, the minimum of the criterion function $J_K(\boldsymbol{\Phi}) = \sum_{i=1}^{M} \log \sigma_i$ is a valid definition of the KL basis functions. Wickerhauser suggests creating a variance map $\Gamma_{\boldsymbol{\Sigma}}(\gamma)$ (see equation (4.8)) for a wavelet packet or cosine packet dictionary and using the best basis algorithm (see section 3.3.4.1) to minimize J_K.

The number of basis functions to keep $\hat{M}$ can be chosen to be a fixed integer, or so that the first $\hat{M}$ basis functions contain a fixed percentage of the total variance. The

basis functions returned from the best basis algorithm are called the joint best basis for the ensemble $\mathcal{X}$. Let them be arranged as columns of the matrix $\Phi \in \mathbb{R}^{M \times \hat{M}}$. The projection of each vector $\boldsymbol{x}_i \in \mathcal{X}$ on the joint best basis is given by $\hat{\boldsymbol{\alpha}}_i = \Phi^T \boldsymbol{x}_i$, the ensemble of which has a covariance matrix given by $\Sigma_{\hat{\boldsymbol{\alpha}}}$. Therefore, the eigenvectors of $\Sigma_{\hat{\boldsymbol{\alpha}}} \in \mathbb{R}^{\hat{M} \times \hat{M}}$ give the KL basis functions of the ensemble $\{\hat{\boldsymbol{\alpha}}_i\}_{i=1}^N$ which can be used to decorrelate the projection coefficients at a cost of only $O(\hat{M}^3)$ which is usually small since $\hat{M} \ll M$. Let these basis functions be arranged as columns of a matrix $K \in \mathbb{R}^{\hat{M} \times \hat{M}}$. Then the best basis approximate KL transform is defined by the factors $K^T \Phi^T$, and the basis functions are defined to be the columns of the matrix product ΦK. This algorithm is described more formally in algorithm 6.

Algorithm 6 Basis Functions for the Best Basis Approximate KL Transform

Given: the following quantities

- an ensemble of signals $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ organized into a data matrix $\boldsymbol{X} = [\boldsymbol{x}_1 \boldsymbol{x}_2 \ldots \boldsymbol{x}_N]$, where $\boldsymbol{x}_i \in \Omega \subset \mathbb{R}^M$.
- a wavelet packet or cosine packet dictionary $\mathcal{D} = \{\varphi_\gamma | \gamma \in \Lambda\}$.
- the number of basis functions to keep $\hat{M} \leq M$.

step 1: compute the variance map $\Gamma_{\boldsymbol{\Sigma}}(\gamma)$ for the data set $\mathcal{X}$ and dictionary $\mathcal{D}$ using equation (4.8).

step 2: use the best basis algorithm defined in algorithm 1 to find the bases in the dictionary $\mathcal{D}$ that minimizes the criterion $J_K = \sum_{i=1}^{M} \log \Gamma_{\boldsymbol{\Sigma}}(\gamma_i)$. Organize the $\hat{M}$ basis functions $\{\varphi_i\}_{i=1}^{\hat{M}}$ corresponding to the largest variance $\Gamma_{\boldsymbol{\Sigma}}(\gamma_i)$ as columns of the matrix $\Phi \in \mathbb{R}^{M \times \hat{M}}$.

step 3: Find the coefficients of projection $\boldsymbol{\alpha}_i = \Phi^T \boldsymbol{x}_i \in \mathbb{R}^{\hat{M}}$ and compute the covariance matrix $\Sigma_{\hat{\boldsymbol{\alpha}}}$ using equation (4.4).

step 4: perform an eigenanalysis on $\Sigma_{\hat{\boldsymbol{\alpha}}}$ to the obtain eigenpairs $\{(k_i, e_i)\}_{i=1}^{\hat{M}}$. Organize the eigenvectors k_i into columns of a matrix $K \in \mathbb{R}^{\hat{M} \times \hat{M}}$ in descending order of e_i.

step 5: return the columns of the matrix product ΦK which represent the best basis approximate KL basis functions.

4.5.3 Dictionary Projection Pursuit Approximate KL Transform

The dictionary projection pursuit (DPP) approximate KL basis functions are found by first searching for the maximum directions of variance using the DPP algorithm, and then decorrelating the coefficients of projection as was done for the best basis method described in section 4.5.2. A projection pursuit (PP) algorithm to find the true KL basis functions could be described as follows. The first basis φ_1 is chosen so that the variance of the projection coefficients $\varphi_1^T \Sigma \varphi_1$ is maximized. The next basis function φ_2 is chosen so that the variance of the projection coefficients $\varphi_2^T \Sigma \varphi_2$ is maximized, subject to the orthogonality constraint $\varphi_2^T \varphi_1 = 0$. The next basis function φ_3 is chosen so that the variance of the projection coefficients $\varphi_3^T \Sigma \varphi_3$ is maximized, subject to the orthogonality constraints $\varphi_3^T \varphi_1 = 0$ and $\varphi_3^T \varphi_2 = 0$ ... etc.

The DPP method performs this identical search pattern, except that only a finite number of directions are searched at each iteration, and orthogonality is encouraged through penalization rather than strictly enforced. The DPP method for finding approximate KL basis functions is described formally in algorithm 7. Assuming that N is the number of vectors in the ensemble $\mathcal{X}$, M is the dimensionality of the vectors, and $\hat{M}$ is the number of basis functions that are kept, then the computational complexity of this algorithm can be computed as follows:

- Compute the variance map: $O(NM \log M)$.
- Use the DPP algorithm to find the best $\hat{M}$ basis functions: $O(\hat{M}M \log M)$.
- Diagonalize the covariance matrix of the $\hat{M}$ selected basis functions: $O(\hat{M}^3)$.

Summing these up, this algorithm has an overall complexity of $O((N+\hat{M})M \log M + \hat{M}^3)$. Since $\hat{M}$ is usually much less than N, the complexity of this algorithm is essentially the same as the best basis approximate KL algorithm discussed in section 4.5.2, both of which are dramatically more efficient than the KL algorithm which has complexity $O(NM^2 + M^3)$.

Algorithm 7 Basis Functions for the Dictionary Projection Pursuit Approximate KL Transform

Given: the following quantities

- an ensemble of signals $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ organized into a data matrix $\boldsymbol{X} = [\boldsymbol{x}_1 \boldsymbol{x}_2 \ldots \boldsymbol{x}_N]$, where $\boldsymbol{x}_i \in \boldsymbol{\Omega} \subset \mathbb{R}^M$.
- a wavelet packet or cosine packet dictionary $\boldsymbol{\mathcal{D}} = \{\boldsymbol{\varphi}_\gamma | \gamma \in \Lambda\}$.
- the number of basis functions to keep $\hat{M} \leq M$.

step 1: compute the variance map $\Gamma_{\boldsymbol{\Sigma}}(\gamma)$ for the data set $\mathcal{X}$ and dictionary $\boldsymbol{\mathcal{D}}$ using equation (4.8).

step 2: use the dictionary projection pursuit algorithm defined in algorithm 4 to maximize the criterion function $J(\boldsymbol{\varphi}_\gamma^T \boldsymbol{X}) = \Gamma_{\boldsymbol{\Sigma}}(\gamma)$. Organize the $\hat{M}$ basis function $\{\boldsymbol{\varphi}_i\}_{i=1}^{\hat{M}}$ corresponding to the largest variance $\Gamma_{\boldsymbol{\Sigma}}(\gamma_i)$ as columns of the matrix $\boldsymbol{\Phi} \in \mathbb{R}^{M \times \hat{M}}$.

step 3: Find the coefficients of projection $\boldsymbol{\alpha}_i = \boldsymbol{\Phi}^T \boldsymbol{x}_i \in \mathbb{R}^{\hat{M}}$ and compute the covariance matrix $\boldsymbol{\Sigma}_{\hat{\boldsymbol{\alpha}}}$ using equation (4.4).

step 4: perform an eigenanalysis on $\boldsymbol{\Sigma}_{\hat{\boldsymbol{\alpha}}}$ to the obtain eigenpairs $\{(k_i, e_i)\}_{i=1}^{\hat{M}}$. Organize the eigenvectors k_i into columns of a matrix $K \in \mathbb{R}^{\hat{M} \times \hat{M}}$ in descending order of e_i.

step 5: return the columns of the matrix product $\boldsymbol{\Phi} K$ which represent the dictionary projection pursuit approximate KL basis functions.

4.5.4 KL Transform Numerical Experiments

The experiment in this section uses the multiscale synthetic data set presented in section 5.6 to show how the variance accumulates on the different sets of basis functions. All of the details of this dataset are not particularly important to understand the results of this section. Any multivariate dataset that results from sampling underlying continuous waveforms would show similar results. The input data have a dimension $M = 256$, and the number of basis functions kept by each of the algorithms was $\hat{M} = 32$. The basis functions will be referred to as STD, KLT, KLTBB and KLTDPP for the standard basis functions, the KL basis functions, the best basis approximate KL basis functions and the dictionary projection pursuit approximate KL basis functions respectively. The order 2 Coiflet wavelet packet dictionary was used in the KLTBB and KLTDPP algorithms.

Referring to figure 4.2, it can be seen that in all cases the STD bases and KLT bases show respectively the slowest and fastest accumulation of variance, while the KLTBB and KLTDPP bases show an intermediate rate of accumulation. Before the coefficients are decorrelated, the KLTDPP bases show a faster accumulation of variance than the KLTBB bases. This is due to the multiscale nature of the datatset. The KLTBB algorithm must choose between the large scale subspace containing four waveforms and a small scale subspace containing eight waveforms. It chooses the small scale subspace and thus must represent the large scale waveforms with linear combinations of basis functions from other subspaces. For this reason, the accumulation of variance is slower for KLTBB than KLTDPP which does not suffer from the orthogonal subspace problem. However, once the coefficients are decorrelated, both KLTDPP and KLTBB show the same rate of variance accumulation as KLT for the first 12 basis functions, after which KLT is slightly better. These results encourage the use of the KLTBB and KLTDPP algorithms for higher dimensional datasets where KLT cannot be applied.

The experiments in chapters 5 and 6 compare the KLT, KLTBB and KLTDPP algorithms as methods for extracting features from acoustic spectra for classification purposes. It is not expected that KLTBB or KLTDPP will outperform KLT in any regime, but rather that they will be competetitive with KLT so that their use in higher dimensional problems is justified. These expectations are realized in the experimental

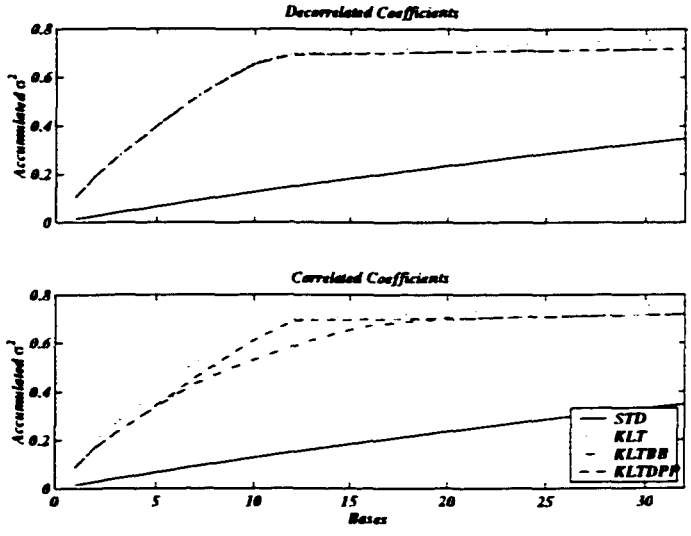

Figure 4.2. *Accumulation of variance in the multiscale dataset (section 5.6) on the standard basis functions (STD), the KL basis functions (KLT), the best basis approximate KL basis functions (KLTBB) and the dictionary projection pursuit approximate KL basis functions (KLTDPP).*

results of these chapters.

4.6 Discriminant Dictionary Projection Pursuit

Discriminant dictionary projection pursuit applies the dictionary projection pursuit algorithm (section 4.4) to the problem of waveform feature extraction for pattern recognition, and for our purposes to the problem of spectral feature extraction. Due to the flexibility of the dictionary projection pursuit algorithm, it can be applied unaltered to this problem simply by optimizing a discriminant criterion function.

4.6.1 Discriminant Criteria

A discriminant criterion $J(\boldsymbol{p}, \boldsymbol{q})$ for a two class problem is defined in terms of two sequences $\boldsymbol{p}$ and $\boldsymbol{q}$. The two sequences can be scalar sequences that define univariate criterion or vector sequences that define multivariate criterion, and they can have the same or different lengths. A class of information-theoretic discriminant criterion

functions measure the difference in distribution between two scalar sequences [70], of which symmetric relative entropy is an example (section 4.6.1.1). For these criterion functions, the sequences $\boldsymbol{p}$ and $\boldsymbol{q}$ are scalar quantities with the same length which are defined for each basis function in a proposed bases for the two classes. Therefore, in order to evaluate the criterion, a full set of basis functions is required.

A less restrictive univariate criterion called the modified Fisher criterion is presented in section 4.6.1.2. For this criterion, the sequences $\boldsymbol{p}$ and $\boldsymbol{q}$ correspond to the projection coefficients on a *single* basis function of the two classes, and thus each basis function can be analyzed individually. It should be emphasized that this type of criterion function *cannot* be optimized by Saito and Coifman's LDB algorithm but *can* be optimized by the dictionary projection pursuit algorithm.

There are of course many other types of criterion functions that can be optimized with the dictionary projection pursuit algorithm, many of which have been thoroughly analyzed by Huber [61] and Jones and Sibson [64] and thus are not discussed here.

4.6.1.1 Symmetric Relative Entropy

When the sequences are scalar sequences $\boldsymbol{p} = \{p_i\}_{i=1}^M$ and $\boldsymbol{q} = \{q_i\}_{i=1}^M$ with $\sum_{i=1}^M p_i = \sum_{i=1}^M q_i = 1$, then several discriminant criterion can be defined to measure how differently the two sequences are distributed. This is the type of discriminant criterion required for the LDB algorithm of Saito and Coifman [108]. A popular discriminant criterion of this nature is the symmetric relative entropy (SRE) which is defined as

$$J_{SRE}(\boldsymbol{p}, \boldsymbol{q}) \stackrel{\text{def}}{=} \sum_{i=1}^{M} p_i \log \frac{p_i}{q_i} + q_i \log \frac{q_i}{p_i}, \tag{4.16}$$

with the convention, $\log(0) = -\infty$, $\log(x/0) = +\infty$ for $x > 0$, $0 \cdot (\pm\infty) = 0$. For multiclass cases with sequences given by $\{\boldsymbol{p}^{(k)}\}_{k=1}^K$, the discriminant criterion is defined as the sum of the $K(K-1)/2$ pairwise combinations

$$J_{SRE}(\{\boldsymbol{p}^{(k)}\}_{k=1}^K) \stackrel{\text{def}}{=} \sum_{n=1}^{K-1} \sum_{k=n+1}^{K} J_{SRE}(\boldsymbol{p}^{(n)}, \boldsymbol{p}^{(k)}), \tag{4.17}$$

In order to use this criterion function with their LDB algorithm, Saito and Coifman define a normalized energy map for each class $\Gamma^{(k)}(\gamma) = \Gamma_{\boldsymbol{Q}}^{(k)}(\gamma)/\beta^{(k)}$, where $\Gamma_{\boldsymbol{Q}}^{(k)}(\gamma)$

is defined in equation (4.7), and

$$\beta^{(k)} = \frac{\sum_{n=1}^{N^{(k)}} \|\boldsymbol{x}_i^{(k)}\|^2}{N^{(k)}} \tag{4.18}$$

is the average energy of the signals from class k. Since the SRE measure is additive, it can be computed for each basis individually and the SRE of a subspace $\boldsymbol{\Omega}$ spanned by orthogonal basis vectors $\{\boldsymbol{\varphi}_\gamma | \gamma \in \Lambda_0\}$ can be computed by summing the SRE values of each basis

$$J_{SRE}(\Lambda_0) = \sum_{\gamma \in \Lambda_0} J_{SRE}\left(\{\Gamma^{(k)}(\gamma)\}_{k=1}^{K}\right). \tag{4.19}$$

Due to this additivity, LDB can use the best basis algorithm (section 3.3.4.1) to maximize the criterion J_{SRE} over a wavelet packet or cosine packet dictionary. Many other criterion functions of this nature are given by Saito and Coifman but symmetric relative entropy seems to be the favorite and thus was implemented in this thesis.

4.6.1.2 Modified Fisher Criterion

Although the dictionary projection pursuit algorithm developed in this thesis can be used with criterion like symmetric relative entropy, it can also be used to optimize other criterion that LDB cannot. In general, criterion functions used for the dictionary projection pursuit algorithm are defined to be functions of $\boldsymbol{\varphi}_\gamma^T \boldsymbol{X}$, the coefficients of projection of the dataset on a given basis in the dictionary $\boldsymbol{\mathcal{D}} = \{\boldsymbol{\varphi}_\gamma | \gamma \in \Lambda\}$. The criterion is computed for each basis function independently.

Given a training set $\mathcal{L}_t = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ having K classes $\{\boldsymbol{\omega}^{(k)}\}_{k=1}^{K}$, the quantities $\Gamma_{\boldsymbol{\mu}}^{(k)}(\gamma)$, and $\Gamma_{\boldsymbol{\Sigma}}^{(k)}(\gamma)$ can be computed using equations (4.6) and (4.8). Assume that the classes have *a priori* probabilities given by $p(\boldsymbol{\omega}^{(k)})$, then the following maps can be defined. The overall mean map is defined by

$$\Gamma_{\boldsymbol{\mu}}(\gamma) \stackrel{\text{def}}{=} \sum_{k=1}^{K} p(\boldsymbol{\omega}^{(k)}) \Gamma_{\boldsymbol{\mu}}^{(k)}(\gamma), \tag{4.20}$$

the within-class variance map is defined by

$$\Gamma_{\boldsymbol{\Sigma}_W}(\gamma) \stackrel{\text{def}}{=} \sum_{k=1}^{K} p(\boldsymbol{\omega}^{(k)}) \Gamma_{\boldsymbol{\Sigma}}^{(k)}(\gamma), \tag{4.21}$$

the between-class variance map is defined by

$$\Gamma_{\Sigma_B}(\gamma) \stackrel{\text{def}}{=} \sum_{k=1}^{K} p(\omega^{(k)})(\Gamma_{\mu}^{(k)}(\gamma) - \Gamma_{\mu}(\gamma))^2, \tag{4.22}$$

and the total variance map is defined by

$$\Gamma_{\Sigma_T}(\gamma) \stackrel{\text{def}}{=} \Gamma_{\Sigma_B}(\gamma) + \Gamma_{\Sigma_W}(\gamma). \tag{4.23}$$

The modified fisher criterion is then defined as

$$J_{MF}(\boldsymbol{\varphi}_{\gamma}^{T}\boldsymbol{X}) = \frac{\Gamma_{\Sigma_B}(\gamma)}{\Gamma_{\Sigma_W}(\gamma)} \text{sig}\left(\frac{\sqrt{\Gamma_{\Sigma_T}(\gamma)}}{\sigma_{est}}, \psi, \xi\right), \tag{4.24}$$

where σ_{est} is the estimated standard deviation of the white noise in the dataset, and sig is the generalized sigmoid function

$$\text{sig}(x, \psi, \xi) = \frac{1}{1 + \exp[-\psi(x - \xi)]}, \tag{4.25}$$

which is plotted in figure 4.3. The parameter ψ controls the slope of the function while ξ defines the offset. For the work in this thesis, $\psi = 1$ and $\xi = 1.5$, and σ_{est} was estimated using the median value of $\Gamma_{\Sigma_T}(\gamma)$ on the first level wavelet coefficients (*i.e.*, $s = 1$, $f = 1$).

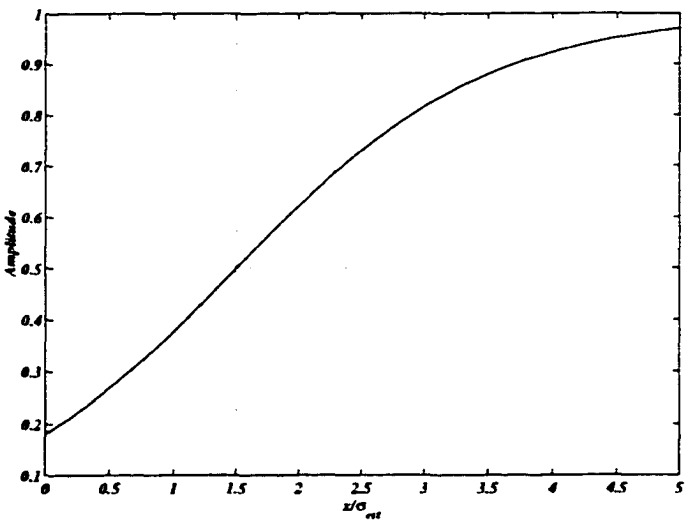

Figure 4.3. *The generalized sigmoid activation function with* $\psi = 1$ *and* $\xi = 1.5$.

The first part of the criterion $\Gamma_{\Sigma_B}(\gamma)/\Gamma_{\Sigma_W}(\gamma)$ is based on a univariate version of Fisher's multiclass criterion

$$J(\boldsymbol{\varphi}_\gamma^T \boldsymbol{X}) = \frac{\boldsymbol{\varphi}_\gamma^T \boldsymbol{\Sigma}_B \boldsymbol{\varphi}_\gamma}{\boldsymbol{\varphi}_\gamma^T \boldsymbol{\Sigma}_W \boldsymbol{\varphi}_\gamma}, \tag{4.26}$$

where $\boldsymbol{\Sigma}_B$ and $\boldsymbol{\Sigma}_W$ are defined in equations (2.29) and (2.27). In words, this criterion measures the ratio of the between-class variance to the within-class variance along the basis function $\boldsymbol{\varphi}_\gamma$. For the two class case with equal *a priori* probabilities, it reduces to the ratio of the squared difference between the means of the two classes and the average variance in the two classes.

The sigmoid part of the criterion is an attempt to overcome the problem of small variance subspaces fooling Fisher's criterion. The idea is that if there is white noise in the dataset with $\sigma^2 \approx \sigma^2_{est}$, then all basis functions that do not have variance $\sigma^2_\gamma = \boldsymbol{\varphi}_\gamma^T \boldsymbol{\Sigma}_T \boldsymbol{\varphi}_\gamma$ significantly larger than σ^2 should not be used as features. This is the same idea as using the KL transform for feature extraction except that when $\sigma^2_\gamma \gg \sigma^2$ the sigmoid function flattens out and the criterion function becomes Fisher's criterion.

4.6.2 Discriminant Dictionary Projection Pursuit Features

This section takes a look at typical features that the discriminant dictionary projection pursuit (DDPP) algorithm extracts with the modified Fisher criterion. The examples are all taken from the datasets studied in chapters 5 and 6. For the synthetic data, the variables were set to $SNR = 10$, $f_s = 64$, $N = 64$, and $Bases = 5$. For the recorded data, the variables were set to $N = 128$ and $Bases = 25$. To understand this discussion it would be beneficial for the reader to quickly look at the setup for each experiment in sections 5.4, 5.5, 5.6, 6.2.2, and 6.3.2.

The first four features extracted for each of the synthetic datasets studied in chapter 5 are plotted in figure 4.4. For the triangular dataset (section 5.4), the first feature represents the difference between the a_1 and a_2 waveforms which is also approximately the first KL basis function. The remaining features represent differences between the a_1 and a_3 waveforms and the a_2 and a_3 waveforms.

For the common variance dataset (section 5.5), the part of the waveform that has constant mean and variance between the classes is ignored as one would expect since

it provides no discriminatory information. All the features are in the region of the a_6 waveform which is the only waveform providing any discriminant power.

For the multiscale dataset (section 5.6), the oscillating pattern of the mean in the first eight small scale waveforms was picked up in the first two features, and the oscillating pattern of the mean in the four large scale waveforms was picked up in the third feature. The last feature matches a single small scale waveform closely.

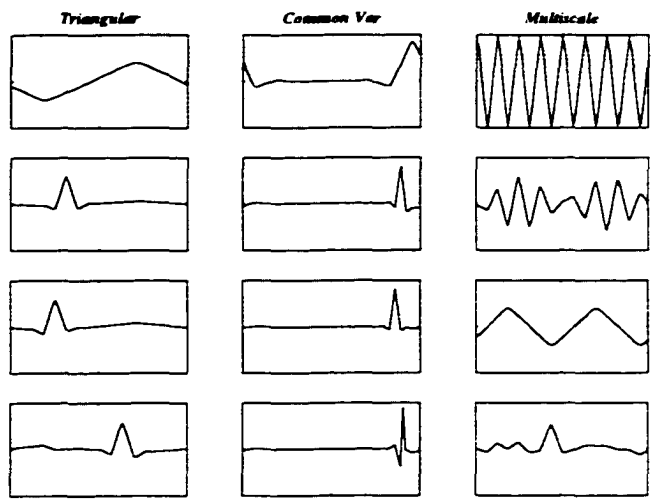

Figure 4.4. *Features extracted by the discriminant dictionary projection pursuit algorithm with the modified Fisher criterion for the synthetic datasets discussed in chapter 5. Each column of plots represents features extracted for a given dataset ordered with the best feature at the top.*

The first five features extracted for each of the recorded datasets studied in chapter 6 are plotted in figure 4.5. For the madras dataset (section 6.2.2), it is interesting to note that the features do not match the pattern of 1/3 octave passbands at all, which indicates that a 1/3 octave filter bank is not the best pre-processor for recognizing typical noise monitoring sounds. This is significant since the noise monitoring community currently considers 1/3 octave features the state of the art (see section 6.2.2). While the first three features pick up large scale differences between the classes, the last two features pick up two high resolution spectral peaks in the train class (see figure 6.3).

For the phoneme dataset (section 6.3), the features extracted confirm the long

known fact that most of the information used to discriminate phoneme sounds is located in the lower frequencies of the spectrum. It is interesting to note that the features used to discriminate phonemes and the features used to discriminate noise monitoring sounds are quite different. This suggests that the feature extraction techniques that have been developed by the speech recognition community over the years by trial and error should *not* be blindly applied to other problems involving sound recognition.

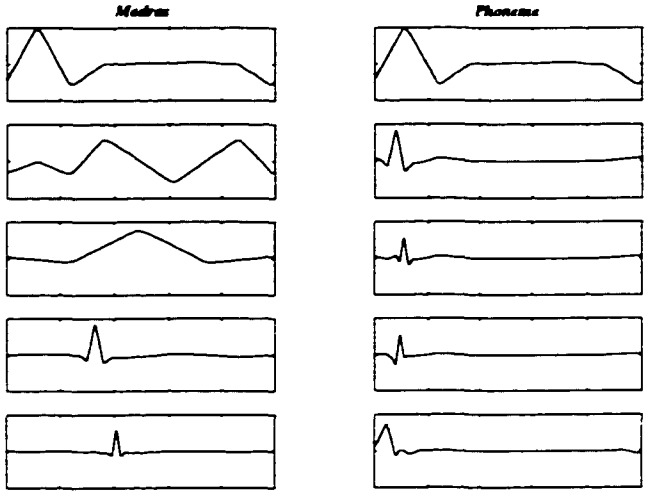

Figure 4.5. *Features extracted by the discriminant dictionary projection pursuit algorithm with the modified Fisher discriminant function for the recorded datasets discussed in chapter 6. Each column of plots represents features extracted for a given dataset ordered with the best feature at the top.*

This section showed how easy it is to interprete the features that are returned by the discriminant dictionary projection pursuit algorithm. In fact, this algorithm has been applied simply for the purpose 'seeing' what features distinguish various classes of sound spectra on numerous occasions since it was written. It is therefore a valuable exploratory tool as well as good algorithm for extracting features for pattern recognition. The last section of this chapter describes each of the feature extraction algorithms that are compared in chapter 5 and 6.

4.7 Algorithms used for Experiments

4.7.1 Standard Bases (STD)

The STD algorithm does not use a feature extraction process. It simply passes the input features unaltered to the classifier. This is true even when the *Bases* variable changes in the sub-experiments.

4.7.2 Discriminant Dictionary Projection Pursuit with Modified Fisher Criterion (DDPPMF)

The DDPPMF feature extraction method uses the discriminant dictionary projection pursuit algorithm (section 4.6) to extract features that maximize the modified Fisher criterion (section 4.6.1).

4.7.3 Discriminant Dictionary Projection Pursuit with Symmetric Relative Entropy Criterion (DDPPSRE)

The DDPPSRE feature extraction method uses the discriminant dictionary projection pursuit algorithm (section 4.6) to extract features that maximize the symmetric relative entropy criterion (section 4.6.1).

4.7.4 KL Transform (KLT)

The KLT feature extraction method uses the KL transform (section 4.5.1) to extract features corresponding to the highest variance.

4.7.5 Best Basis Approximate KL Transform (KLTBB)

The KLTBB feature extraction method uses the best basis approximate KL transform (section 4.5.2) to extract features corresponding to the highest variance.

4.7.6 Dictionary Projection Pursuit Approximate KL Transform (KLTDPP)

The KLTDPP feature extraction method uses the dictionary projection pursuit approximate KL transform (section 4.5.3) to extract features corresponding to the highest variance.

4.7.7 Local Discriminant Bases (LDB)

The LDB feature extraction method developed by Saito and Coifman [108] uses the best basis algorithm to extract features that maximize the symmetric relative entropy criterion (section 4.6.1). Other additive criterion are possible but were not implemented.

4.7.8 Discriminant Pursuit (DP)

The DP feature extraction method developed by Buckheit and Donoho [13] uses the matching pursuits algorithm without backfitting (section 3.3.4.2) on the differences between the class means (contrasts) to extract features. This algorithm iteratively selects features that maximize the correlation with one of the $K(K-1)/2$ contrasts, where K is the number of classes. This technique does not use the covariance of the dataset in any way.

4.7.9 Weighted Discriminant Pursuit (WDP)

The WDP feature extraction method developed by Buckheit and Donoho [13] is a modified version of the DP method. The contrasts between the class means are scaled by the inverse within-class covariance[7] $\mathbf{\Sigma}_W^{-1}$ and the correlation with each contrast is weighted by the inverse squared Mahalanobis distance [123] between the class means $D_{ij}^{-2} = \left[(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \mathbf{\Sigma}_W^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)\right]^{-1}$. As discussed in section 4.3.2, using the inverse

[7]Actually they use the pooled covariance which is only slightly different from the within-class covariance in that the pooled covariance tries to correct for bias when averaging the covariance matrices from each class.

of the within-class covariance matrix is a questionable operation since in the neo-classical setting, where N is often less than M, the estimated within-class covariance is guaranteed to be singular [43]. Indeed, this algorithm performs very poorly in the neo-classical setting.

Chapter 5

Experimental Results for Synthetic Data

5.1 Introduction

The experiments in this chapter use the signal model described in section 5.2 to create synthetic datasets to show the relative strengths and weaknesses of each of the feature extraction algorithms described in section 4.7. In all cases, Fisher's LDA described in section 2.2.4.4 was used as a classifier. This classifier was chosen due to its widespread use and acceptance as a robust classifier. Simulations with other classifiers such as linear and quadratic Gaussian plug-in, and CART (see section 2.2.4) show the same performance trends for each of the feature extraction techniques described here, so they are not presented.

Comparing classification rates of different feature extraction algorithms while keeping all other variables constant is a useful measure of their performance. By varying the parameters of each feature extraction algorithm in the same way it is possible to see when some feature extraction algorithms fail and when some shine. Unfortunately, there is rarely a universally superior algorithm for specific data analysis tasks, and feature extraction is no different. Some algorithms do very well under some conditions and very poorly under others. This is why it is so important to study a broad range of parameter values as is done in this chapter. Using synthetic data gives complete understanding of the problem domain. The signal model developed in section 5.2 represents an easy to understand and very useful model for building synthetic datasets with desired properties. Most importantly, it allows the Bayes error rate for the problem to be computed so that each feature extraction algorithm

can be compared to a theoretically optimal solution.

5.2 Signal Model

The signal model used in this work provides a flexible way of defining signals which is useful for waveform pattern recognition problems in the following ways:

1. It provides a foundation for the theoretical development of a waveform pattern recognition system.
2. It provides formulae to create synthetic signals with varying sampling rates and signal to noise ratios for testing a waveform pattern recognition system.
3. It allows the Bayes error rate to be easily computed.

5.2.1 Definition

The model is defined as

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{\alpha} + \boldsymbol{\epsilon}, \tag{5.1}$$

where $\boldsymbol{A} \in \mathbb{R}^{M \times \hat{M}}$ has $\hat{M}$ linearly independent deterministic waveforms as columns, $\boldsymbol{\alpha} \in \mathbb{R}^{\hat{M}}$ is a Normally distributed random variable $\mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}})$ with mean $\boldsymbol{\mu}_{\boldsymbol{\alpha}} \in \mathbb{R}^{\hat{M}}$ and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}} \in \mathbb{R}^{\hat{M} \times \hat{M}}$, and $\boldsymbol{\epsilon} \in \mathbb{R}^{M}$ is a Normally distributed pure noise term $\mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}})$ with zero mean and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} \in \mathbb{R}^{M \times M}$. The noise term is usually taken to be white with a covariance matrix given by $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} = \sigma^2 \boldsymbol{I}$, where σ^2 is a scalar giving the variance of the white noise, and $\boldsymbol{I} \in \mathbb{R}^{M \times M}$ is the identity matrix. In words, a signal is composed of a Normal distribution of deterministic waveforms plus some noise.

The $\boldsymbol{A}\boldsymbol{\alpha}$ term contains the information about the underlying system being studied, while the $\boldsymbol{\epsilon}$ term contains random noise that is not related to the system, but is usually due to the measurement process or some other form of contamination. The waveforms are defined as continuous functions on $t = [0, 1]$ which are sampled at a given rate f_s to produce $M = f_s$ samples. This allows the same problem to be studied with a variety of sampling rates. The notation t and f_s are used due to the familiar concept of sampling a time domain signal, but the model does not require this. In fact, in the two recorded sound experiments studied in chapter 6, the x-axis is frequency.

5.2.2 Relationships and Transformations

The following results can be easily derived by applying expectations to equation (5.1) (see Brown and Hwang [11] for example). The mean of $\boldsymbol{x}$ is given by

$$\boldsymbol{\mu}_{\boldsymbol{x}} = E\{\boldsymbol{x}\} = \boldsymbol{A}\boldsymbol{\mu}_{\boldsymbol{\alpha}}, \tag{5.2}$$

and the covariance matrix of $\boldsymbol{x}$ is given by

$$\boldsymbol{\Sigma}_{\boldsymbol{x}} = E\left\{(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}})(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}})^T\right\} = \boldsymbol{A}\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}\boldsymbol{A}^T + \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}. \tag{5.3}$$

In practice, the signals $\boldsymbol{x}$ will be observed, and their orthogonal projection onto the subspace spanned by the columns of $\boldsymbol{A}$ will be desired. The coefficients of the projection are given by

$$\boldsymbol{z} = \boldsymbol{H}\boldsymbol{x}, \tag{5.4}$$

where $\boldsymbol{H} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T$ is the coefficient matrix for the subspace span$\{\boldsymbol{A}\}$, and $\boldsymbol{z} \in \mathbb{R}^M$ has a dimension equal to the number of columns in $\boldsymbol{A}$.

The following formulae give the parameters of $\boldsymbol{z}$ in terms of the known (*i.e.*, defined in the signal model) parameters of $\boldsymbol{\alpha}$. The mean of $\boldsymbol{z}$ is given by

$$\boldsymbol{\mu}_{\boldsymbol{z}} = E\{\boldsymbol{z}\} = \boldsymbol{\mu}_{\boldsymbol{\alpha}}, \tag{5.5}$$

and the covariance matrix of $\boldsymbol{z}$ is given by

$$\boldsymbol{\Sigma}_{\boldsymbol{z}} = E\left\{(\boldsymbol{z} - \boldsymbol{\mu}_{\boldsymbol{z}})(\boldsymbol{z} - \boldsymbol{\mu}_{\boldsymbol{z}})^T\right\} = \boldsymbol{\Sigma}_{\boldsymbol{\alpha}} + \boldsymbol{H}\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\boldsymbol{H}^T. \tag{5.6}$$

Since a linear transformation of a multivariate Normal random variable is also a multivariate Normal random variable, it follows that $\boldsymbol{\alpha}$, $\boldsymbol{x}$, and $\boldsymbol{z}$ are all Normally distributed with mean and covariance matrices given above (see Flury [43], Theorem 3.2.1 for a proof). These are useful results because now, noise can be added in the $\boldsymbol{x}$-domain, and its effect is known in the $\boldsymbol{z}$-domain. All the information provided by the signal model given in equation (5.1) is contained in the distribution of the random vector $\boldsymbol{z}$.

5.2.3 Signal to Noise Ratio

The signal to noise ratio for a given signal model is defined as

$$SNR = 10 \log \left(\frac{\mathbf{tr}\left(\boldsymbol{\Sigma}_{\boldsymbol{x}}\right) + \|\boldsymbol{\mu}_{\boldsymbol{x}}\|^2}{\mathbf{tr}\left(\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\right)} \right) \text{dB}, \tag{5.7}$$

where **tr** is the trace operation which sums the diagonal elements, and the other symbols are defined above. This is the ratio of the *total* expected signal energy with the *total* expected noise energy.

5.2.4 Normalization

The signal models in this work are normalized so that the expected total energy of the signal is unity,

$$E_{total} = E_{signal} + E_{noise} = \mathbf{tr}\left(\boldsymbol{\Sigma}_{\boldsymbol{x}}\right) + \|\boldsymbol{\mu}_{\boldsymbol{x}}\|^2 + \mathbf{tr}\left(\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\right) = 1. \tag{5.8}$$

5.2.5 Monte Carlo Estimation of the Bayes Error Rate

The Bayes error rate, as explained in section 2.2.3.1, can be computed very easily using the signal model given by equation (5.1) for each class. Let

$$\boldsymbol{x}^{(k)} = \boldsymbol{A}\boldsymbol{\alpha}^{(k)} + \boldsymbol{\epsilon}, \tag{5.9}$$

be the signal model for each class $\boldsymbol{\omega}^{(k)}$ with corresponding *a priori* probabilities $p(\boldsymbol{\omega}^{(k)})$. The matrix of deterministic waveforms $\boldsymbol{A}$ is the same for all classes but a given class may only use a subset of the waveforms to define its signals. This is accomplished by inserting zeros at the appropriate place in $\boldsymbol{\mu}_{\boldsymbol{\alpha}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}$. Notice that $\boldsymbol{\epsilon}$ is the same for all classes. This is a reasonable assumption since the noise that is introduced into the signals is usually caused by the measurement process which is typically the same for all classes. It is preferable to work with the random variable $\boldsymbol{z} \in \mathbb{R}^{\hat{M}}$ as defined by equation (5.4), rather than $\boldsymbol{x} \in \mathbb{R}^{M}$ since $\hat{M}$ is usually much smaller than M. The algorithm used to compute the Bayes error rate for a given signal model is described in algorithm 8.

Typical values for N and R used in this thesis were 500 and 50 respectively. The conditional probability density functions used for the signal model are Normal, so the

Algorithm 8 Monte Carlo Bayes Error Rate Estimation

Given: the following

- K classes $\omega^{(k)}$ with known *a priori* probabilities $p(\omega^{(k)})$ and known signal model parmaters
  - $\boldsymbol{\mu}_{\alpha}^{(k)} \in \mathbf{R}^{\hat{M}}$ and $\boldsymbol{\Sigma}_{\alpha}^{(k)} \in \mathbf{R}^{\hat{M}\times\hat{M}}$ which define the Normal parameters for the distribution of $\boldsymbol{\alpha}^{(k)}$.
  - $\boldsymbol{A} \in \mathbf{R}^{M\times\hat{M}}$ which defines the deterministic waveforms for the problem.
  - $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} \in \mathbf{R}^{M\times M}$ which gives the covariance matrix for the zero mean Normally distributed noise.
- N - the number of samples to use for each trial.
- R - the number of times to repeat the trial.

step 1: Create N random samples as follows:

1. Partition the unit interval $[0, 1]$ with the *a priori* probabilities $p(\omega^{(k)})$, and let the value of a uniform deviate u on the partition define the class $y_i = \omega^{(k)}$.
2. use the selected class index k to create a random vector $\boldsymbol{\alpha}_i$ with a distribution given by $\mathcal{N}(\boldsymbol{\mu}_{\alpha}^{(k)}, \boldsymbol{\Sigma}_{\alpha}^{(k)})$, and a random vector $\boldsymbol{\epsilon}_i$ with a distribution given by $\mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}^{(k)})$. Set $\boldsymbol{x}_i = \boldsymbol{A}\boldsymbol{\alpha}_i + \boldsymbol{\epsilon}_i$.
3. store the pairs as a set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$.

step 2: Project each vector $\boldsymbol{x}_i$ onto the subspace $\text{span}\{\boldsymbol{A}\}$ to obtain $\boldsymbol{z}_i = \boldsymbol{H}\boldsymbol{x}_i = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{x}_i$ and classify using $\hat{y}_i = \underline{\boldsymbol{D}}(\boldsymbol{z}_i)$ where $\underline{\boldsymbol{D}}$ is the Bayes classifier defined in equation (2.9) with conditional probability density functions $p(\boldsymbol{z}|\boldsymbol{\omega}^{(k)})$ being Normal with parameters given by $\boldsymbol{\mu}_{\boldsymbol{z}}^{(k)}$ and $\boldsymbol{\Sigma}_{\boldsymbol{z}}^{(k)}$ (computed using equations (5.5) and (5.6)).

step 3: Compute the estimated error $\boldsymbol{E}_r = N^{-1}\sum_{i=1}^{N}\mathcal{I}(\hat{y}_i \neq y_i)$, where $\mathcal{I}$ is the indicator function which returns a one when the argument is true and a zero otherwise.

step 4: Repeat steps 1–3 R times and compute the median error $\boldsymbol{E}_{est} = \text{med}(\{\boldsymbol{E}_r\}_{r=1}^{R})$, which is the estimated Bayes error rate for the problem.

multivariate deviates in step 1 of the algorithm are easily computed by generating univariate Normal deviates along the eigenvectors of the covariance matrix (see Press et al. [101] for algorithms to generate uniform and Normal deviates).

The Bayes error rate was computed for each synthetic data set studied in this chapter which gives the absolute lower limit on the performance of any of the feature extraction/classifier combinations. The Bayes error rate is plotted as a solid line in all the plots.

5.3 Experimental setup

Each feature reduction technique was evaluated in terms of its holdout error rate (section 2.2.6.2) on a scale of 0–1 as a function of the following parameters,

N - the number of frames used to train the adapted feature extraction algorithm and classifier,

SNR- the signal to noise ratio of the signal as defined in section 5.2.3,

f_s – the sampling rate of the waveform as defined in section 5.2.1, which is equivalent to the dimensionality of the problem since the waveforms are defined on the interval $[0, 1]$,

Bases- the number of bases that were kept by the feature extraction algorithm. Note that the STD method keeps all the basis functions regardless of the value of *Bases*,

which are all known to be important factors with the respect to the performance of a pattern recognition system.

Since it is not feasible to test all combinations of parameters, the common technique of choosing reasonable values for all of the parameters, and then varying each parameter individually in discrete steps while keeping the others fixed was adopted. So within each experiment in this chapter there are four sub-experiments called N, SNR, f_s and *Bases* where the name of the experiment indicates the parameter that is being varied. The parameter values for each sub-experiment are shown in Table 5.1[1].

[1]The only exception is for the multiscale dataset which does not use $f_s = 16$ since the signals are not defined for this number of samples.

sub-experiment	Parameter Values
N	$\{16, 32, 64, 128, 256\}$
SNR	$\{-5, 0, 5, 10, 15, 20\}$
f_s	$\{16, 32, 64, 128, 256\}$
$Bases$	$\{1, 2, 3, 5, 7, 9\}$

Table 5.1. *Parameter values used for sub-experiments*

When the parameters are not being varied, they are fixed at $N = 64$, $SNR = 10$, $f_s = 64$, and $Bases = 5$.

5.3.1 Box Plots

Since the holdout error rate is a statistic (*i.e.*, it is an estimate of the true error rate), 50 experiments were performed for each combination of parameters given in table 5.1 to obtain a distribution of error rates. Rather than simply plot the mean and standard deviation of the distribution, as is commonly done in the engineering and scientific literature, box plots are used in this thesis which are very common in the statistical literature. A box plot communicates much more useful information than an error bar plot, since it uses robust statistics and doesn't make any assumptions about the underlying distribution.

Consider the set of data given by

$$x = \{0, 3, 5, 10, 12, 12, 14, 15, 15, 15, 15, 16, 16, 17, 25, 31, 80, 90, 95, 99\} \qquad (5.10)$$

which is plotted in figure 5.1 both as a box plot and as an error bar plot. Clearly, most of the data is clustered around 15, with a few points deviating significantly $\{0, 3, 5, 25, 31\}$, and a few outliers with much higher values $\{80, 90, 95, 99\}$. The mean and standard deviation of this set is given by 29.25 and 32.5 respectively. These numbers do not describe the distribution very well. However, rank statistics do a very nice job of describing the data [44], which is what the box plot is based on. The lower quartile, median and upper quartile are given by 12, 15 and 28 respectively, which are represented by the lower bound, middle line and upper bound of the box. The whiskers in the plot extend to the data points that are within 1.5 times the interquartile distance ($1.5 \cdot (28 - 12) = 24$) from the top and bottom of the box.

Therefore, on the bottom of the box, the whisker extends to 0, and on the top of the box, the whisker extends to 31. Any data that is outside the whiskers are plotted as '+' symbols, and represent outliers.

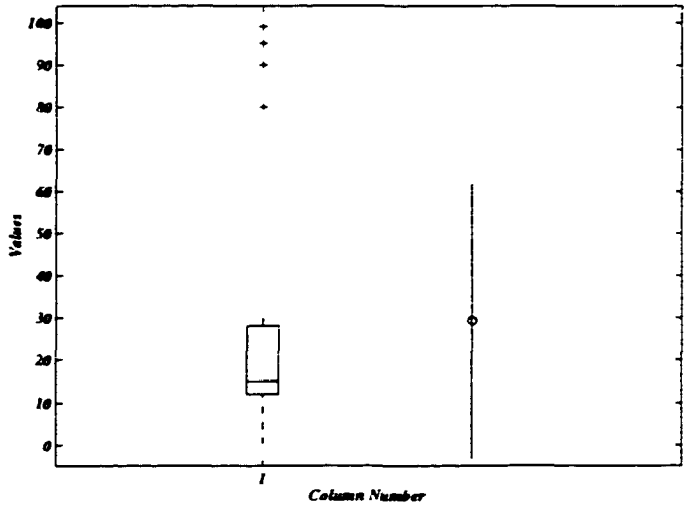

Figure 5.1. *Box plot and error bar plot of hypothetical data given in equation* (5.10).

5.4 Triangular Waveforms

This synthetic dataset was inspired by the well-known and extensively studied 'waveform' dataset of Breiman et al. [9]. The only difference is that the triangular waveform data used here has the trivial extension of allowing for different sampling rates and noise levels. In addition, instead of using uniform random variables to combine waveforms in a given class, the signals here use a Normal distribution with a correlation coefficient of -0.95, which is required to fit our signal model described in section 5.2.

There are three columns in the $\boldsymbol{A}$ matrix which are defined as

$$a_k(nT) = \kappa \max\left(1 - s|nT - t_k|, 0\right), \tag{5.11}$$

for $k = \{1, 2, 3\}$, where $T = 1/f_s$ is the sampling interval, $s = 1/0.28$, $t_1 = 0.28$, $t_2 = 0.67$, $t_3 = 0.48$, and κ is a constant chosen so that $\|a_k\| = 1$. These waveforms are plotted in figure 5.2.

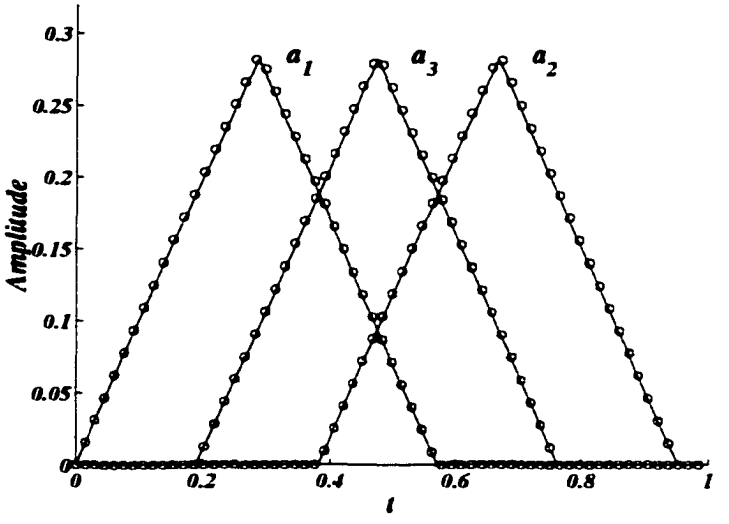

Figure 5.2. *Triangular waveforms for the* $\boldsymbol{A}$ *matrix. The solid line shows the continuous version of the waveform, and the circles represent the sampled version with* $f_s = 64$.

The Normal density parameters of $\boldsymbol{\alpha}$ for each class are given as,

$$\boldsymbol{\mu}_{\boldsymbol{\alpha}}^{(1)} = \begin{bmatrix} 0.5 & 0.5 & 0 \end{bmatrix}^T \qquad \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{(1)} = \sigma_{\boldsymbol{\alpha}}^2 \begin{bmatrix} 1 & \rho & 0 \\ \rho & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.12a}$$

$$\boldsymbol{\mu}_{\boldsymbol{\alpha}}^{(2)} = \begin{bmatrix} 0.5 & 0 & 0.5 \end{bmatrix}^T \qquad \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{(2)} = \sigma_{\boldsymbol{\alpha}}^2 \begin{bmatrix} 1 & 0 & \rho \\ 0 & 0 & 0 \\ \rho & 0 & 1 \end{bmatrix} \tag{5.12b}$$

$$\boldsymbol{\mu}_{\boldsymbol{\alpha}}^{(3)} = \begin{bmatrix} 0 & 0.5 & 0.5 \end{bmatrix}^T \qquad \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{(3)} = \sigma_{\boldsymbol{\alpha}}^2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & \rho \\ 0 & \rho & 1 \end{bmatrix}, \tag{5.12c}$$

where $\sigma_{\boldsymbol{\alpha}}^2 = 0.3$, and $\rho = -0.95$. Finally, the noise added in this model is white and Normally distributed as given by $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$. Typical waveforms from each of the classes are plotted in figure 5.3. In words the first class is a highly anti-correlated linear combination of waveforms a_1 and a_2, the second class is a highly anti-correlated linear combination of waveforms a_1 and a_3, and the third class is a highly anti-correlated linear combination of waveforms a_2 and a_3.

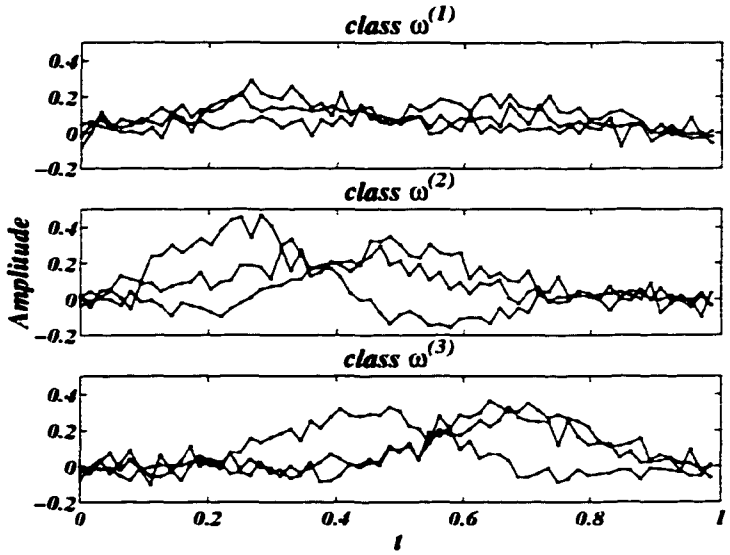

Figure 5.3. *Typical waveforms from the triangular waveform classes with* $f_s = 64$ *and* $SNR = 10$.

5.4.1 Experimental Results

The experimental results of applying each feature extraction algorithm (section 4.7) to the triangular waveform dataset for each sub-experiment (section 5.3) are plotted at the end of this section in figures 5.4–5.7. A tabular presentation of the results for each sub-experiment are given in tables 5.2–5.5. The results are discussed in the following sections.

5.4.1.1 sub-experiment *N*

The Bayes error rate for this sub-experiment is 0.049 independent of N.

For $N = \{16, 32\}$ KLT outperforms every other feature extraction technique. This is perhaps not surprising since this feature set is ideally suited for KLT feature extraction. That is, all the discriminant information is contained in a plane, and thus any variance out of this plane is just white noise.

All other algorithms have reasonable results with error rates only slightly higher than KLT, except for STD and WDP. The reason for the poor performance of these algorithms for small N is that both use the full size 64×64 within-class covariance matrix Σ_W which is guaranteed to be singular or at least badly scaled. The STD

method doesn't reduce the dimension of the feature vector at all, so Fisher's LDA (see section 2.2.4.4) uses the poorly scaled $\mathbf{\Sigma}_W$ to solve the generalized eigenvalue problem and thus has large errors. The WDP algorithm attempts to scale the contrasts between the class means by $\mathbf{\Sigma}_W^{-1}$ (see section 4.7.9) which when $\mathbf{\Sigma}_W$ is poorly scaled produces erroneous results.

For $N = \{64, 128, 256\}$, many techniques (DDPPMF, LDB, DP, KLTBB, and KLTDPP) catch up to KLT obtaining error rates of ~ 0.13. DDPPSRE only reaches an error rate of ~ 0.18, the reasons for which are discussed further in the summary. The STD and WDP start to show improvements in this regime, but never achieve error rates as low as the other techniques.

5.4.1.2 sub-experiment *SNR*

The Bayes error rate for this sub-experiment starts at 0.289 for $SNR = -5$, decreases quickly at first, and then starts to level off, achieving a final error rate of 0.016 for $SNR = 20$.

The STD and WDP algorithms perform very poorly while all the other 'good' algorithms achieve similar error rates for all values of *SNR*. An exception is the DDPPSRE algorithm which shows slightly higher error rates than the other 'good' algorithms, the reasons for which are discussed in the summary.

A common trend that is observed is that the error rates for each of the algorithms are closer to the Bayes error rate for low *SNR* values. This makes sense because the covariance matrices of each of the classes without noise are very different which violates the assumption of Fisher's LDA classifier. As white noise is added to the signal (*i.e.*, the *SNR* decreases) the covariance matrices become more similar to one another in congruence with the assumption of the classifier which then performs better.

5.4.1.3 sub-experiment f_s

The Bayes error rate for this sub-experiment starts at 0.102 for $f_s = 16$ and gradually drops to 0.024 for $f_s = 256$.

The STD and WDP algorithms show an increase in error rate as f_s increases while the remainder of the algorithms show a decrease. The reason for this is that both

STD and WDP try to estimate the full $f_s \times f_s$ within-class covariance matrix $\mathbf{\Sigma}_W$ which becomes more difficult to do as f_s increases since the number of samples in each class is fixed at 64. The other techniques, like the Bayes error rate, improve as f_s increases because the waveforms are sampled at a higher rate which provides more discriminant information about the problem.

The median error rates for all the techniques except STD and WDP are similar for all values of f_s. An exception is the DDPPMF algorithm which shows slightly higher median error rates and also a large number of outliers with high error rates for $f_s = \{64, 128, 256\}$. The reason for this is unclear.

5.4.1.4 sub-experiment *Bases*

The Bayes error rate for this sub-experiment is 0.050 for all values of *Bases*.

The STD and WDP algorithms show consistently poor results for all values of *Bases*. The reason for this is the same as discussed above. The rest of the algorithms show consistently smaller error rates as *Bases* increases, with the biggest drop occuring when *Bases* goes from 1 to 2 and the minimum error rate being ~ 0.14 for *Bases* ≥ 3. KLT actually achieves this error rate for *Bases* $= 2$. Since all the discriminant information for this dataset lies in a plane, this result is not surprising. The WDP algorithm shows consistently poor result with some improvement as *Bases* increases and since the STD algorithm is not modified as *Bases* changes, it obtains consistent error rates of ~ 0.30.

An exception to the general decline in error rate as *Bases* increases is the DP algorithm which shows a large dispersion in the error rates for *Bases* $= \{7, 9\}$. It is expected that at some point as *Bases* increases, the error rates for all the feature extraction methods should increase as discussed in section 2.2.5.3. But why does DP show this phenomenon before the other algorithms? Since this is only observed for this dataset, and it is only in this dataset that the classes have very different and highly elliptical covariance matrices, the answer likely is related to these two properties, but a definitive answer is not available from these experiments alone.

5.4.1.5 Summary

The results from this experiment show the clear advantage of using feature extraction algorithms when using Fisher's LDA classifier in the neo-classical setting where dimensionality of the input feature is larger than or on the same order of magnitude as the number of samples that are available to train the classifier. It is also clear that Buckheit and Donoho's WDP algorithm performs poorly in this regime since they attempt to invert the within-class covariance matrix which is guaranteed to be singular when $N < f_s$, where f_s gives the dimensionality of the input feature vector.

A phenomenon that occured consistently for all of the sub-experiments in this dataset is that the LDB algorithm outperformed the DDPPSRE algorithm even though both algorithms maximize the same criterion function[2]. So for this dataset, it appears that the best basis algorithm outperforms the dictionary projection pursuit (DPP) algorithm. This is not surprising since the waveforms that make up this dataset all have the same scale, and thus most of the discriminant information for this problem is contained in a single wavelet packet subspace which the best basis algorithm has no problem finding. The DPP algorithm on the other hand seeks to find basis functions one at a time and is thus likely to be more susceptible to noise. The discrepancy between these two feature extraction algorithms is not observed in the other synthetic feature sets.

Considering the KLT algorithms only, it is reassuring to see that the approximate versions of this algorithm KLTBB and KLTDPP perform as well or almost as well as the KLT algorithm in all cases. This encourages the use of these algorithms for problems where the dimensionality of the feature vector is much higher, such as multiframe acoustic features, where the KLT algorithm is too computationally intensive to use.

From the error rates of this experiment alone, there does not appear to be clear advantage of choosing one feature extraction algorithm over another, except for the fact that STD and WDP are clearly poor performers, and DDPPSRE shows slightly poorer performance than the other algorithms.

[2]Recall that both **LDB** and **DDPPSRE** select features using the **SRE** criterion so the only difference between these techniques is that in **LDB**, the best bases algorithm is used to optimize the criterion, and in **DDPPSRE**, the **DPP** algorithm is used to optimize the criterion.

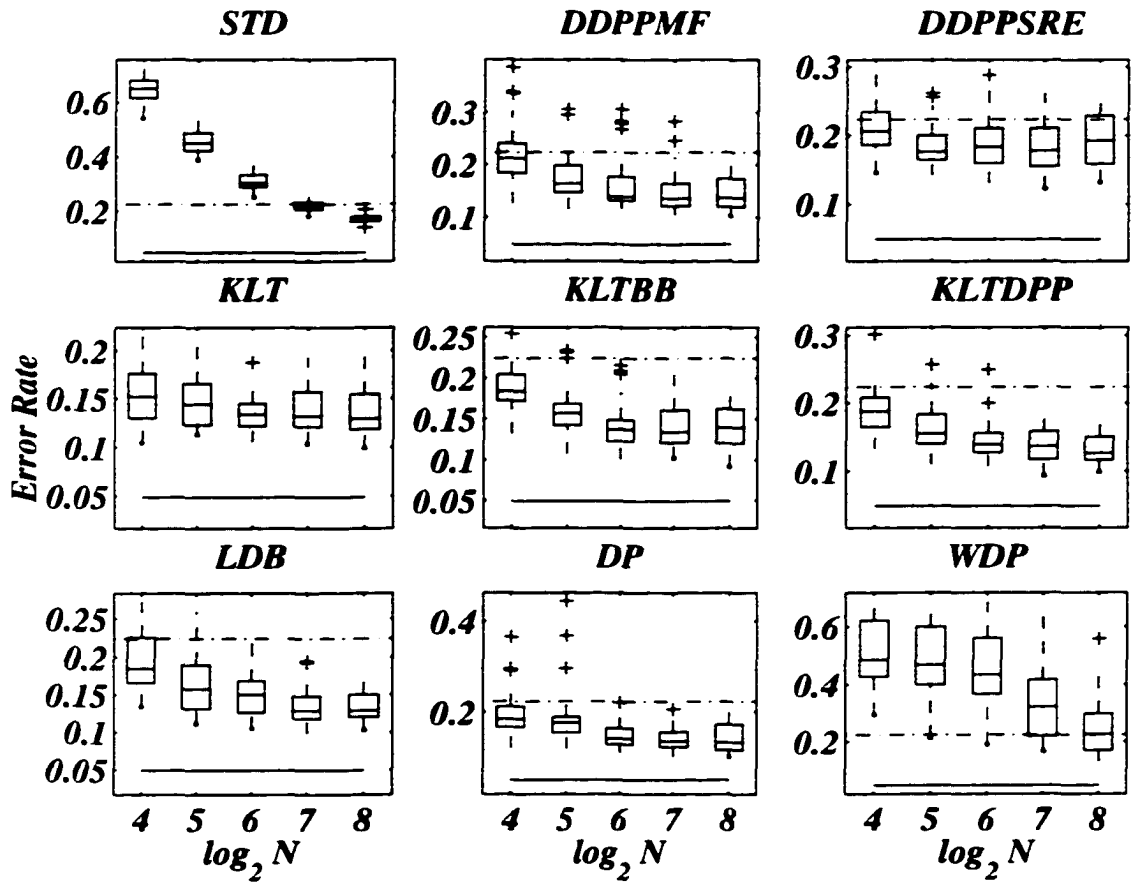

Figure 5.4. *Results for the triangular waveform experiment. The variable N indicates the number of samples from each class that were used to train the classifier. The solid line shows the Bayes error rate and the dashed line shows upper y limit from the graph with the smallest upper y limit.*

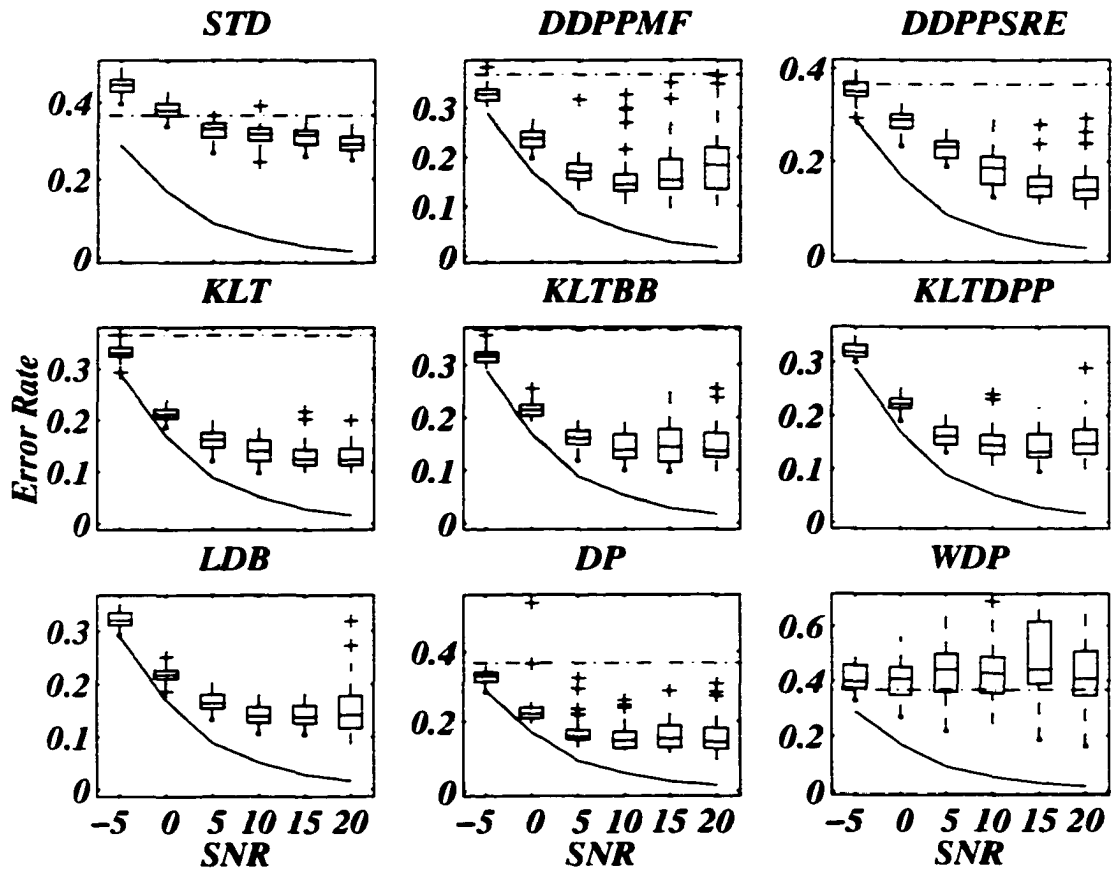

Figure 5.5. *Results for the triangular waveform experiment. The variable SNR indicates the signal to noise ratio of the synthetic waveforms, as described in section 5.2.3. The solid line shows the Bayes error rate and the dashed line shows upper y limit from the graph with the smallest upper y limit.*

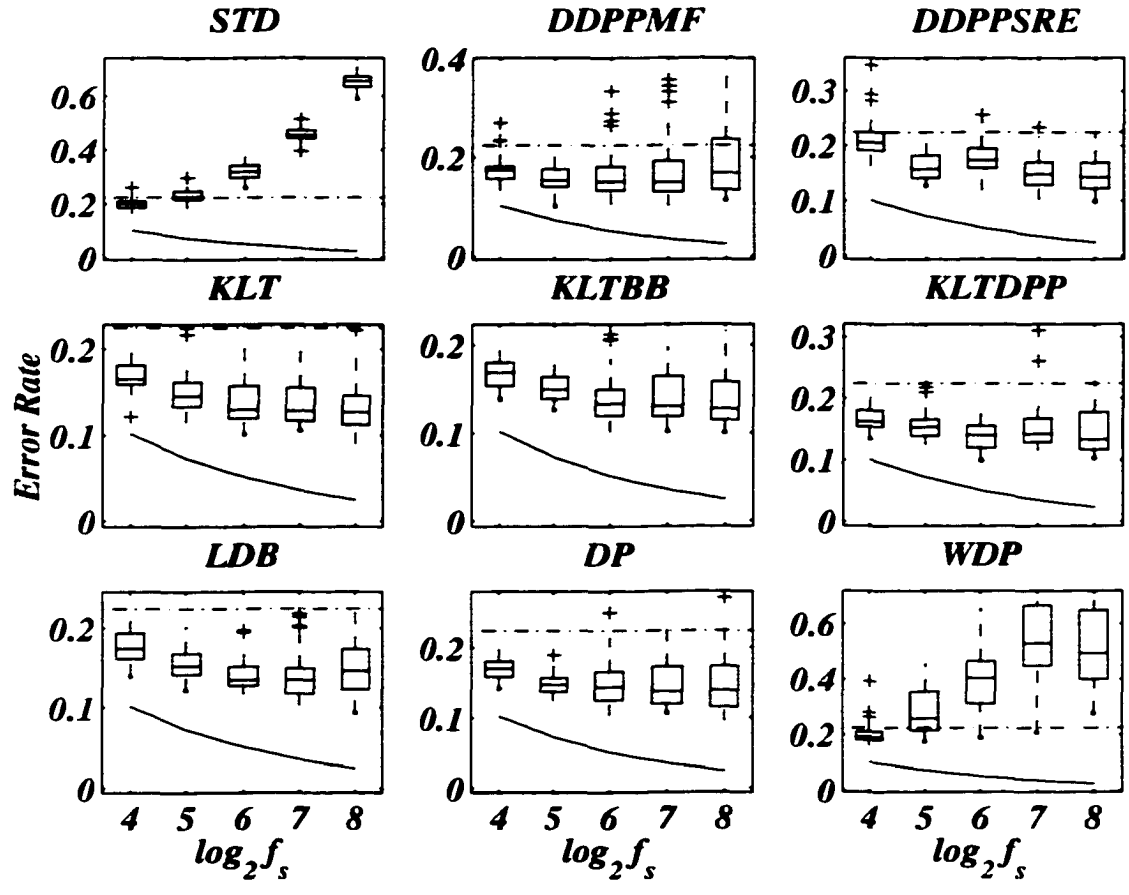

Figure 5.6. *Results for the triangular waveform experiment. The variable f_s indicates the sampling frequency of the synthetic waveforms, as described in section 5.2.1. The solid line shows the Bayes error rate and the dashed line shows upper y limit from the graph with the smallest upper y limit.*

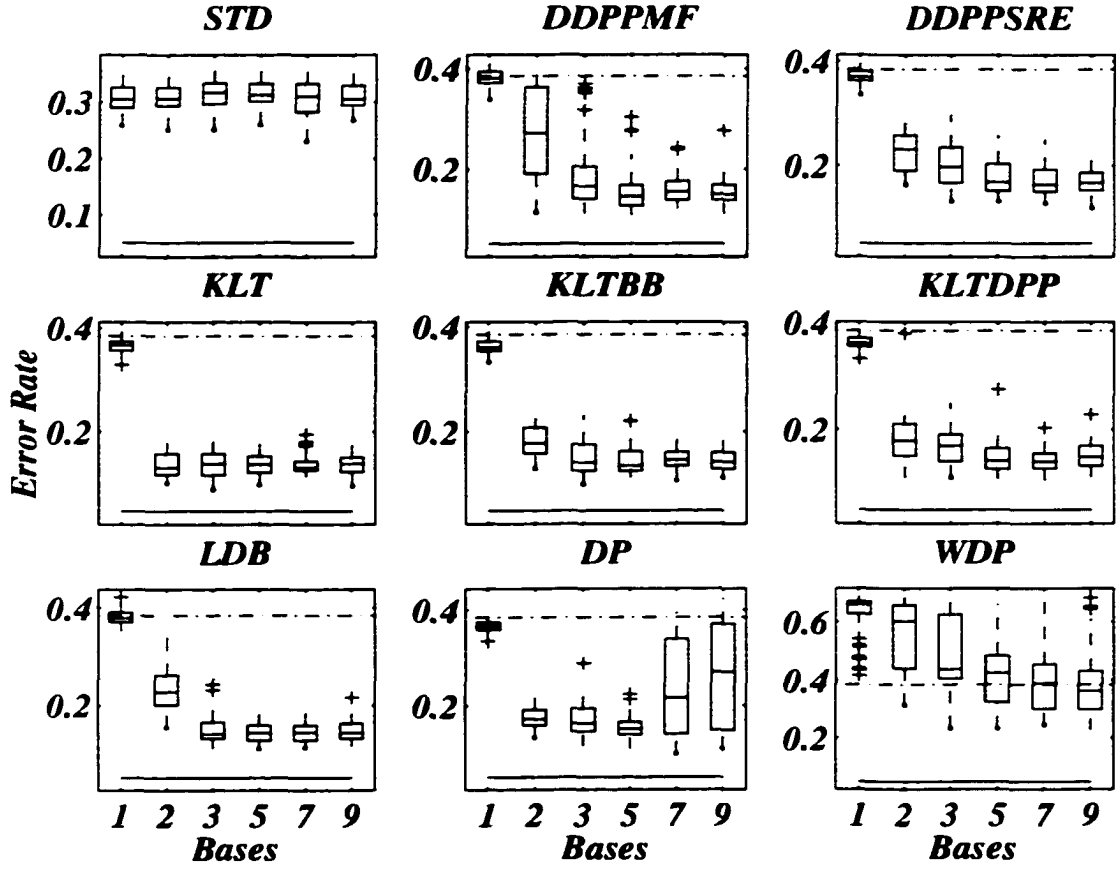

Figure 5.7. *Results for the triangular waveform experiment. The variable Bases indicates the number of basis functions that the feature extraction method kept. The solid line shows the Bayes error rate and the dashed line shows upper y limit from the graph with the smallest upper y limit.*

Algorithm	16	32	64	128	256
	0.678	0.487	0.333	0.232	0.180
STD	**0.650**	**0.450**	**0.303**	**0.219**	**0.170**
	0.616	0.423	0.290	0.207	0.165
	0.240	0.199	0.177	0.164	0.174
DDPPMF	**0.211**	**0.164**	**0.140**	**0.136**	**0.137**
	0.185	0.147	0.132	0.121	0.120
	0.235	0.200	0.210	0.211	0.229
DDPPSRE	**0.207**	**0.176**	**0.183**	**0.178**	**0.192**
	0.186	0.164	0.160	0.155	0.159
	0.176	0.166	0.145	0.157	0.155
KLT	**0.152**	**0.144**	**0.134**	**0.132**	**0.130**
	0.131	0.123	0.122	0.121	0.119
	0.205	0.169	0.148	0.161	0.163
KLTBB	**0.184**	**0.157**	**0.137**	**0.134**	**0.140**
	0.172	0.143	0.122	0.121	0.120
	0.209	0.183	0.156	0.159	0.150
KLTDPP	**0.188**	**0.155**	**0.139**	**0.136**	**0.127**
	0.165	0.140	0.128	0.117	0.117
	0.225	0.190	0.168	0.147	0.150
LDB	**0.185**	**0.157**	**0.150**	**0.129**	**0.129**
	0.166	0.131	0.126	0.118	0.121
	0.212	0.189	0.164	0.156	0.171
DP	**0.185**	**0.177**	**0.141**	**0.135**	**0.132**
	0.166	0.154	0.128	0.123	0.115
	0.623	0.600	0.561	0.419	0.299
WDP	**0.487**	**0.469**	**0.435**	**0.323**	**0.226**
	0.428	0.400	0.368	0.219	0.171
Bayes	**0.049**	**0.049**	**0.049**	**0.049**	**0.049**

Table 5.2. *Results for sub-experiment N in the triangular waveform experiment. The center bold number gives the median error rate, the upper number gives the 75^{th} percentile, and the lower number gives the 25^{th} percentile.*

Algorithm	-5	0	5	10	15	20
	0.459	0.397	0.345	0.332	0.327	0.312
STD	**0.446**	**0.380**	**0.333**	**0.318**	**0.316**	**0.292**
	0.427	0.366	0.309	0.301	0.291	0.277
	0.336	0.252	0.186	0.164	0.194	0.218
DDPPMF	**0.328**	**0.237**	**0.169**	**0.144**	**0.152**	**0.182**
	0.315	0.219	0.154	0.131	0.135	0.135
	0.371	0.303	0.245	0.212	0.169	0.166
DDPPSRE	**0.351**	**0.292**	**0.233**	**0.187**	**0.148**	**0.139**
	0.341	0.273	0.209	0.152	0.126	0.121
	0.341	0.221	0.176	0.162	0.143	0.145
KLT	**0.331**	**0.211**	**0.163**	**0.141**	**0.125**	**0.123**
	0.325	0.204	0.148	0.121	0.114	0.114
	0.324	0.223	0.175	0.167	0.176	0.170
KLTBB	**0.316**	**0.213**	**0.160**	**0.137**	**0.144**	**0.137**
	0.305	0.203	0.149	0.122	0.116	0.125
	0.333	0.232	0.179	0.163	0.166	0.174
KLTDPP	**0.319**	**0.221**	**0.162**	**0.144**	**0.131**	**0.147**
	0.311	0.214	0.146	0.128	0.122	0.128
	0.334	0.226	0.180	0.156	0.158	0.178
LDB	**0.320**	**0.217**	**0.164**	**0.139**	**0.138**	**0.141**
	0.311	0.210	0.154	0.126	0.126	0.116
	0.340	0.238	0.176	0.169	0.187	0.181
DP	**0.330**	**0.221**	**0.155**	**0.146**	**0.150**	**0.141**
	0.313	0.209	0.148	0.124	0.125	0.122
	0.456	0.450	0.498	0.487	0.616	0.507
WDP	**0.398**	**0.409**	**0.442**	**0.427**	**0.442**	**0.407**
	0.373	0.349	0.359	0.357	0.390	0.347
Bayes	**0.289**	**0.170**	**0.088**	**0.051**	**0.028**	**0.016**

Table 5.3. *Results for sub-experiment SNR in the triangular waveform experiment. The center bold number gives the median error rate, the upper number gives the* 75^{th} *percentile, and the lower number gives the* 25^{th} *percentile.*

Algorithm	16	32	64	128	256
	0.210	0.246	0.342	0.470	0.671
STD	**0.199**	**0.226**	**0.318**	**0.453**	**0.655**
	0.185	0.216	0.299	0.444	0.633
	0.180	0.176	0.181	0.192	0.236
DDPPMF	**0.174**	**0.153**	**0.151**	**0.149**	**0.168**
	0.157	0.140	0.133	0.130	0.135
	0.225	0.182	0.195	0.170	0.169
DDPPSRE	**0.207**	**0.158**	**0.174**	**0.149**	**0.143**
	0.192	0.142	0.160	0.129	0.123
	0.181	0.161	0.156	0.154	0.145
KLT	**0.165**	**0.145**	**0.130**	**0.129**	**0.127**
	0.159	0.134	0.120	0.117	0.112
	0.180	0.164	0.150	0.164	0.158
KLTBB	**0.169**	**0.149**	**0.134**	**0.131**	**0.128**
	0.154	0.139	0.120	0.120	0.116
	0.182	0.166	0.156	0.169	0.179
KLTDPP	**0.164**	**0.154**	**0.141**	**0.143**	**0.134**
	0.155	0.140	0.120	0.130	0.118
	0.194	0.168	0.152	0.150	0.174
LDB	**0.174**	**0.152**	**0.134**	**0.135**	**0.146**
	0.162	0.142	0.128	0.118	0.123
	0.180	0.156	0.165	0.172	0.173
DP	**0.170**	**0.146**	**0.143**	**0.137**	**0.139**
	0.158	0.137	0.125	0.119	0.115
	0.211	0.355	0.465	0.667	0.649
WDP	**0.191**	**0.257**	**0.404**	**0.531**	**0.495**
	0.181	0.214	0.311	0.450	0.401
Bayes	**0.102**	**0.073**	**0.051**	**0.035**	**0.024**

Table 5.4. *Results for sub-experiment f_3 in the triangular waveform experiment. The center bold number gives the median error rate, the upper number gives the 75^{th} percentile, and the lower number gives the 25^{th} percentile.*

Algorithm	1	2	3	5	7	9
	0.326	0.326	0.335	0.333	0.333	0.330
STD	**0.305**	**0.307**	**0.318**	**0.313**	**0.311**	**0.305**
	0.291	0.294	0.297	0.302	0.283	0.295
	0.395	0.363	0.204	0.168	0.176	0.168
DDPPMF	**0.380**	**0.272**	**0.165**	**0.146**	**0.154**	**0.149**
	0.371	0.190	0.141	0.126	0.138	0.138
	0.381	0.258	0.235	0.204	0.192	0.187
DDPPSRE	**0.370**	**0.231**	**0.197**	**0.169**	**0.164**	**0.167**
	0.362	0.190	0.168	0.152	0.151	0.152
	0.373	0.158	0.159	0.154	0.144	0.152
KLT	**0.367**	**0.132**	**0.140**	**0.138**	**0.133**	**0.140**
	0.357	0.120	0.118	0.123	0.128	0.124
	0.372	0.207	0.175	0.162	0.160	0.160
KLTBB	**0.360**	**0.177**	**0.141**	**0.134**	**0.145**	**0.142**
	0.354	0.158	0.124	0.125	0.134	0.128
	0.371	0.211	0.191	0.165	0.156	0.169
KLTDPP	**0.361**	**0.179**	**0.169**	**0.141**	**0.140**	**0.148**
	0.356	0.151	0.140	0.126	0.127	0.132
	0.391	0.262	0.164	0.158	0.158	0.162
LDB	**0.380**	**0.228**	**0.140**	**0.143**	**0.144**	**0.143**
	0.371	0.201	0.132	0.127	0.128	0.131
	0.373	0.191	0.193	0.166	0.339	0.370
DP	**0.366**	**0.172**	**0.163**	**0.150**	**0.216**	**0.270**
	0.358	0.158	0.146	0.139	0.140	0.148
	0.671	0.658	0.626	0.485	0.457	0.431
WDP	**0.660**	**0.602**	**0.438**	**0.427**	**0.390**	**0.362**
	0.627	0.441	0.405	0.324	0.301	0.299
Bayes	**0.050**	**0.050**	**0.050**	**0.050**	**0.050**	**0.050**

Table 5.5. *Results for sub-experiment Bases in the triangular waveform experiment. The center bold number gives the median error rate, the upper number gives the* 75^{th} *percentile, and the lower number gives the* 25^{th} *percentile.*

5.5 Common Variance Waveforms

This synthetic dataset was created to show the well known weakness of the KLT method as a form of feature extraction. Since KLT looks at the variance of the whole dataset, it has no way of knowing whether the variance is due to scatter which is common to all classes, or whether the variance is due to differences between classes. It simply chooses the basis functions with the largest variance, as described in section 4.7.4. To show this effect, a two class dataset was created where there are five waveforms that have identical mean values and large common variances for both classes. In a sixth waveform, the two classes have different means and identical variances. Therefore, while there is large variance in six waveforms, only one basis function contains any discriminatory power. The dataset was created according to the signal model described in section 5.2 as follows.

There are six columns in the $\boldsymbol{A}$ matrix that are defined as

$$a_k(nT) = \kappa \max\left(1 - s|nT - t_k|, 0\right), \tag{5.13}$$

for $k = \{1, 2, 3, 4, 5, 6\}$, where $T = 1/f_s$ is the sampling interval, $s = 1/2\Delta T$, $t_k = 2k\Delta T$, with $\Delta T = 1/14$, and κ is a constant chosen so that $\|a_k\| = 1$. These waveforms are plotted in figure 5.8.

The Normal density parameters of $\boldsymbol{\alpha}$ for each class are given as,

$$\boldsymbol{\mu}_{\boldsymbol{\alpha}}^{(1)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T \qquad \boldsymbol{\mu}_{\boldsymbol{\alpha}}^{(2)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & -1 \end{bmatrix}^T \tag{5.14a}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{(1)} = \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{(2)} = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.14b}$$

Finally, the noise added in this model is white and Normally distributed $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$. Typical waveforms from each of the classes are plotted in figure 5.9.

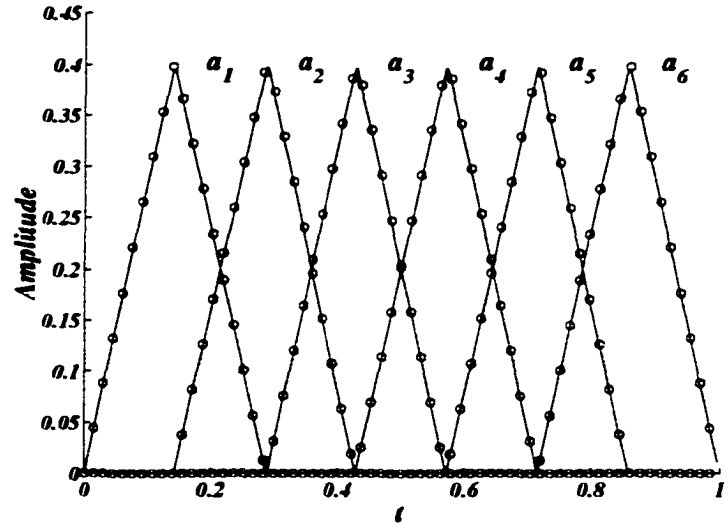

Figure 5.8. *Common variance waveforms for the* $\mathbf{A}$ *matrix. The solid line shows the continuous version of the waveform, and the circles represent the sampled version with* $f_s = 64$.

5.5.1 Experimental Results

The experimental results of applying each feature extraction algorithm (section 4.7) to the common variance waveform dataset for each sub-experiment (section 5.3) are plotted at the end of this section in figures 5.10–5.13. A tabular presentation of the results for each sub-experiment are given in tables 5.6–5.9. The results are discussed in the following sections.

5.5.1.1 sub-experiment N

The Bayes error rate for this sub-experiment is 0.167 for all values of N.

The lowest error rates for this sub-experiment are achieved by the DDPPMF and DP algorithms which essentially become Bayes classifiers for $N \geq 64$. The LDB and DDPPSRE algorithms show similar error rates for all values of N, and are only slightly higher than DDPPMF and DP. STD and WDP perform poorly for low values of N with error rates of ~ 0.5, but gradually improve as N increases reaching an error rate ~ 0.2 for $N = 256$. The reasons for this poor performance are the same as those presented for the triangular waveform experiment.

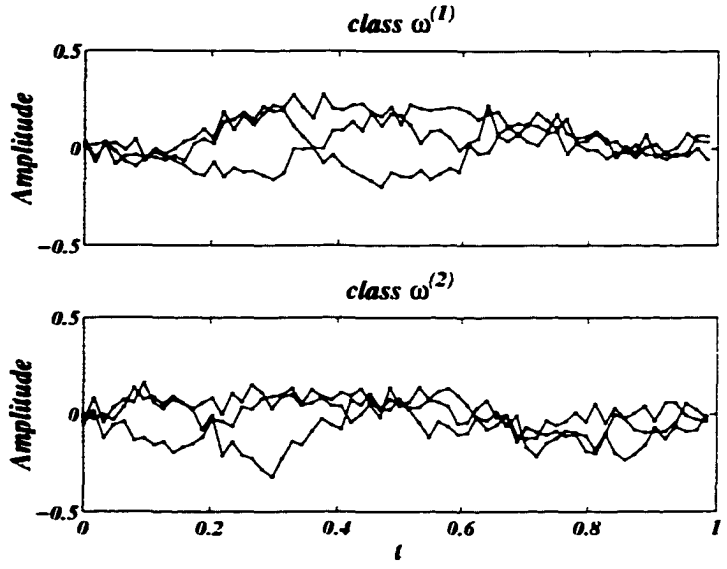

Figure 5.9. *Typical waveforms from the common variance waveform classes with* $f_s = 64$ *and* $SNR = 10$.

The KLT algorithm and its approximate forms KLTBB and KLTDPP perform poorly for all values of N with error rates of ~ 0.4. Admittedly, this experiment was designed to show this phenomenon. Since this data set is made up of six waveforms, the first five showing large variance but no differences between the classes and the sixth waveform showing small variance and a small difference between the classes, the KLT algorithms choose basis functions that are correlated with the first five waveforms and miss all of the discriminant power in the dataset. The KLTDPP algorithm shows a reasonable median error rate of ~ 0.2 for $N \geq 64$, but shows a large dispersion with many trial error rates occuring around ~ 0.4. This effect is attributed to random chance.

5.5.1.2 sub-experiment *SNR*

The Bayes error rate for this sub-experiment starts at 0.267 for $SNR = -5$, decreases quickly at first, and then starts to level off, achieving a final error rate of 0.16 for $SNR = 20$.

The same general comments that were made for sub-experiment N apply here. To summarize, DDPPMF and DP are essentially Bayes classifiers for all values of SNR,

the DDPPSRE and LDB algorithm have slightly higher error rates, STD and WDP perform poorly, and the KLT algorithms fail miserably.

The phenomenon observed for the triangular waveform dataset where the error rates become closer to the Bayes error rate as the *SNR* decreases is not observed here. This is because all the classes in the dataset have the same covariance matrix and thus the assumptions of Fisher's LDA are not violated.

5.5.1.3 sub-experiment f_s

The Bayes error rate for this sub-experiment starts at 0.183 for $f_s = 16$, and then gradually decreases achieving a final error rate of 0.16 for $f_s = 256$.

The error rates for the DDPPMF and DP algorithms are only slightly larger than the Bayes error rate for all values of f_s, while the DDPPSRE and LDB algorithms have slightly larger error rates. The KLT algorithms perform poorly as expected from the discussion in sub-experiment *N*. In the same manner as the triangular dataset, the STD and WDP algorithm show increasingly higher error rates as f_s increases for the same reasons discussed in section 5.4.1.3.

5.5.1.4 sub-experiment *Bases*

The Bayes error rate for this sub-experiment is 0.167 for all values of *Bases*.

The DDPPMF algorithm is clearly the superior algorithm for small *Bases* values achieving an error rate only slightly larger than the Bayes rate. The next closest contender is the DP algorithm but for *Bases* = 1 the ratio of the median error rate for the DP algorithm and the DDPPMF algorithm is 3× the interquartile distance of the DDPPMF algorithm. The next best algorithms are the DDPPSRE and LDB techniques which show similar error rates between each other but seem to require *Bases* ≥ 5 for the error rates to be even comparable to the DDPPMF and DP error rates. As expected, the WDP algorithms perform poorly for all values of *Bases*, and since the STD algorithm is not modified as *Bases* changes, it obtains consistent error rates of ~ 0.28.

The KLT algorithms show a sudden decrease in the error rate when *Bases* > 5. This is to be expected since only five waveforms were set up to have high variance with no discriminant power. After that, these algorithms can easily find the basis

that does contain the discriminant information.

5.5.1.5 summary

This experiment was designed to show the weakness of the KLT algorithms when there are high variance subspaces in the dataset without any discriminant power. Since the KLT algorithms only look for high variance when choosing basis functions, they are fooled by these subspaces. This point was made abundantly clear in this experiment.

Several other important observations can also be made from this experiment. All the disciminant information for this dataset exists on a line, so only one basis function is needed to extract this information. The only algorithm that was able to do this was the DDPPMF algorithm which was able to achieve near Bayes error rate of 0.18 when keeping only one basis function. The DP algorithm did a reasonable job as well when keeping only one basis function obtaining an error rate of ~ 0.25, but all the other algorithms gave median error rates greater than ~ 0.4.

Since both DDPPSRE and LDB obtain essentially the same error rates in all experiments, it appears that the best basis algorithm and dictionary projection pursuit algorithm behave similarly for this particular dataset. The fact that DDPPMF outperforms DDPPSRE in all cases suggest that the modified Fisher (MF) criterion is a better projection criterion than the symmetric relative entropy (SRE) criterion for this particular problem. Herein lies the main advantage of the discriminant dictionary projection pursuit (DDPP) algorithm over the local discriminant bases (LDB) algorithm, since the LDB algorithm cannot be used with the MF criterion.

It was also obvious in this experiment that using no feature extraction (*i.e.*, the STD method) or using the WDP algorithm is a poor choice for the same reasons discussed in section 5.4.1.5.

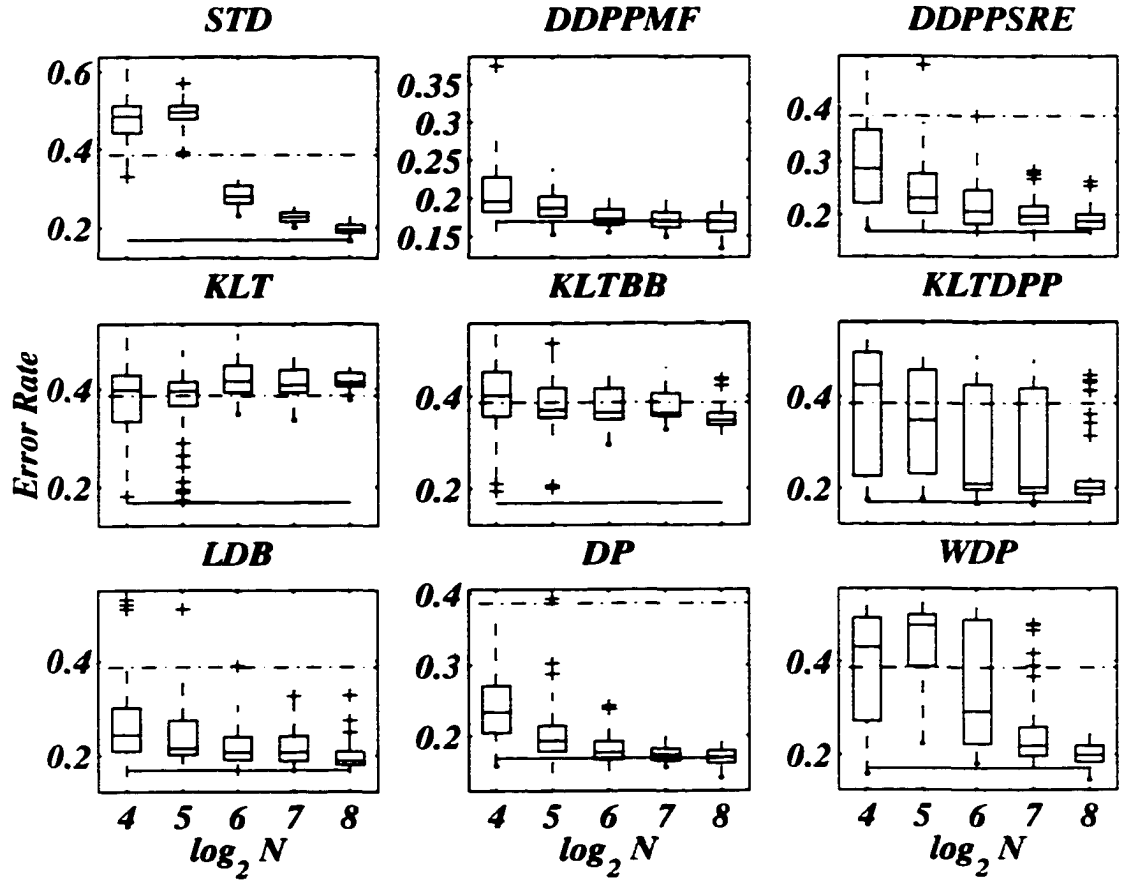

Figure 5.10. *Results for the common variance waveform experiment. The variable N indicates the number of samples from each class that were used to train the classifier. The solid line shows the Bayes error rate and the dashed line shows upper y limit from the graph with the smallest upper y limit.*

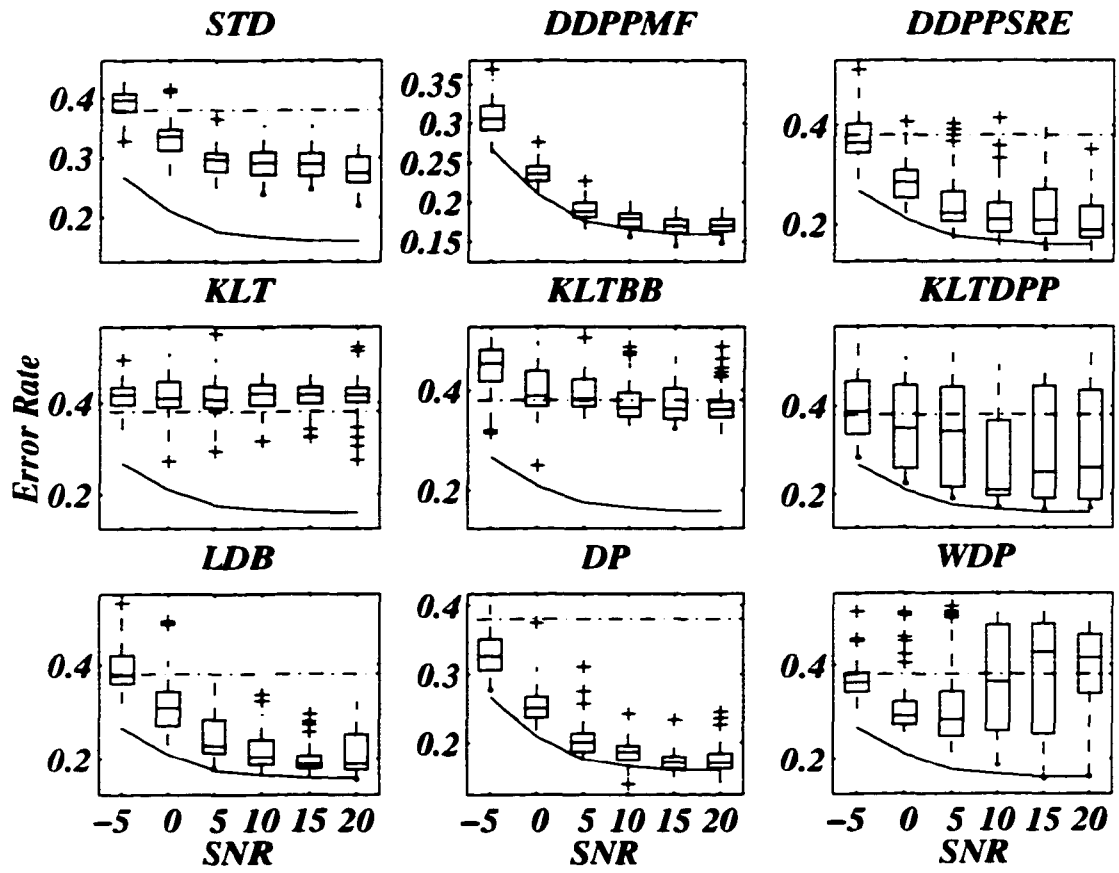

Figure 5.11. *Results for the common variance waveform experiment. The variable SNR indicates the signal to noise ratio of the synthetic waveforms, as described in section 5.2.3. The solid line shows the Bayes error rate and the dashed line shows upper y limit from the graph with the smallest upper y limit.*

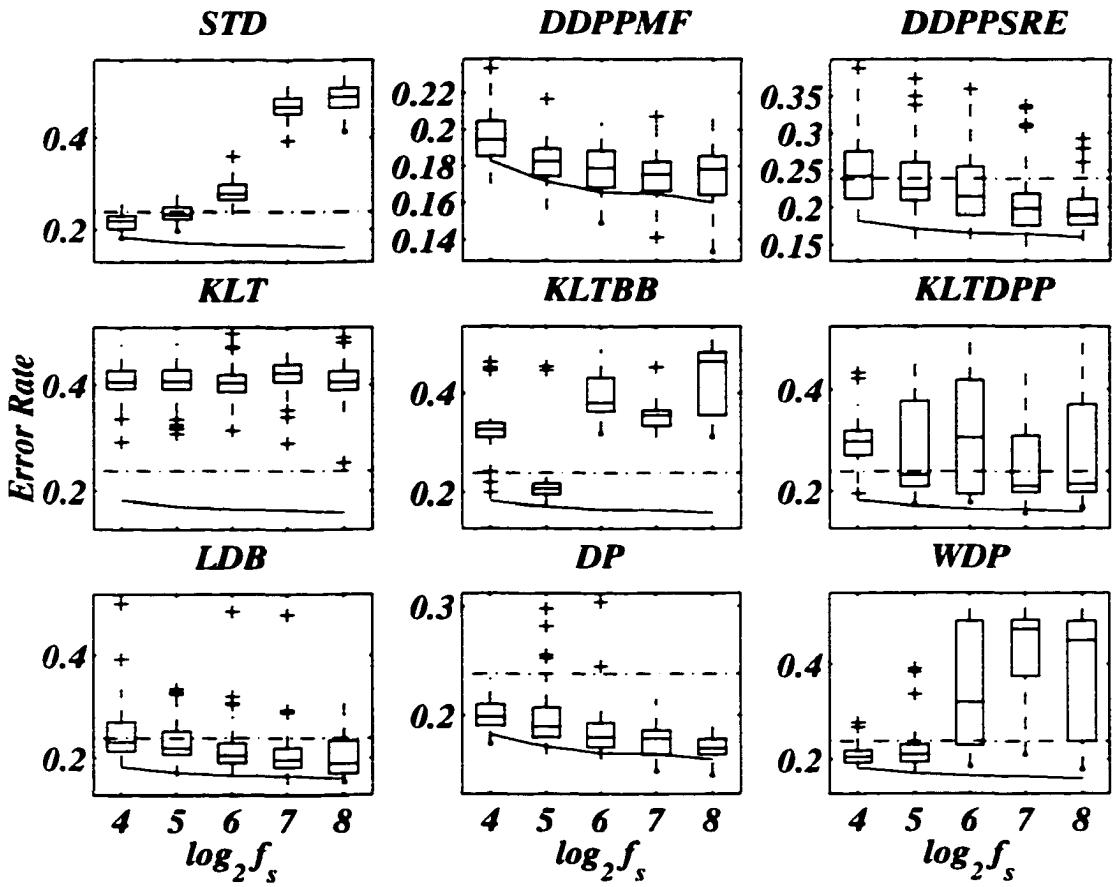

Figure 5.12. *Results for the common variance waveform experiment. The variable f_s indicates the sampling frequency of the synthetic waveforms, as described in section 5.2.1. The solid line shows the Bayes error rate and the dashed line shows upper y limit from the graph with the smallest upper y limit.*

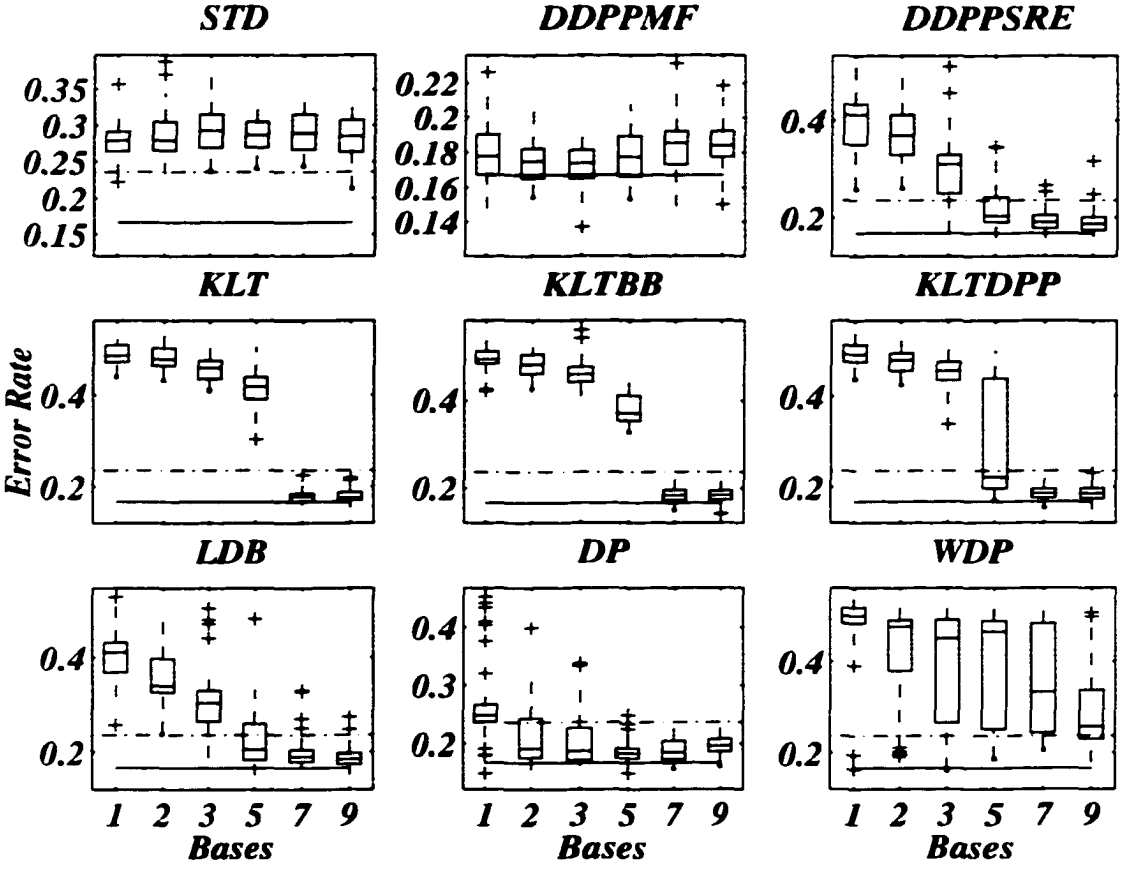

Figure 5.13. *Results for the common variance waveform experiment. The variable Bases indicates the number of basis functions that the feature extraction method kept. The solid line shows the Bayes error rate and the dashed line shows upper y limit from the graph with the smallest upper y limit.*

Algorithm	16	32	64	128	256
	0.512	0.512	0.307	0.239	0.207
STD	**0.484**	**0.496**	**0.280**	**0.227**	**0.194**
	0.441	0.479	0.263	0.216	0.187
	0.228	0.202	0.185	0.181	0.180
DDPPMF	**0.196**	**0.187**	**0.172**	**0.170**	**0.169**
	0.183	0.176	0.165	0.161	0.156
	0.360	0.277	0.247	0.217	0.202
DDPPSRE	**0.287**	**0.232**	**0.207**	**0.198**	**0.189**
	0.222	0.204	0.184	0.184	0.175
	0.429	0.415	0.448	0.438	0.432
KLT	**0.398**	**0.396**	**0.415**	**0.407**	**0.413**
	0.334	0.366	0.393	0.393	0.406
	0.453	0.417	0.416	0.405	0.363
KLTBB	**0.401**	**0.371**	**0.364**	**0.362**	**0.346**
	0.357	0.354	0.350	0.356	0.337
	0.497	0.458	0.427	0.418	0.214
KLTDPP	**0.426**	**0.348**	**0.208**	**0.201**	**0.199**
	0.226	0.231	0.197	0.189	0.186
	0.301	0.274	0.240	0.241	0.208
LDB	**0.243**	**0.215**	**0.206**	**0.207**	**0.189**
	0.210	0.201	0.190	0.189	0.181
	0.271	0.215	0.192	0.182	0.179
DP	**0.234**	**0.194**	**0.176**	**0.173**	**0.170**
	0.205	0.178	0.167	0.165	0.162
	0.493	0.500	0.489	0.259	0.218
WDP	**0.429**	**0.476**	**0.293**	**0.219**	**0.198**
	0.272	0.389	0.221	0.197	0.183
Bayes	**0.169**	**0.169**	**0.169**	**0.169**	**0.169**

Table 5.6. *Results for sub-experiment N in the common variance waveform experiment. The center bold number gives the median error rate, the upper number gives the* 75^{th} *percentile, and the lower number gives the* 25^{th} *percentile.*

Algorithm	-5	0	5	10	15	20
	0.408	0.348	0.308	0.310	0.307	0.302
STD	**0.397**	**0.336**	**0.297**	**0.292**	**0.290**	**0.274**
	0.378	0.313	0.277	0.272	0.270	0.259
	0.323	0.245	0.199	0.185	0.178	0.178
DDPPMF	**0.306**	**0.236**	**0.188**	**0.179**	**0.170**	**0.170**
	0.293	0.227	0.181	0.168	0.161	0.163
	0.403	0.308	0.266	0.242	0.271	0.236
DDPPSRE	**0.365**	**0.284**	**0.222**	**0.209**	**0.208**	**0.188**
	0.345	0.253	0.205	0.184	0.180	0.173
	0.434	0.447	0.436	0.439	0.434	0.433
KLT	**0.417**	**0.409**	**0.407**	**0.420**	**0.417**	**0.417**
	0.396	0.391	0.387	0.395	0.398	0.401
	0.481	0.439	0.422	0.395	0.403	0.378
KLTBB	**0.453**	**0.389**	**0.383**	**0.365**	**0.363**	**0.361**
	0.419	0.369	0.367	0.348	0.343	0.346
	0.456	0.446	0.441	0.366	0.445	0.435
KLTDPP	**0.386**	**0.348**	**0.342**	**0.209**	**0.249**	**0.260**
	0.337	0.259	0.215	0.196	0.190	0.188
	0.420	0.344	0.285	0.240	0.206	0.253
LDB	**0.378**	**0.309**	**0.229**	**0.205**	**0.189**	**0.189**
	0.360	0.272	0.213	0.188	0.180	0.178
	0.351	0.268	0.213	0.195	0.179	0.184
DP	**0.327**	**0.251**	**0.201**	**0.187**	**0.171**	**0.170**
	0.307	0.237	0.186	0.174	0.162	0.162
	0.384	0.323	0.344	0.485	0.487	0.465
WDP	**0.364**	**0.291**	**0.284**	**0.365**	**0.427**	**0.415**
	0.343	0.274	0.247	0.260	0.254	0.340
Bayes	**0.267**	**0.210**	**0.176**	**0.166**	**0.160**	**0.160**

Table 5.7. *Results for sub-experiment SNR in the common variance waveform experiment. The center bold number gives the median error rate, the upper number gives the* 75^{th} *percentile, and the lower number gives the* 25^{th} *percentile.*

Algorithm	16	32	64	128	256
	0.229	0.247	0.297	0.487	0.509
STD	**0.219**	**0.232**	**0.277**	**0.467**	**0.490**
	0.201	0.222	0.265	0.452	0.466
	0.204	0.189	0.188	0.182	0.185
DDPPMF	**0.194**	**0.183**	**0.179**	**0.175**	**0.178**
	0.185	0.174	0.168	0.166	0.164
	0.276	0.261	0.255	0.218	0.210
DDPPSRE	**0.242**	**0.225**	**0.214**	**0.198**	**0.189**
	0.211	0.209	0.189	0.176	0.177
	0.428	0.429	0.420	0.440	0.427
KLT	**0.407**	**0.408**	**0.404**	**0.423**	**0.407**
	0.393	0.392	0.388	0.407	0.391
	0.339	0.217	0.432	0.365	0.483
KLTBB	**0.327**	**0.207**	**0.381**	**0.355**	**0.464**
	0.312	0.196	0.363	0.334	0.355
	0.320	0.377	0.419	0.310	0.371
KLTDPP	**0.298**	**0.233**	**0.306**	**0.211**	**0.214**
	0.271	0.209	0.194	0.198	0.198
	0.270	0.253	0.229	0.219	0.233
LDB	**0.232**	**0.219**	**0.205**	**0.196**	**0.189**
	0.213	0.207	0.190	0.181	0.169
	0.210	0.207	0.193	0.186	0.178
DP	**0.199**	**0.190**	**0.180**	**0.179**	**0.170**
	0.191	0.180	0.171	0.164	0.164
	0.218	0.231	0.490	0.493	0.490
WDP	**0.205**	**0.211**	**0.320**	**0.473**	**0.449**
	0.193	0.195	0.230	0.375	0.238
Bayes	**0.183**	**0.172**	**0.166**	**0.165**	**0.160**

Table 5.8. *Results for sub-experiment f_s in the common variance waveform experiment. The center bold number gives the median error rate, the upper number gives the 75^{th} percentile, and the lower number gives the 25^{th} percentile.*

Algorithm	1	2	3	5	7	9
	0.292	0.306	0.316	0.306	0.316	0.308
STD	**0.280**	**0.280**	**0.293**	**0.287**	**0.290**	**0.286**
	0.265	0.266	0.270	0.270	0.268	0.264
	0.190	0.182	0.181	0.189	0.192	0.192
DDPPMF	**0.178**	**0.175**	**0.174**	**0.177**	**0.185**	**0.184**
	0.167	0.165	0.165	0.166	0.173	0.177
	0.434	0.412	0.330	0.242	0.205	0.201
DDPPSRE	**0.411**	**0.370**	**0.310**	**0.204**	**0.191**	**0.187**
	0.351	0.330	0.249	0.190	0.178	0.174
	0.509	0.502	0.475	0.439	0.185	0.187
KLT	**0.485**	**0.477**	**0.459**	**0.418**	**0.178**	**0.176**
	0.472	0.463	0.434	0.392	0.169	0.171
	0.511	0.503	0.475	0.411	0.195	0.193
KLTBB	**0.492**	**0.481**	**0.457**	**0.371**	**0.183**	**0.183**
	0.483	0.460	0.441	0.354	0.172	0.173
	0.513	0.494	0.475	0.439	0.195	0.197
KLTDPP	**0.492**	**0.479**	**0.456**	**0.222**	**0.186**	**0.186**
	0.475	0.456	0.436	0.196	0.176	0.175
	0.432	0.398	0.331	0.260	0.204	0.197
LDB	**0.411**	**0.340**	**0.304**	**0.205**	**0.191**	**0.185**
	0.369	0.327	0.265	0.184	0.180	0.174
	0.267	0.242	0.226	0.190	0.203	0.206
DP	**0.248**	**0.190**	**0.186**	**0.182**	**0.184**	**0.195**
	0.237	0.175	0.171	0.174	0.172	0.185
	0.517	0.490	0.492	0.489	0.483	0.337
WDP	**0.500**	**0.476**	**0.450**	**0.465**	**0.332**	**0.257**
	0.483	0.379	0.266	0.253	0.244	0.230
Bayes	**0.167**	**0.167**	**0.167**	**0.167**	**0.167**	**0.167**

Table 5.9. *Results for sub-experiment Bases in the common variance waveform experiment. The center bold number gives the median error rate, the upper number gives the* 75^{th} *percentile, and the lower number gives the* 25^{th} *percentile.*

5.6 Multiscale Waveforms

This synthetic dataset was created to simulate the case when discriminatory information exists on multiple scales. It is conjectured that the LDB and KLTBB algorithms will perform poorly for this type of data. Any algorithm that uses the best basis algorithm (see section 3.3.4.1) is limited to choosing basis functions from subspaces of the wavelet packet representation which are orthogonal to one another. Sometimes, good features for discrimination may exist in two subspaces that are not orthogonal to one another and the algorithm must choose one subspace over the other, thus missing some potentially good features. The algorithm will have to represent the features from the subspace that was not included as linear combinations of basis functions in the subspaces that it kept. Ultimately, this means that to achieve the same classification error rates, best basis algorithms must keep more basis functions in comparison to algorithms that use dictionary projection pursuit (DDPPSRE, DDPPMF and KLT-DPP) or discriminant pursuit (DP and WDP) since these algorithms do not suffer from this limitation.

A two class dataset was created with eight waveforms located in the left half of the interval having small scale, and four waveforms spread evenly on the interval having large scale. This situation is typical of how information is distributed in acoustic spectra where a lot of information is contained at low frequencies with narrow bandwidth, and less information is contained at higher frequencies with large bandwidth. The covariance matrix for both classes is identical, but the mean is different for each waveform by the same amount with an alternating pattern. Therefore, each waveform contains the same amount of discriminant information.

The dataset was created according to the signal model described in section 5.2 as follows. The columns of the $\boldsymbol{A}$ matrix are given by the Coiflet order 2 wavelet packet basis functions with indices shown in table 5.10, where $m0 = \log_2(f_s)$ (recall that f_s also gives the length of the signal). These waveforms are plotted in figure 5.14. Notice the slight shift in the waveforms for different sampling rates. This is a consequence of how the wavelets are constructed and no attempt was made to align the waveforms for different sampling rates.

$\boldsymbol{A}$ column	s	f	p
1	$m0-5$	0	0
2	$m0-5$	0	2
3	$m0-5$	0	4
4	$m0-5$	0	6
5	$m0-5$	0	8
6	$m0-5$	0	10
7	$m0-5$	0	12
8	$m0-5$	0	14
9	$m0-3$	0	0
10	$m0-3$	0	1
11	$m0-3$	0	2
12	$m0-3$	0	3

Table 5.10. *Wavelet packet indices for multiscale waveforms where* $m0 = \log_2(f_s)$

The Normal density parameters of $\boldsymbol{\alpha}$ for each class are given as,

$$\boldsymbol{\mu}_{\boldsymbol{\alpha}}^{(1)} = \begin{bmatrix} 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 \end{bmatrix}^T \tag{5.15a}$$

$$\boldsymbol{\mu}_{\boldsymbol{\alpha}}^{(2)} = \begin{bmatrix} 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 \end{bmatrix}^T \tag{5.15b}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{(1)} = \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{(2)} = \boldsymbol{I}_{12}, \tag{5.15c}$$

where $\boldsymbol{I}_{12}$ is a 12×12 identity matrix. Finally, the noise added in this model is white and Normally distributed $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$. Typical waveforms from each of the classes are plotted in figure 5.15.

5.6.1 Experimental Results

The experimental results of applying each feature extraction algorithm (section 4.7) to the multiscale waveform dataset for each sub-experiment (section 5.3) are plotted at the end of this section in figures 5.16–5.19. A tabular presentation of the results for each sub-experiment are given in tables 5.11–5.14. The results are discussed in the following sections.

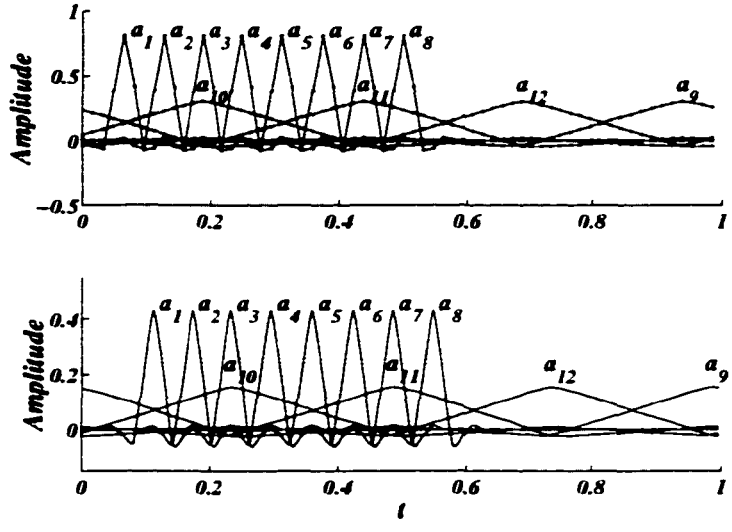

Figure 5.14. *Multiscale waveforms for the* $\mathbf{A}$ *matrix. The top graph shows the waveforms for* $f_s = 64$*, with solid dots at the sampled points. The bottom graph shows the waveforms for* $f_s = 256$*, but solid dots at the sampled points are not shown. Notice the slight shift in the waveforms.*

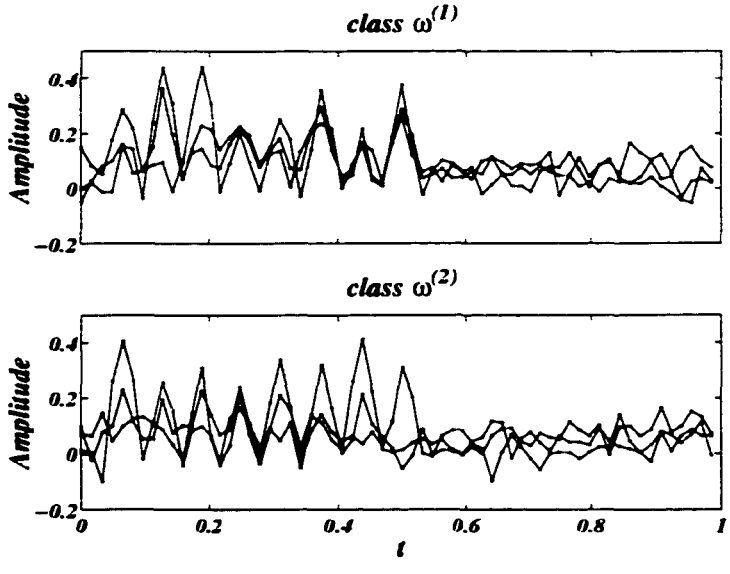

Figure 5.15. *Typical waveforms from the multiscale waveform classes with* $f_s = 64$ *and* $SNR = 10$.

5.6.1.1 sub-experiment N

The Bayes error rate for this sub-experiment is 0.048 for all values of N.

The algorithms DDPPMF, KLT, KLTBB, KLTDPP, and DP all obtain similar and very good error rates of $\sim$ 0.09 for $N = 16$ and decrease to $\sim$ 0.06 for $N = 256$. The KLT algorithm may be slightly superior and the KLTBB algorithm may be slightly inferior to the other algorithms for $N = 16$ achieving an error rate of $\sim$ 0.07 and $\sim$ 0.11 respectively. Following close behind, but clearly inferior are the DDPPSRE and LDB algorithms which obtain consistently higher error rates than the aforementioned algorithms. The STD and WDP algorithms perform very poorly for $N < 128$, but show reasonable result for N larger than this.

5.6.1.2 sub-experiment SNR

The Bayes error rate for this sub-experiment starts at 0.185 for $SNR = -5$, decreases quickly at first, and then starts to level off, achieving a final error rate of 0.04 for $SNR = 20$.

The three algorithms DDPPMF, KLT, and DP clearly outperform all the other algorithms obtaining near Bayes error rates for all values of SNR. The three algorithms DDPPSRE, LDB, KLTDPP and KLTBB obtain error rates that are consistent among each other, and slightly higher than the previously mentioned algorithms. The STD and WDP algorithms give significantly higher error rates than the other six algorithms for all values of SNR.

A curious phenomenon that is observed for the WDP algorithm is that the error rate seems to increase as the SNR increases from 5 to 20. This seems counter-intuitive but is completely understandable by the following argument. Since the sampling rate for this experiment is 64 and the number of samples used for training from each class is 64, the within-class covariance matrix of each class will theoretically be singular but due to round off errors will most likely just be badly scaled. As the signal to noise ratio (SNR) increases, the samples that are used to compute the covariance matrix become confined more and more to the subspace defined by the waveforms of the signal model which further increases the singularity of the within-class covariance matrix, and makes the scaling by the inverse of the matrix more problematic[3]. Referring to

[3]In some cases, the covariance matrix becomes truly singular making the inversion process im-

figures 5.4.1.2 and 5.5.1.2, it can be seen that the same phenomenon is observed for other experiments but to a lesser extent.

5.6.1.3 sub-experiment f_s

The Bayes error rate for this sub-experiment starts at 0.063 for $f_s = 32$, and gradually decreases to an error rate of 0.043 for $f_s = 256$.

The best performance is obtained for the DDPPMF, KLT and DP algorithms which achieve near Bayes error rates for all values of f_s. The DDPPSRE, KLTBB, KLTDPP, and LDB follow close behind with slightly higher error rates for all values of f_s. The STD and WDP algorithms show reasonable error rates for $f_s = 32$ but quickly escalate as f_s increases reaching error rates of ~ 0.45 for large values of f_s for the same reason that were discussed in section 5.4.1.3.

5.6.1.4 sub-experiment *Bases*

The Bayes error rate for this sub-experiment is 0.049 for all values of *Bases*.

For *Bases* $= \{1, 2\}$, the KLT algorithm clearly outperforms all the other algorithms obtaining error rates of ~ 0.06. The only other algorithm to obtain error rates below 0.1 for both of these sampling frequencies is the DDPPMF algorithm. For *Bases* ≥ 3, the DDPPMF, KLT and DP algorithms show the best performance with error rates of ~ 0.06. The DDPPSRE, KLTBB, KLTDPP and LDB algorithms show a more gradual decrease in error rate as *Bases* increases but achieve the same error rate as the previously mentioned algorithms for *Bases* ≥ 128. The WDP algorithm shows consistently poor result with some improvement as *Bases* increases and since the STD algorithm is not modified as *Bases* changes, it obtains consistent error rates of ~ 0.14.

5.6.1.5 summary

The original motivation of this experiment was to show that the LDB and KLTBB algorithms which use the best basis algorithm to optimize a criterion function should be required to keep more basis functions than the other algorithms to achieve the

possible. In these cases, the classifer fails competely and an error rate of ~ 0.5 is acheived

same error rate, as discussed in the introduction of this experiment. The results of this section seem to disprove that conjecture since the DDPPSRE and LDB algorithms have almost identical error rates as a function of *Bases*. Since these two algorithms optimize the same criterion function with different algorithms (*i.e.*, best basis and dictionary projection pursuit respectively), the DDPPSRE should have obtained lower error rates than LDB for small values of *Bases* if the conjecture were true.

However, the reason that this phenomenon was not observed may be because of a poorly designed dataset and not because the conjecture is false. The dataset was defined so that the deterministic waveforms in the matrix $\boldsymbol{A}$ (see section 5.2) exist on different scales and thus should conflict with the best basis strategy. However, the discriminant information in the dataset may exist in a subspace that is quite different from the waveforms that define the classes. As shown in figure 4.4, the features that the DDPPMF algorithm extracted for this dataset were highly oscillatory reflecting the pattern of the mean values for the waveforms and thus the discriminant features exist in a wavelet packet subspace with a large frequency index f. The waveforms that define that dataset on the other hand were all chosen from a wavelet packet subspace with frequency index $f = 0$. Despite this shortcoming, this dataset was maintained in order to be faithful with the original idea of creating the datasets before the experiments were performed and thus avoiding the bias of only presenting experiments where the dictionary projection pursuit algorithms performed best.

In any case, this experiment still shed much light on the performance of each of the algorithms. For this dataset, the DDPPMF, KLT and DP algorithms clearly performed the best, with KLT being slightly superior when only 1 or 2 basis functions were kept. The DDPPSRE and LDB algorithms performed almost identically in all cases but worse than the previously mentioned algorithms. This suggest that there is little difference in the performance of the best basis and dictionary projection pursuit algorithms, and that the modified Fisher criterion (which DDPPMF uses) is a superior criterion over the symmetric relative entropy criterion (which DDPPSRE uses).

The approximate KLT algorithms KLTBB and KLTDPP did not perform as well as the KLT algorithm, but they did perform well enough to enourage their use in high dimensional problems where the KLT algorithm cannot be applied.

The WDP algorithm performed poorly in all cases that the number of training samples from each class N was smaller than or on the same order of magnitude as the dimension of the feature vector f_s, due to the difficulties of inverting the singular within-class covariance matrix as discussed in section 5.4.1.1.

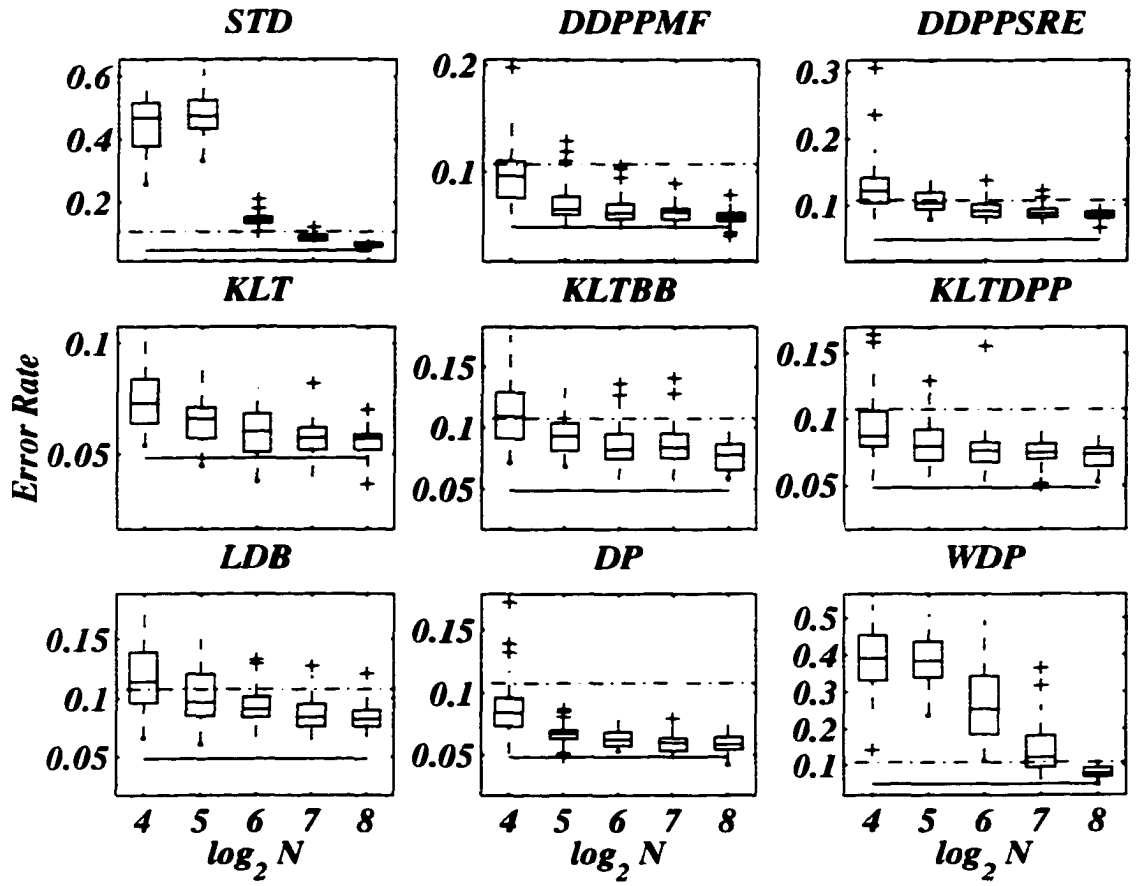

Figure 5.16. *Results for the multiscale waveform experiment. The variable N indicates the number of samples from each class that were used to train the classifier. The solid line shows the Bayes error rate and the dashed line shows upper y limit from the graph with the smallest upper y limit.*

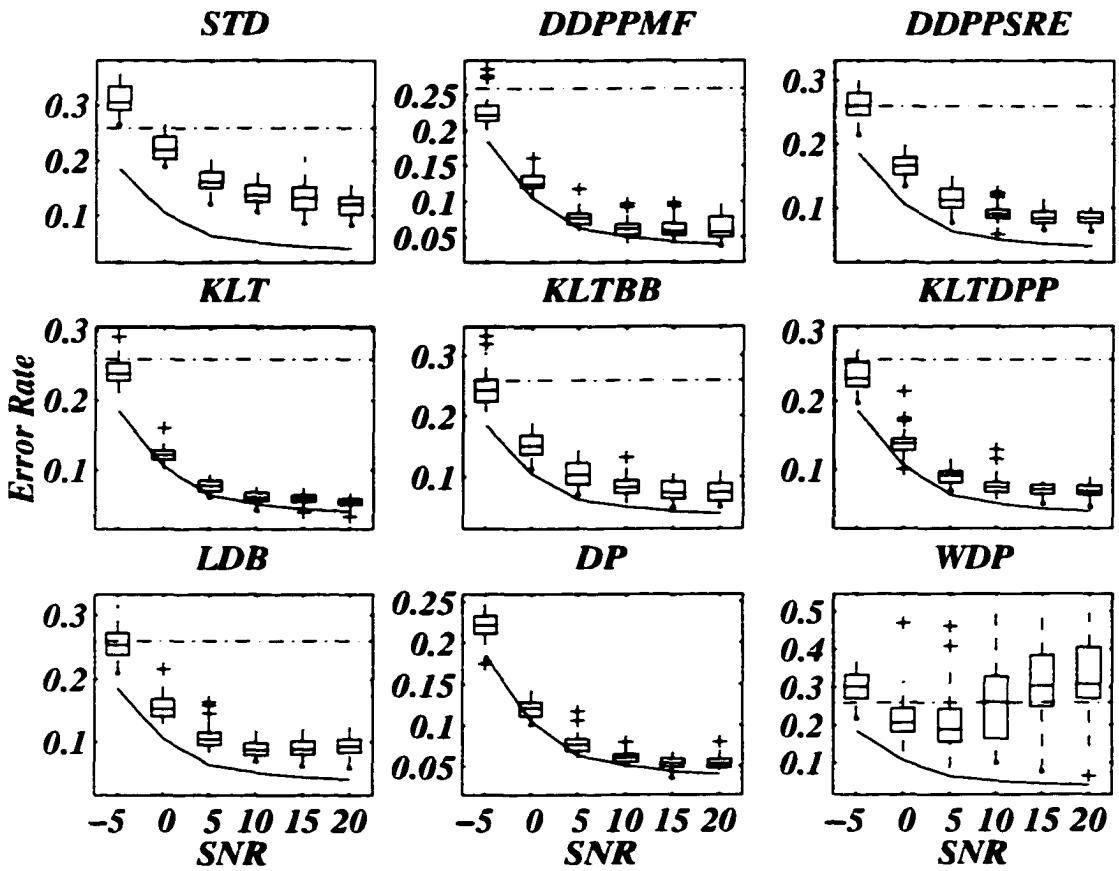

Figure 5.17. *Results for the multiscale waveform experiment. The variable SNR indicates the signal to noise ratio of the synthetic waveforms, as described in section 5.2.3. The solid line shows the Bayes error rate and the dashed line shows upper y limit from the graph with the smallest upper y limit.*

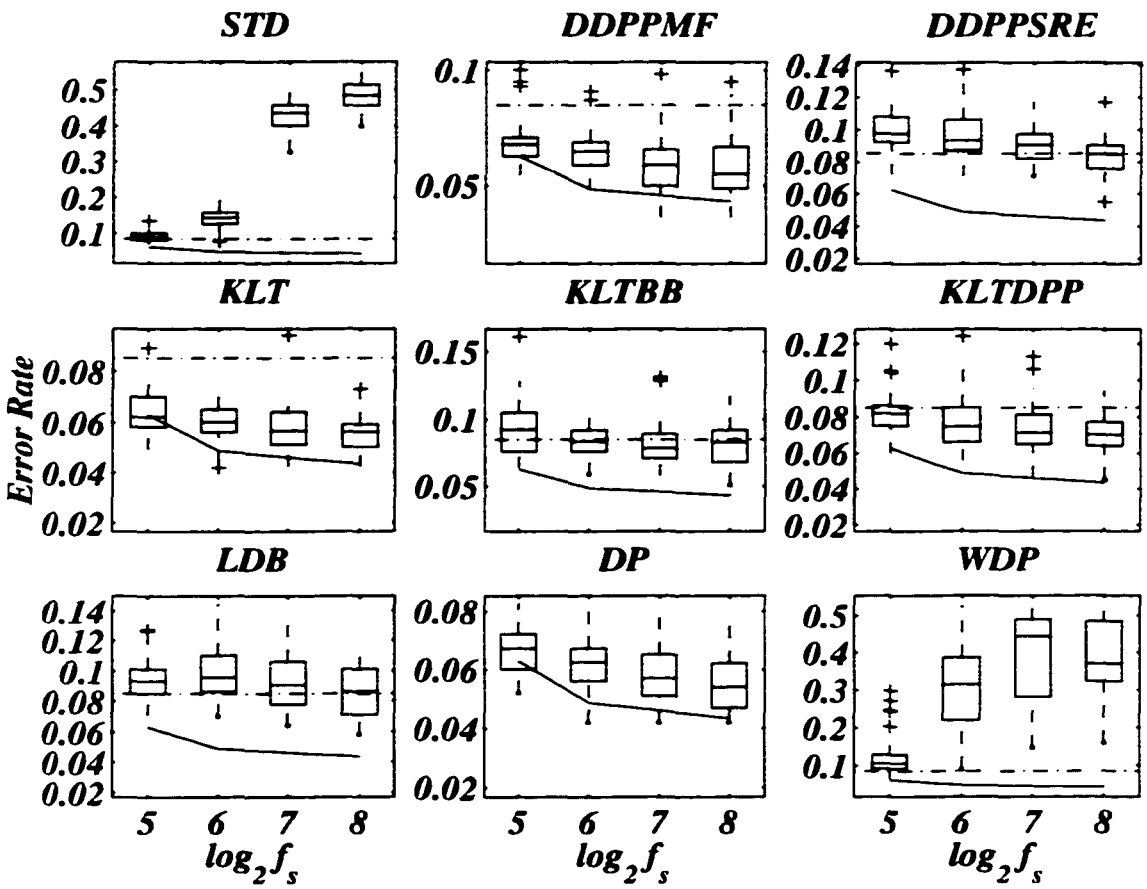

Figure 5.18. *Results for the multiscale waveform experiment. The variable f_s indicates the sampling frequency of the synthetic waveforms, as described in section 5.2.1. The solid line shows the Bayes error rate and the dashed line shows upper y limit from the graph with the smallest upper y limit.*

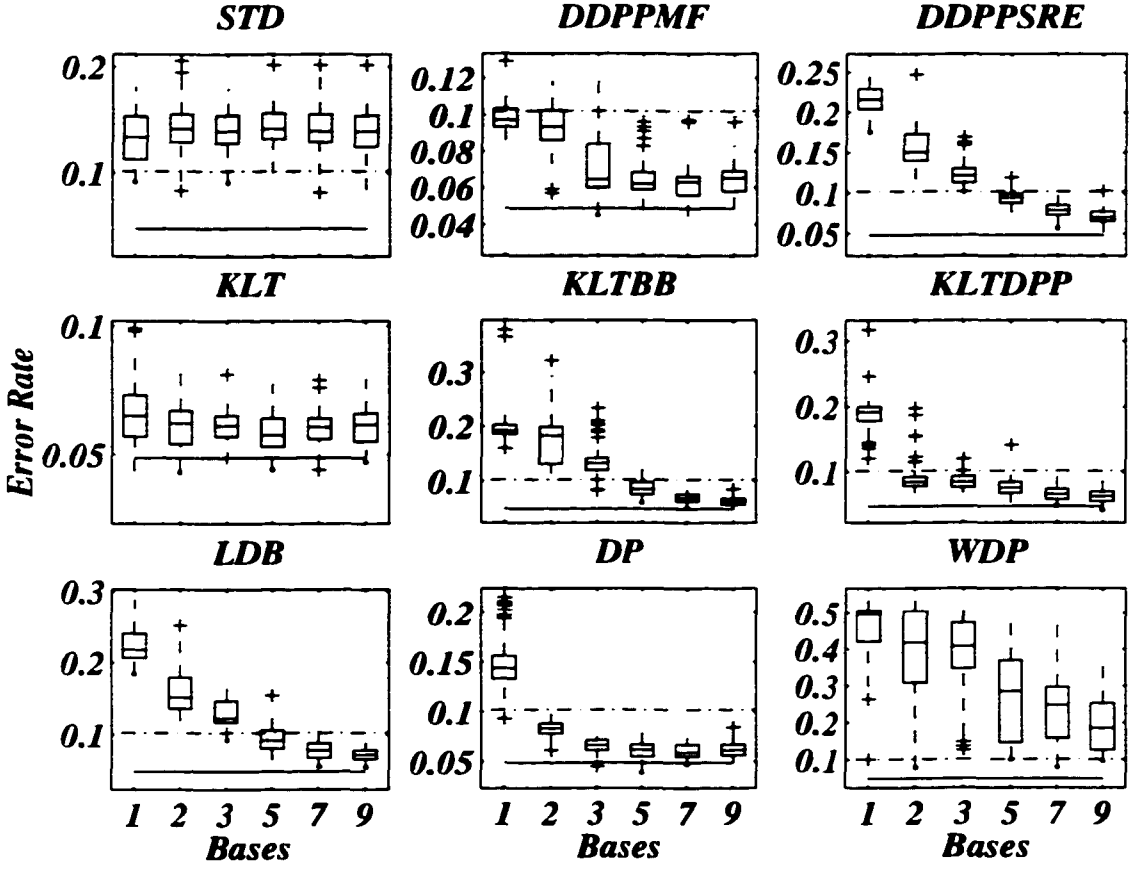

Figure 5.19. *Results for the multiscale waveform experiment. The variable Bases indicates the number of basis functions that the feature extraction method kept. The solid line shows the Bayes error rate and the dashed line shows upper y limit from the graph with the smallest upper y limit.*

Algorithm	16	32	64	128	256
	0.515	0.523	0.154	0.096	0.072
STD	**0.467**	**0.474**	**0.148**	**0.090**	**0.068**
	0.379	0.434	0.136	0.080	0.061
	0.110	0.077	0.070	0.065	0.061
DDPPMF	**0.097**	**0.065**	**0.061**	**0.062**	**0.058**
	0.076	0.060	0.056	0.055	0.054
	0.141	0.119	0.101	0.094	0.090
DDPPSRE	**0.122**	**0.103**	**0.091**	**0.088**	**0.085**
	0.103	0.094	0.083	0.082	0.081
	0.084	0.071	0.068	0.062	0.059
KLT	**0.073**	**0.066**	**0.060**	**0.058**	**0.057**
	0.064	0.057	0.051	0.052	0.052
	0.129	0.104	0.095	0.095	0.086
KLTBB	**0.109**	**0.093**	**0.082**	**0.084**	**0.077**
	0.091	0.081	0.074	0.075	0.065
	0.106	0.092	0.082	0.081	0.078
KLTDPP	**0.087**	**0.080**	**0.076**	**0.074**	**0.073**
	0.079	0.069	0.067	0.070	0.064
	0.139	0.120	0.101	0.095	0.090
LDB	**0.114**	**0.097**	**0.091**	**0.084**	**0.083**
	0.096	0.085	0.084	0.076	0.076
	0.096	0.070	0.068	0.063	0.064
DP	**0.084**	**0.067**	**0.062**	**0.059**	**0.058**
	0.073	0.063	0.057	0.053	0.054
	0.452	0.435	0.342	0.180	0.092
WDP	**0.390**	**0.383**	**0.253**	**0.122**	**0.077**
	0.329	0.338	0.184	0.094	0.069
Bayes	**0.048**	**0.048**	**0.048**	**0.048**	**0.048**

Table 5.11. *Results for sub-experiment N in the multiscale waveform experiment. The center bold number gives the median error rate, the upper number gives the 75^{th} percentile, and the lower number gives the 25^{th} percentile.*

Algorithm	-5	0	5	10	15	20
	0.335	0.243	0.179	0.155	0.152	0.133
STD	**0.307**	**0.219**	**0.161**	**0.137**	**0.132**	**0.121**
	0.293	0.203	0.150	0.125	0.112	0.103
	0.235	0.136	0.083	0.068	0.069	0.079
DDPPMF	**0.221**	**0.124**	**0.077**	**0.063**	**0.059**	**0.057**
	0.214	0.120	0.068	0.053	0.053	0.051
	0.281	0.179	0.130	0.097	0.094	0.093
DDPPSRE	**0.261**	**0.166**	**0.113**	**0.091**	**0.083**	**0.085**
	0.245	0.152	0.101	0.084	0.076	0.077
	0.254	0.127	0.084	0.067	0.063	0.058
KLT	**0.238**	**0.121**	**0.077**	**0.060**	**0.059**	**0.054**
	0.228	0.115	0.069	0.055	0.054	0.050
	0.260	0.168	0.123	0.094	0.092	0.089
KLTBB	**0.242**	**0.150**	**0.104**	**0.083**	**0.074**	**0.075**
	0.223	0.137	0.088	0.073	0.064	0.060
	0.256	0.144	0.097	0.081	0.077	0.076
KLTDPP	**0.232**	**0.138**	**0.092**	**0.075**	**0.071**	**0.068**
	0.220	0.128	0.081	0.067	0.065	0.064
	0.272	0.168	0.114	0.098	0.101	0.103
LDB	**0.254**	**0.152**	**0.104**	**0.087**	**0.088**	**0.093**
	0.237	0.140	0.095	0.079	0.080	0.083
	0.232	0.128	0.083	0.064	0.059	0.058
DP	**0.221**	**0.121**	**0.076**	**0.061**	**0.054**	**0.053**
	0.211	0.111	0.069	0.055	0.049	0.048
	0.333	0.244	0.242	0.328	0.383	0.404
WDP	**0.303**	**0.207**	**0.187**	**0.262**	**0.304**	**0.309**
	0.271	0.181	0.154	0.163	0.249	0.271
Bayes	**0.185**	**0.105**	**0.063**	**0.051**	**0.043**	**0.040**

Table 5.12. *Results for sub-experiment SNR in the multiscale waveform experiment. The center bold number gives the median error rate, the upper number gives the* 75^{th} *percentile, and the lower number gives the* 25^{th} *percentile.*

Algorithm	32	64	128	256
	0.100	0.157	0.460	0.515
STD	**0.090**	**0.143**	**0.436**	**0.486**
	0.081	0.127	0.402	0.457
	0.071	0.069	0.066	0.067
DDPPMF	**0.068**	**0.065**	**0.059**	**0.055**
	0.063	0.059	0.050	0.049
	0.108	0.106	0.097	0.090
DDPPSRE	**0.098**	**0.093**	**0.091**	**0.085**
	0.092	0.087	0.082	0.076
	0.070	0.065	0.064	0.059
KLT	**0.062**	**0.060**	**0.057**	**0.056**
	0.058	0.056	0.051	0.050
	0.105	0.092	0.089	0.092
KLTBB	**0.093**	**0.084**	**0.079**	**0.083**
	0.076	0.076	0.071	0.068
	0.086	0.085	0.081	0.077
KLTDPP	**0.082**	**0.075**	**0.071**	**0.070**
	0.075	0.066	0.065	0.064
	0.101	0.110	0.106	0.101
LDB	**0.093**	**0.096**	**0.091**	**0.086**
	0.085	0.086	0.078	0.071
	0.072	0.067	0.065	0.062
DP	**0.067**	**0.063**	**0.057**	**0.054**
	0.060	0.056	0.051	0.047
	0.128	0.387	0.490	0.485
WDP	**0.106**	**0.316**	**0.444**	**0.371**
	0.092	0.220	0.282	0.324
Bayes	**0.063**	**0.049**	**0.046**	**0.043**

Table 5.13. *Results for sub-experiment f_s in the multiscale waveform experiment. The center bold number gives the median error rate, the upper number gives the 75^{th} percentile, and the lower number gives the 25^{th} percentile.*

Algorithm	1	2	3	5	7	9
	0.153	0.155	0.154	0.156	0.155	0.154
STD	**0.134**	**0.142**	**0.139**	**0.142**	**0.139**	**0.140**
	0.113	0.129	0.128	0.132	0.129	0.125
	0.103	0.102	0.084	0.068	0.066	0.069
DDPPMF	**0.097**	**0.093**	**0.065**	**0.062**	**0.063**	**0.065**
	0.093	0.086	0.060	0.059	0.056	0.058
	0.229	0.173	0.130	0.099	0.085	0.077
DDPPSRE	**0.216**	**0.151**	**0.122**	**0.094**	**0.079**	**0.071**
	0.204	0.140	0.113	0.088	0.073	0.066
	0.073	0.067	0.065	0.064	0.064	0.066
KLT	**0.065**	**0.062**	**0.061**	**0.058**	**0.061**	**0.062**
	0.057	0.054	0.057	0.053	0.056	0.055
	0.203	0.199	0.141	0.097	0.076	0.067
KLTBB	**0.192**	**0.183**	**0.132**	**0.085**	**0.069**	**0.062**
	0.186	0.131	0.119	0.076	0.063	0.057
	0.199	0.092	0.093	0.085	0.074	0.070
KLTDPP	**0.192**	**0.085**	**0.085**	**0.076**	**0.067**	**0.064**
	0.178	0.079	0.077	0.068	0.059	0.056
	0.241	0.179	0.146	0.105	0.087	0.078
LDB	**0.218**	**0.152**	**0.122**	**0.092**	**0.078**	**0.072**
	0.207	0.136	0.117	0.080	0.067	0.066
	0.156	0.088	0.071	0.067	0.067	0.067
DP	**0.144**	**0.083**	**0.066**	**0.062**	**0.059**	**0.061**
	0.133	0.078	0.061	0.055	0.055	0.056
	0.507	0.505	0.474	0.371	0.298	0.254
WDP	**0.498**	**0.419**	**0.409**	**0.285**	**0.247**	**0.186**
	0.422	0.310	0.349	0.147	0.158	0.127
Bayes	**0.049**	**0.049**	**0.049**	**0.049**	**0.049**	**0.049**

Table 5.14. *Results for sub-experiment Bases in the multiscale waveform experiment. The center bold number gives the median error rate, the upper number gives the* 75^{th} *percentile, and the lower number gives the* 25^{th} *percentile.*

5.7 Conclusions

The results from the synthetic experiments performed in this chapter allow the following conclusions to be drawn.

1. The DDPPMF algorithm performed very well for all of the experiments achieving either the lowest error rate or close to the lowest error rate in all cases.
2. The modified Fisher (MF) criterion (section 4.6.1.2) is a better projection criterion than the symmetric realtive entropy (SRE) criterion (section 4.6.1.1) when used with the discriminant dictionary projection pursuit (DDPP) algorithm in all cases. Since the LDB algorithm cannot use the MF criterion, this presents a distinct advantage of DDPP over LDB as a dictionary optimization algorithm.
3. The LDB algorithm outperforms the DDPPSRE algorithm only when most of the discriminant information for a problem exists in the same wavelet packet subspace, and in all other cases they perform identically. Since both algorithms use the same criterion function, this statement reflects the behaviour of the best basis and dictionary projection pursuit optimization algorithms as well.
4. The WDP algorithm performs poorly in all cases where the number of training samples for each class N is smaller than or on the same order of magnitude as the dimension of the feature vector f_s.
5. The KLT algorithm is a very powerful algorithm that performs as well or better than all of the other algorithms in all cases except when the discriminant information in a problem is contained in a small variance subspace.
6. The approximate KLT algorithms KLTBB and KLTDPP perform slightly worse than KLT in most cases but sufficiently well to warrant their use in higher dimensional problems where KLT is too computationally expensive to implement.

Chapter 6

Experimental Results for Recorded Data

6.1 Introduction

The experiments in this chapter use datasets of recorded sounds from established databases to show the relative strengths and weaknesses of each of the feature extraction algorithms described in section 4.7. In all cases, Fisher's LDA described in section 2.2.4.4 was used as a classifier. This classifier was chosen due to its widespread use and acceptance as a robust classifier. Simulations with other classifiers such as linear and quadratic Gaussian plug-in, and CART (see section 2.2.4) show the same performance trends for each of the feature extraction techniques described here, so they are not presented.

6.2 Noise Monitoring

6.2.1 Introduction

Noise monitoring is becoming increasingly important, especially in highly populated areas near airports, train stations and highways where noise regulations strictly specify the acceptable noise levels. Historically, noise monitoring consisted of measuring average sound levels over relatively long time periods (hours or days), but recently, there has been more emphasis on determining the nature of the sound so that more effective control can be implemented. The latest work in this field has been compiled and summarized in Couvreur's thesis [25], where he also develops a pattern

recognition framework based on a 1/3 octave pre-processor with linear and quadratic classifiers. One of the goals of this section is to show that adaptive feature extraction techniques applied to short time log periodograms can significantly reduce the error rates over the use of fixed 1/3 octave filter bank features, which is currently considered the state of the art by the noise monitoring community. Additionally, this section is used to evaluate the adaptive feature extraction techniques against one another using real data.

6.2.2 Madras Database

The MADRAS[1] database of environmental noise sources has been constructed for the purpose of developing new monitoring instruments with the ability to automatically identify and quantify the various acoustic sources that make up a given acoustic environment. The MADRAS project is a European consortium with several partners contributing to the funding and research [26]. The database consists of several hundred high quality recordings of common environmental sounds such as planes, cars, trucks, trains, factories, machine shops, chain saws, etc., and the recording conditions of each noise source are documented[2]. All the recordings used in the following experiments have sampling rates of 25600 Hz and 16 bit resolution.

For the experiments in this section, sound recordings were selected of planes, trains, cars and trucks from the MADRAS database. The planes are all propeller planes; the train sounds contain both the track noise and the engine noise; the car and truck sounds are all drive-bys at relatively close proximity (within 3m), and a combination of road noise and engine noise is heard. A total of 16 recordings from each class were chosen, which resulted in a total of 11 minutes of recordings. Recordings of planes and trains were longer than those of cars and trucks; so they make up proportionally more of the recording time.

[1]MADRAS - Methods for Automatic Detection and Recognition of Acoustic Sources.

[2]This data was kindly provided by Christophe Couvreur of Lernout and Hauspie Inc. with permission from the MADRAS group.

Noise Source	Occurrence
car	683
truck	1510
plane	1980
train	3914

Table 6.1. *MADRAS Occurrence Table*

6.2.3 Pre-processing

Two different pre-processing methods were used. The first mimics the 1/3 octave pre-processing that Couvreur used in his work [25] (it actually uses the same software) and is intended to represent a low dimensional manual form of feature extraction based on *a priori* knowledge. The second computes the log periodogram and is intended to represent raw data that requires an algorithm to adaptively search for features based on a set of training data. The frame size in both cases was set to 80 ms without overlap. This is significantly shorter than the 1 s frames that Couvreur [25] used which he chose based on the argument that current noise monitoring equipment already produces one third octave spectra with this time resolution. However, extensive testing showed that classification accuracy does not improve for window sizes above 80 ms, so this window size was used for our experiments. This results in a total of 8087 frames, with the relative numbers in each class shown in table 6.1.

6.2.3.1 1/3 Octave Pre-processor

The 1/3 octave pre-processor is described by Couvreur [25], so only a cursory description is given here. The pre-processor forms 18-dimensional feature vectors from the partial powers in the third octave bands ranging from 100 Hz to 5000 Hz. The partial power in a third octave band is the ratio of the portion of the signal RMS power contained in that band to the total RMS power of the signal[3]. The passbands for each of the filters in the bank are plotted in figure 6.1, and typical spectra of each of the classes are plotted in figure 6.2. Notice that the passbands are equally spaced and have equal widths when plotted on a logarithmic x-axis. This is commonly called

[3]Actually, they should be called partial energies since it represents an integral of a power, but this terminology has been used extensively in the literature so we will not attempt to change it.

constant-Q filtering. On a linear scale, the filter passbands become progressively wider and spaced further apart as the center frequency of the passband increases. The term 1/3 octave is due to the fact that there are exactly 3 passbands per octave (*i.e.*, power of 2). The only modification made from Couvreur's procedure was to add 20 to the dB values of the partial powers. This was done to conveniently set the zero point of the scale to 1 % of the total signal power, which makes visual inpsection of the spectra more intuitive.

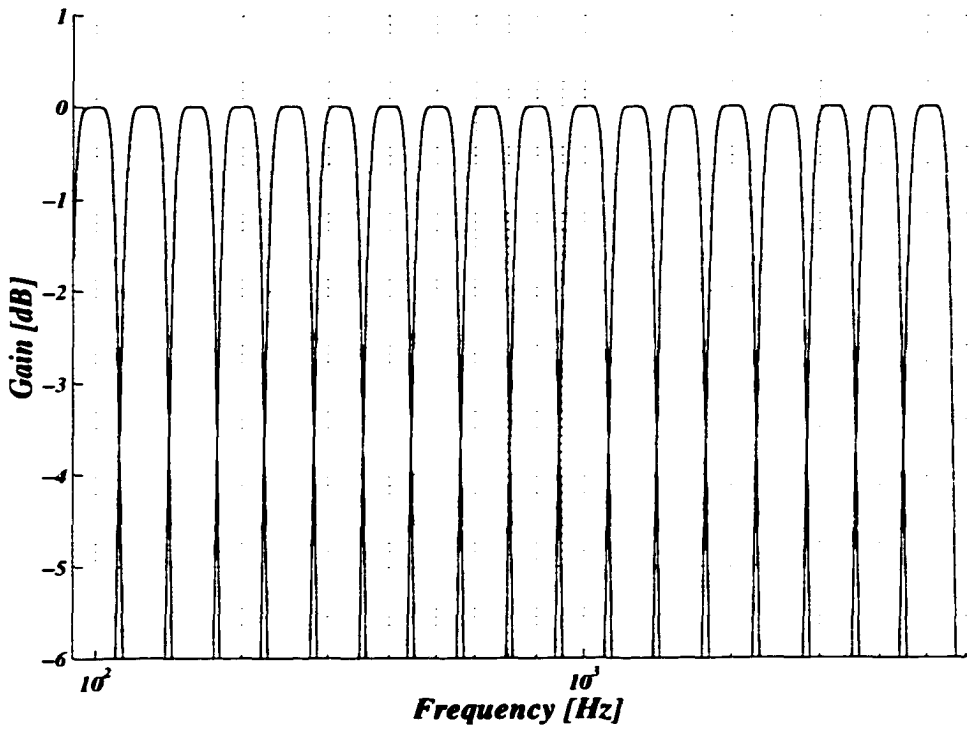

Figure 6.1. *Passbands for the 1/3 octave filter bank.*

6.2.3.2 Log Periodogram Pre-processor

The log periodogram[4] pre-processing of the MADRAS database recordings consisted of the following steps:

[4]This is just the windowed FFT amplitude spectrum as described in section 2.4.2.1. The term periodogram is used to be consistent with the literature [13, 58].

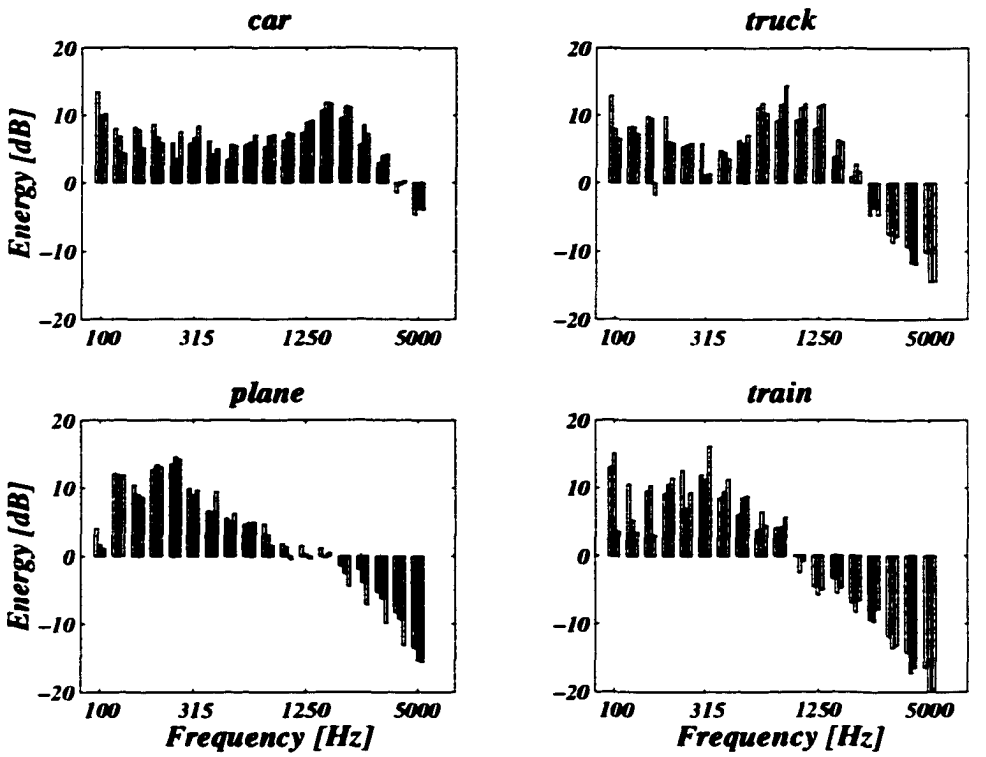

Figure 6.2. *Typical 1/3 octave spectra from the MADRAS database. Each band shows the partial power from three separate recordings from the class. A given recording is located in the same relative position in each band.*

1. The time domain signals were downsampled by 2 using a sixth order Chebychev filter with 0.5 dB ripple and 10.75 kHz corner to drop the sampling rate from $f_s = 25.6$ kHz to 12.8 kHz.
2. The time domain signals were high pass filtered with a sixth order Butterworth filter with the -3 dB point set to 75 Hz. This was done to remove low frequency artifacts from the car and truck recordings.
3. The time domain signal was buffered into 512 sample non-overlapping frames.
4. A Hanning window[5] was applied to each frame.
5. The FFT of each frame was computed, which will be denoted as $\boldsymbol{y}$.

[5]As pointed out by Antoniou [1], the proper name for this window is the von Hann window, but since it is almost exclusively called the Hanning window in the literature, we will not try to undo the wrong-doings of early researchers.

6. The normalized periodogram in dB was then computed as

$$x[i] = 10 \log \frac{|y[i]|^2}{\sum_i |y[i]|^2} + 20, \tag{6.1}$$

where the additional 20 makes 1% of the total signal energy the zero point of the dB scale, as was done for the 1/3 octave pre-processor. The only operation that is critical for pattern recognition is the log function. The transformation to the dB scale was done only because it is a familiar scale for humans to understand.

Typical spectra of each of the classes are plotted in figure 6.3. Notice that the shape of these periodograms are quite different than the shape of the 1/3 octave spectra in figure 6.2. In particular, the high frequency energy is higher in the 1/3 octave spectra than the periodogram spectra. This is because the 1/3 octave filter bank integrates progressively more of the energy into a single band as the center frequency of the passband increases.

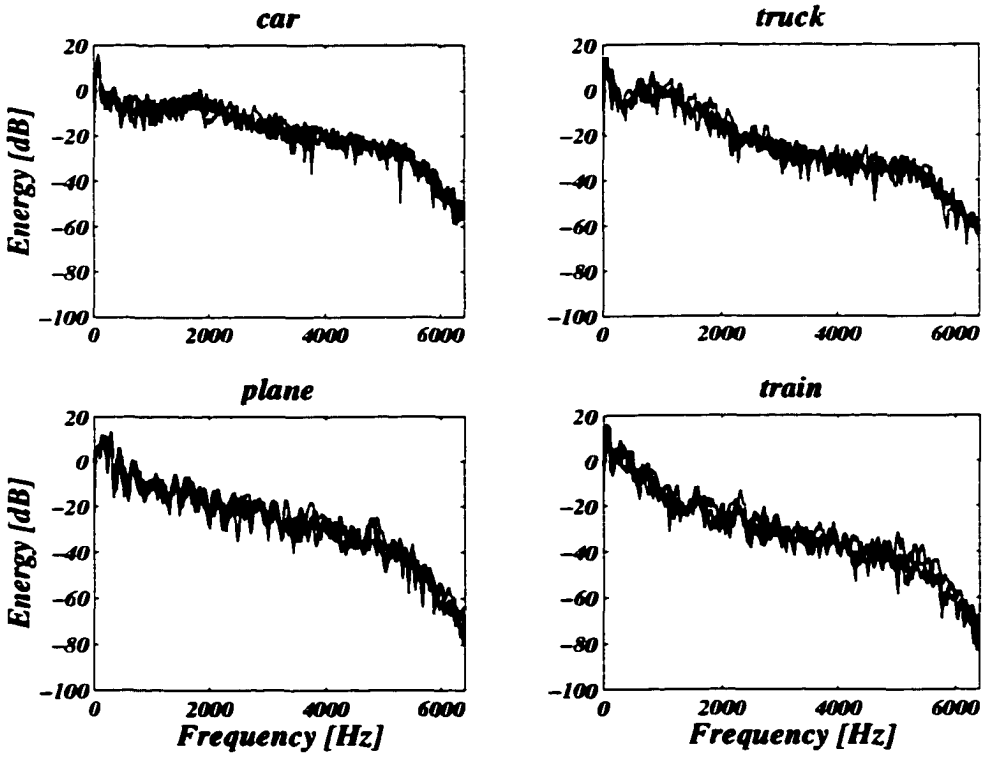

Figure 6.3. *Typical log periodograms from the MADRAS database.*

6.2.4 Experimental Setup

It is not reasonable to assume that frames from the same physical recording are independent, as is usually assumed for the evaluation of pattern recognition systems, so special care was taken to ensure that frames from the same recording were not used for both training and evaluation. That is, if a given frame is used for training the adapted feature extraction algorithm and classifier, then any frame from the same recording is prohibited from being used to evaluate the system. If this precaution was not taken, then the evaluation results would be biased to lower error rates than the true error rates.

The fixed 1/3 octave features (section 6.2.3.1) and each adapted feature extraction technique applied to the log periodogram features (section 6.2.3.2) were evaluated in terms of their cross validation error rate (section 2.2.6.3) on a scale of 0–1, as a function of,

N - the *total* number of frames that were used to train the adapted feature extraction algorithm and classifier.

Bases - The number of basis functions that were kept for the adapted techniques only (*i.e.*, the 1/3 octave preprocessor always keeps 18 basis functions and the STD method keeps all the basis functions).

In the experiments where N is varying, *Bases* is fixed at 25, and when *Bases* is varying, N is fixed at 128.

The procedure can be summarized as follows,

1. Partition the dataset into 16 sections, such that each section contains one recording from each class.
2. Classify the frames in each section by randomly selecting N frames from the remainder of the dataset to use as a training set.

In this way, every frame is classified once using independent training data and a total of 16 trials are performed which gives a distribution of error rates that can be displayed using box plots as described in section 5.3.1.

6.2.5 Experimental Results

The experimental results of applying each feature extraction algorithm (section 4.7) to the MADRAS dataset for each sub-experiment (section 6.2.4) are plotted in figures 6.4–6.6. A tabular presentation of the results for each sub-experiment are given in tables 6.2–6.3.

6.2.5.1 sub-experiment *N*

For $N = 32$, the KLT algorithm performs very well giving an error rate of ~ 0.19, while all other algorithms give error rates greater than ~ 0.4. For larger values of N the best algorithms are KLT, KLTBB, and LDB which achieve error rates as low as ~ 0.1. The next best algorithm in this regime is DDPPSRE, followed close behind by DDPPMF, KLTDPP and OCT3.

The STD and WDP methods perform poorly for $N \leq 256$ but begin to show improvements for larger values of N, and actually achieve error rates close to but slightly higher than the other algorithms for $N = 1024$. The reason for the poor performance of these algorithms for small N is that both use the full size 256×256 within-class covariance matrix $\mathbf{\Sigma}_W$ which is guaranteed to be singular or at least badly scaled. The STD method doesn't reduce the dimension of the feature vector at all, so Fisher's LDA (see section 2.2.4.4) uses the poorly scaled $\mathbf{\Sigma}_W$ to solve the generalized eigenvalue problem and thus has large errors. The WDP algorithm attempts to scale the contrasts between the class means by $\mathbf{\Sigma}_W^{-1}$ (see section 4.7.9) which when $\mathbf{\Sigma}_W$ is poorly scaled produces erroneous results.

6.2.5.2 sub-experiment *Bases*

Not a lot of extra information was obtained from this sub-experiment. The same performance trends that were observed in sub-experiment N are observed here, and in all cases, the performance of each algorithm seems to be relatively independent of the number of basis functions that are kept.

6.2.5.3 summary

One of the goals of this experiment was to show that the classification performance of noise monitoring data can be improved by using adapted feature extraction techniques rather than fixed 1/3 octave features which is currently the standard in the field. This objective was achieved since for $N \geq 128$ and $Bases = 25$, the KLT, KLTBB, and LDB algorithms reduced the OCT3 median error rate by approximately 40 %, DDPPSRE reduced the OCT3 median error rate by approximately 25 %, and DDPPMF reduced the OCT3 median error rate by approximately 10 %.

Another interesting observation is that the SRE criterion (see section 4.6.1.1) appears to be a better criterion for this problem than the MF criterion (see section 4.6.1.2) since DDPPSRE outperforms DDPPMF. This is in contrast to the results for the synthetic data in chapter 5, where the MF criterion outperformed SRE in all cases. This paradox raises an important point about pattern recognition problems which is that there is no universally best method. The value of a given method is very strongly dependent on the type of data that is being analyzed.

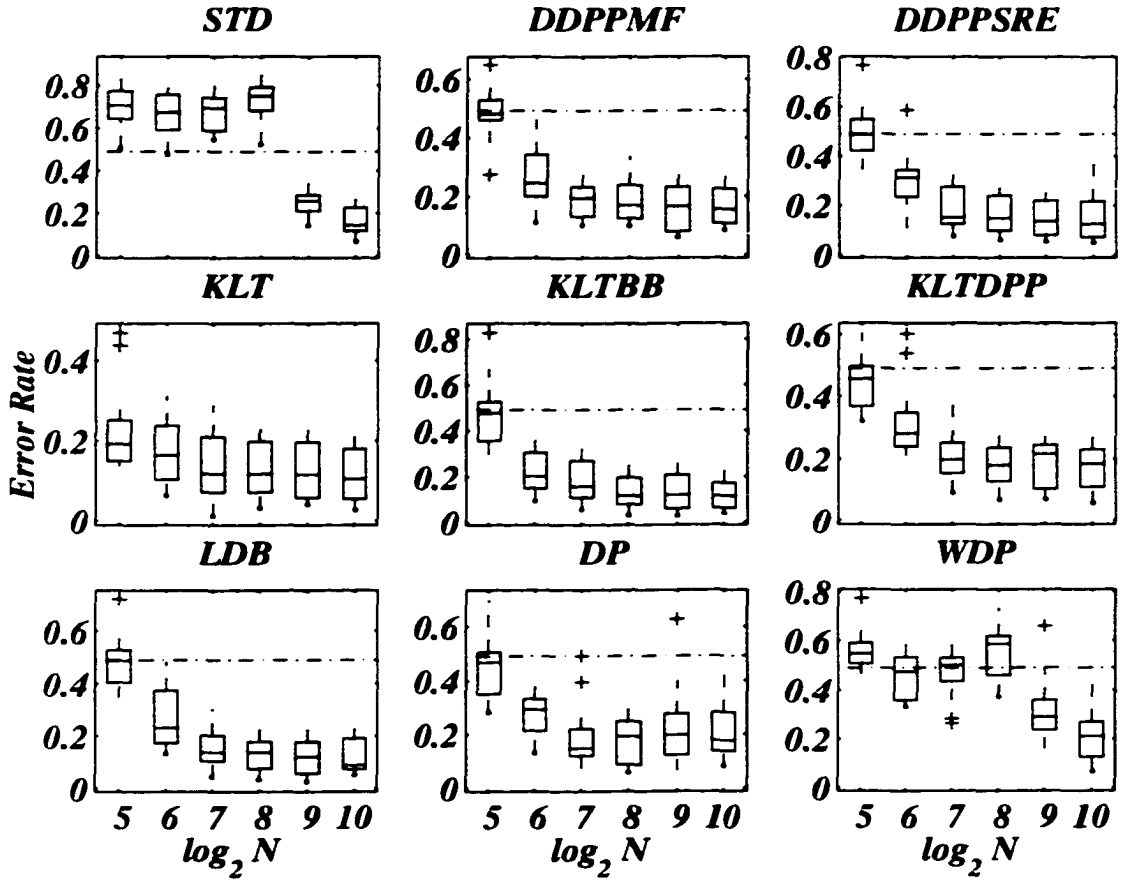

Figure 6.4. *Results for the Madras experiment. The variable N indicates the total number of frames that were used to train the feature extraction algorithms and classifier.*

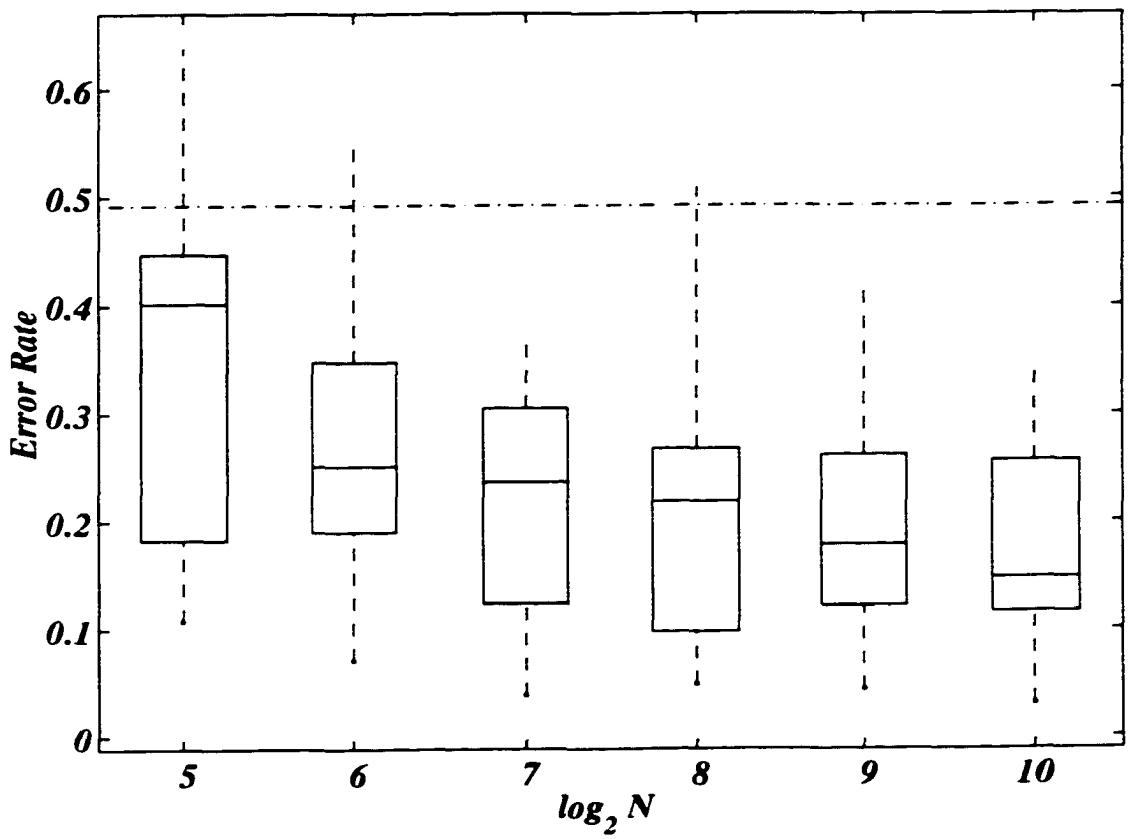

Figure 6.5. *Results for the Madras experiment using 1/3 octave filter bank mean square energies as features. The variable N indicates the total number of frames that were used to train the classifier.*

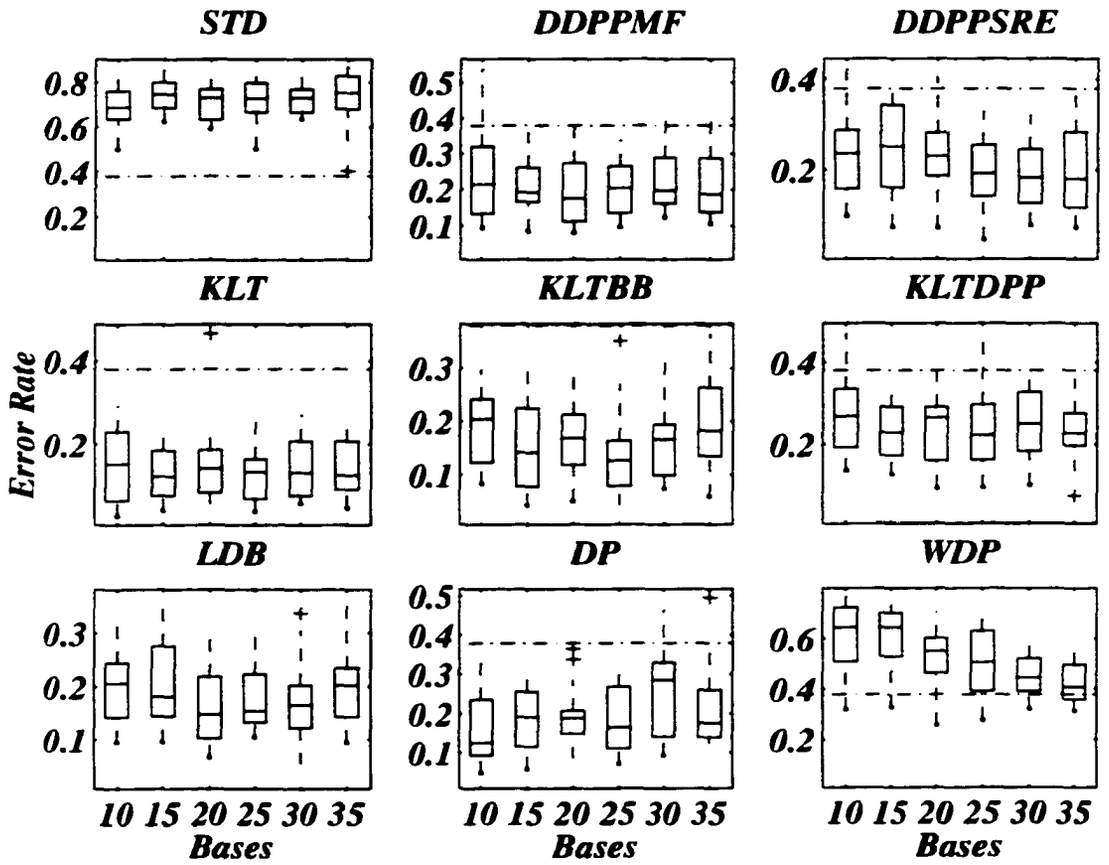

Figure 6.6. *Results for the Madras experiment. The variable Bases indicates the number of basis functions that were kept by the adapted feature extraction technique.*

Algorithm	32	64	128	256	512	1024
	0.775	0.755	0.741	0.792	0.283	0.226
STD	**0.707**	**0.674**	**0.694**	**0.749**	**0.260**	**0.144**
	0.644	0.590	0.589	0.683	0.210	0.116
	0.529	0.343	0.231	0.237	0.230	0.226
DDPPMF	**0.481**	**0.244**	**0.192**	**0.170**	**0.165**	**0.154**
	0.460	0.199	0.131	0.124	0.080	0.107
	0.550	0.345	0.277	0.241	0.222	0.218
DDPPSRE	**0.488**	**0.314**	**0.151**	**0.150**	**0.136**	**0.127**
	0.424	0.237	0.126	0.100	0.082	0.073
	0.252	0.237	0.209	0.198	0.197	0.179
KLT	**0.192**	**0.163**	**0.116**	**0.116**	**0.115**	**0.105**
	0.150	0.102	0.072	0.071	0.058	0.055
	0.527	0.306	0.271	0.197	0.207	0.172
KLTBB	**0.478**	**0.201**	**0.157**	**0.118**	**0.120**	**0.114**
	0.359	0.150	0.110	0.080	0.059	0.063
	0.497	0.348	0.253	0.235	0.245	0.230
KLTDPP	**0.457**	**0.283**	**0.197**	**0.180**	**0.216**	**0.183**
	0.369	0.240	0.155	0.127	0.103	0.109
	0.529	0.373	0.203	0.178	0.179	0.191
LDB	**0.490**	**0.230**	**0.139**	**0.138**	**0.122**	**0.089**
	0.405	0.175	0.106	0.075	0.057	0.073
	0.508	0.333	0.222	0.249	0.276	0.281
DP	**0.469**	**0.293**	**0.149**	**0.192**	**0.198**	**0.178**
	0.352	0.214	0.122	0.088	0.123	0.137
	0.590	0.536	0.534	0.618	0.359	0.269
WDP	**0.551**	**0.477**	**0.501**	**0.586**	**0.291**	**0.211**
	0.510	0.359	0.434	0.460	0.236	0.125
	0.447	0.347	0.305	0.267	0.261	0.256
OCT3	**0.401**	**0.251**	**0.236**	**0.219**	**0.178**	**0.148**
	0.182	0.190	0.124	0.098	0.122	0.116

Table 6.2. *Results for sub-experiment N in the MADRAS experiment. The center bold number gives the median error rate, the upper number gives the 75^{th} percentile, and the lower number gives the 25^{th} percentile.*

Algorithm	10	15	20	25	30	35
	0.763	0.801	0.771	0.799	0.769	0.830
STD	**0.690**	**0.747**	**0.730**	**0.728**	**0.732**	**0.754**
	0.637	0.687	0.635	0.667	0.666	0.681
	0.322	0.263	0.276	0.267	0.291	0.289
DDPPMF	**0.215**	**0.193**	**0.175**	**0.205**	**0.198**	**0.187**
	0.135	0.166	0.112	0.138	0.162	0.140
	0.288	0.343	0.283	0.256	0.247	0.285
DDPPSRE	**0.238**	**0.253**	**0.231**	**0.193**	**0.185**	**0.181**
	0.160	0.162	0.188	0.144	0.128	0.119
	0.230	0.185	0.186	0.165	0.207	0.207
KLT	**0.153**	**0.123**	**0.142**	**0.133**	**0.132**	**0.125**
	0.064	0.077	0.085	0.069	0.077	0.091
	0.242	0.225	0.213	0.164	0.194	0.264
KLTBB	**0.205**	**0.142**	**0.169**	**0.127**	**0.167**	**0.183**
	0.123	0.078	0.118	0.079	0.099	0.135
	0.335	0.292	0.291	0.297	0.327	0.276
KLTDPP	**0.269**	**0.229**	**0.266**	**0.222**	**0.251**	**0.227**
	0.192	0.173	0.160	0.162	0.183	0.198
	0.244	0.275	0.218	0.224	0.202	0.234
LDB	**0.206**	**0.181**	**0.148**	**0.154**	**0.165**	**0.203**
	0.142	0.144	0.103	0.132	0.123	0.143
	0.237	0.256	0.208	0.269	0.331	0.262
DP	**0.125**	**0.191**	**0.189**	**0.164**	**0.287**	**0.176**
	0.093	0.116	0.148	0.110	0.141	0.139
	0.725	0.702	0.603	0.630	0.522	0.498
WDP	**0.645**	**0.645**	**0.551**	**0.507**	**0.445**	**0.408**
	0.509	0.530	0.465	0.393	0.393	0.359

Table 6.3. *Results for sub-experiment Bases in the MADRAS experiment. The center bold number gives the median error rate, the upper number gives the* 75^{th} *percentile, and the lower number gives the* 25^{th} *percentile.*

6.3 Phoneme Classification

6.3.1 Introduction

As discussed in chapter 1, speech recognition is one of the main driving forces behind research in acoustic pattern recognition. There are many classifier styles used for speech recognition, including frame classifiers (section 2.3.1), multi-frame classifiers (section 2.3.2), and hidden Markov models (section 2.3.3), but in all cases, good features at the frame-level are required for the successful operation of the system as a whole. As discussed in section 2.2.1, the role of pattern recognition is to convert physical observations into symbols, but for continuous speech, it is not clear as to what level this transformation should take place. Should it take place at the phoneme level, the word level, or the sentence level? Most often, recognition occurs at the phoneme level, and higher-level models, usually based on dynamic programming or hidden Markov modelling, are used to combine phonemes into words, and finally grammatical models are used to convert words into sentences [72, 102]. Selecting good features for phoneme classification thus plays a central role in almost all speech recognition systems. The goal of this section is to evaluate the adapted feature extraction techniques against one another using real data for a problem that is well known and well studied by the speech recognition community.

6.3.2 Phoneme Database

The signals used in the phoneme database were extracted from the TIMIT corpus[6] which is a standardized archive widely used by the speech recognition community. The example used in this section was also studied by Hastie et al. [58] to evaluate their Penalized Discriminant Analysis (PDA) algorithm, and Buckheit and Donoho [13] to evaluate their Discriminant Pursuit (DP) algorithm[7]. The classification problem consists of discriminating between five different phoneme types for approximately 50 different male speakers based on the log periodograms of digitized continuous speech framed in 32 ms segments. The sampling rate was 16 kHz, so each frame contained 512 samples in the time domain which produces 256 sample log periodograms (*i.e.*,

[6]TIMIT Acoustic-Phonetic Continuous Speech Corpus, US Dept of Commerce.

[7]This data was kindly provided by Trevor Hastie of Stanford University.

Phoneme	as in word (bold)	Occurrence
sh	**she**	872
iy	**in**	1163
aa	**dark**	695
dcl	**dark**	757
ao	**water**	1022

Table 6.4. *Phoneme Occurrence Table*

the feature vectors for this problem have a dimension of 256). There are a total of 4509 frames with the number and type of each phoneme class given in table 6.4, and typical spectra of each of the classes plotted in figure 6.7.

6.3.3 Experimental Setup

The setup for this experiment closely follows the procedure used by Hastie et al. [58], which ensures that a given speaker is not used both in the training set and evaluation set. Each adapted feature extraction technique was applied to the log periodogram features and was evaluated in terms of its holdout error rate (section 2.2.6.2) on a scale of 0–1, as a function of,

N - the number of frames that were used to train the adapted feature extraction algorithm and classifier.

Bases - The number of basis functions that were kept. Note that the STD method keeps all the basis functions regardless of the value of *Bases*.

In the experiments that N is varying, *Bases* is fixed at 25, and when *Bases* is varying, N is fixed at 128.

The procedure can be summarized as follows,

1. Choose 50 speakers randomly from the dataset and use all the frames associated with these speakers as the evaluation set $\mathcal{L}_e$.
2. From the remainder of the speakers in the dataset, choose N frames randomly for the training set $\mathcal{L}_t$.
3. Train a classifier with $\mathcal{L}_t$ and evaluate the classifier with $\mathcal{L}_e$.
4. Repeat the above three steps 25 times.

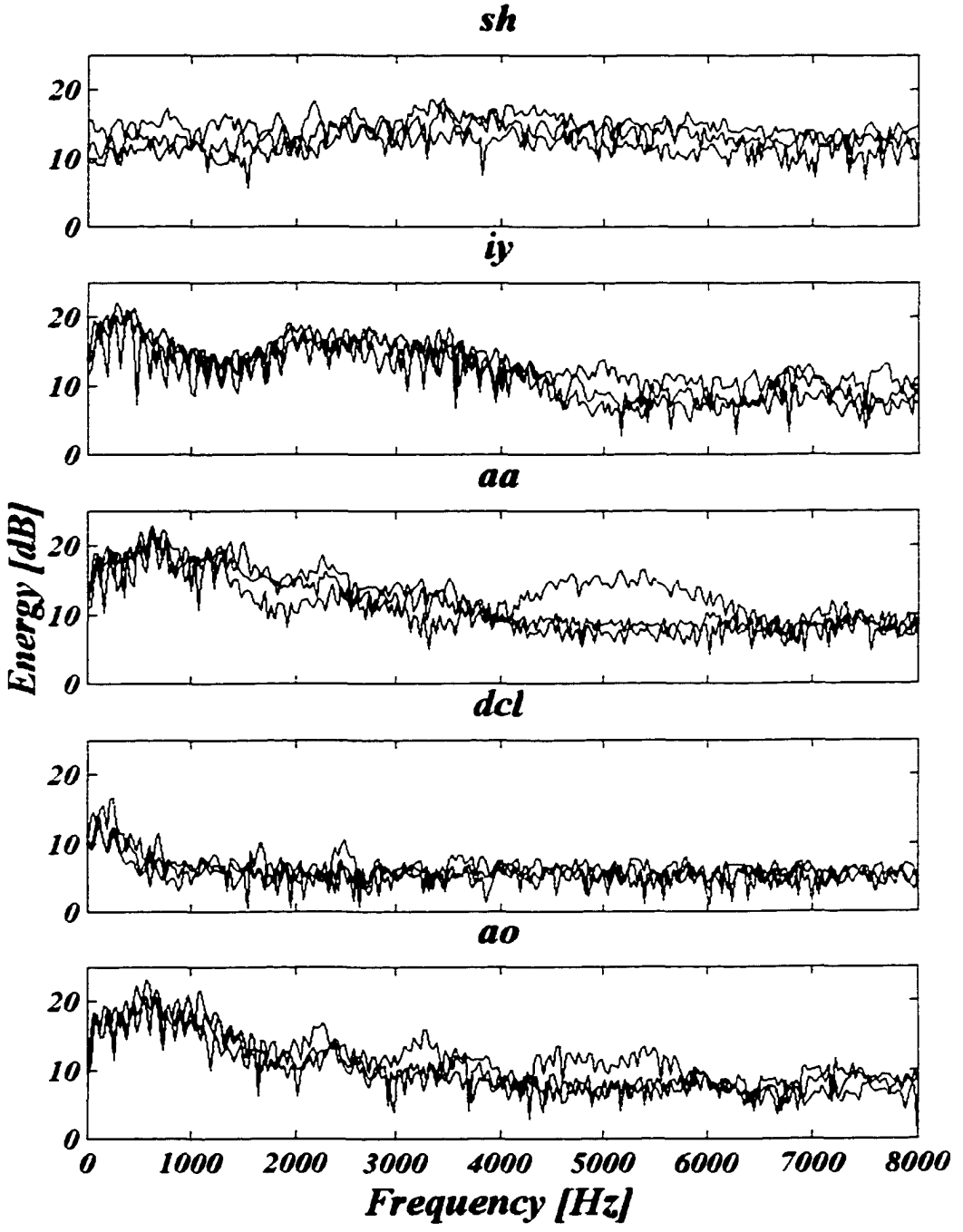

Figure 6.7. *Typical log periodograms from the Phoneme database.*

In this way, a total of 25 trials are performed which gives a distribution of error rates that can be displayed using box plots as described in section 5.3.1.

6.3.4 Experimental Results

The experimental results of applying each feature extraction algorithm (section 4.7) to the phoneme dataset for each sub-experiment (section 6.3.3) are plotted in figures 6.8–6.9. A tabular presentation of the results for each sub-experiment are given in tables 6.5–6.6.

6.3.4.1 sub-experiment *N*

For $N = 32$ the KLT algorithm obtains the lowest error rate of 0.14, while all other techniques have error rates greater than ~ 0.24. However, for $N \geq 64$, DDPPMF, DDPPSRE, KLT, KLTBB, KLTDPP, and LDB all have very low and essentially identical error rates. The STD and WDP algorithm show poor performance for $N \leq 512$ but show reasonable results for $N = 1024$.

The DP algorithm has poor results for all N with error rates greater than ~ 0.13. The reason for the poor performance of this algorithm is that it does not take into account the covariance of the dataset and it iteratively chooses features that maximize the contrast between two classes in the dataset rather than choosing a feature that has a large average contrast between all classes. For this reason, it keeps choosing features that separate the classes that are easy to discriminate between (*i.e.*, they have a large contrast), and never chooses features that are hard to discriminate between (*i.e.*, they have small contrasts). Therefore most of the errors that are incurred for this algorithm occur between the same classes (Buckheit and Donoho [13] observed this fact as well). This is one of the reasons why it is so important to use both synthetic and real data when testing algorithms, since none of the synthetic datasets were created in such a way to reveal this fact.

6.3.4.2 sub-experiment *Bases*

As in the MADRAS experiment, this sub-experiment contains relatively little information. The same performance trends that were observed in sub-experiment *N*

are observed here, and in all cases, the performance of each algorithm seems to be relatively independent of the number of basis functions that are kept.

6.3.4.3 Summary

The most amazing thing about this experiment was how similar the error rates were for the algorithms DDPPMF, DDPPSRE, KLT, KLTBB, KLTDPP, and LDB. They all obtain essentially identical results for all values of $N \geq 64$. There does not appear to be a descrepancy between the performance of the MF criterion and SRE criterion as there was in the synthetic experiments of chapter 5 and in the MADRAS experiment in this chapter. These algorithms clearly outperform the STD, DP and WDP algorithms.

Since this dataset has been studied by other authors, it is possible to do a comparison with their work. Hastie et al. [58] studied this data with their penalized discriminant analysis (PDA) algorithm. They used $N = 1000$ training samples and performed 50 trials to obtain a median error rate. They obtain an error rate of 0.087 for Fisher's LDA without any penalization which is slightly lower but consistent with our value of 0.097 for $N = 1024$. The best performance that they obtain with the PDA algorithm is an error rate of 0.074 which is approximately the same as the DDPPMF, DDPPSRE, KLT, KLTBB, KLTDPP, and LDB algorithms studied in this thesis with $N = 1024$. It therefore appears that adapted feature extraction and penalized discriminant analysis are equally valid ways of controlling the curse of dimensionality (see section 2.2.5) in pattern recognition problems.

Buckheit and Donoho [13] studied this dataset with their weighted discriminant pursuit (WDP) algorithm which was also studied in this thesis. They used $N = 1600$ and performed 25 trials to obtain an average error rate for each method they studied. They report an error rate of 0.166 for Fisher's LDA without any feature extraction which is completely inconsistent with our result and Hastie's result of ~ 0.9 for $N = 1000$. They also report that Hastie et al. obtained an error rate of 0.087 with their PDA algorithm which is also incorrect, since this was the error rate that Hastie reported for the LDA method without penalization. As noted above, their PDA method obtained an error rate of 0.074. They report obtaining error rates of 0.087 and 0.108 for the WDP algorithm with a wavelet packet and cosine packet dictionary

respectively. The error rate obtained in this thesis for WDP with the same wavelet packet dictionary and $N = 1024$ is 0.106, which is higher than their result but also plausible since they used a larger value of N. They do not report error rates for any other values of N which is why they fail to see the catastrophic failure of their method when $N \leq 512$.

Another point should be made regarding a comment that Buckheit and Donoho made in their paper. They claim that the KLT algorithm for extracting features suffers from the same problems that Fisher's method does and thus should produce very poor results when N is small and the dimensionality of the feature vector is high. This is obviously not the case since the KLT algorithm performed very well (better than DP and WDP) in this regime and actually outperformed every other feature extraction algorithm in the extreme when $N = 32$.

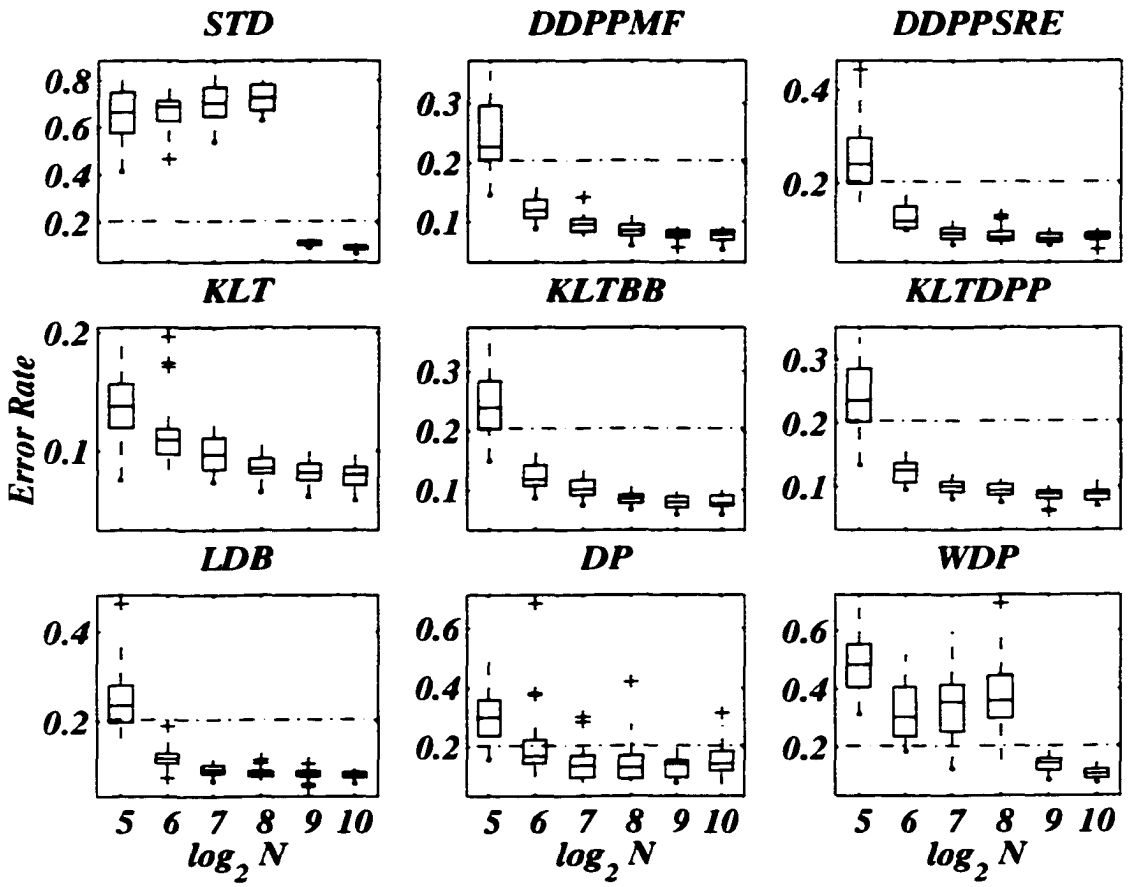

Figure 6.8. *Results for the Phoneme experiment. The variable N indicates the total number of frames that were used to train the feature extraction algorithms and classifier.*

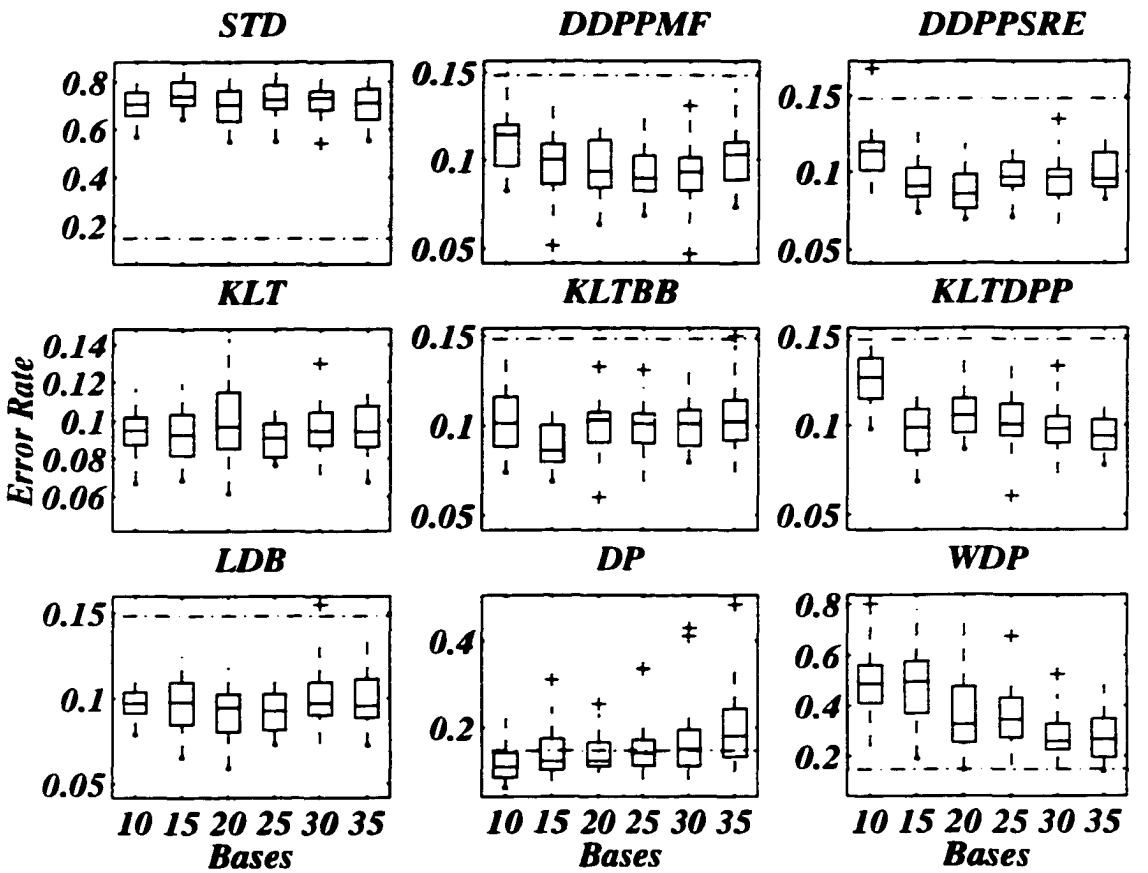

Figure 6.9. *Results for the Phoneme experiment. The variable Bases indicates the number of basis functions that were kept by the adapted feature extraction technique.*

Algorithm	32	64	128	256	512	1024
	0.748	0.712	0.769	0.782	0.122	0.101
STD	**0.663**	**0.687**	**0.705**	**0.725**	**0.114**	**0.097**
	0.577	0.628	0.649	0.671	0.106	0.089
	0.297	0.137	0.104	0.095	0.085	0.085
DDPPMF	**0.227**	**0.120**	**0.096**	**0.086**	**0.081**	**0.081**
	0.206	0.107	0.084	0.078	0.075	0.071
	0.297	0.150	0.104	0.097	0.093	0.094
DDPPSRE	**0.242**	**0.120**	**0.095**	**0.087**	**0.084**	**0.088**
	0.200	0.105	0.083	0.079	0.077	0.082
	0.156	0.118	0.111	0.094	0.089	0.086
KLT	**0.137**	**0.109**	**0.097**	**0.085**	**0.082**	**0.079**
	0.119	0.098	0.084	0.081	0.075	0.071
	0.285	0.143	0.118	0.092	0.088	0.091
KLTBB	**0.239**	**0.118**	**0.101**	**0.086**	**0.079**	**0.077**
	0.203	0.107	0.092	0.078	0.070	0.073
	0.285	0.138	0.108	0.104	0.094	0.093
KLTDPP	**0.236**	**0.127**	**0.101**	**0.094**	**0.089**	**0.088**
	0.202	0.107	0.092	0.087	0.082	0.078
	0.282	0.128	0.102	0.089	0.089	0.087
LDB	**0.236**	**0.117**	**0.092**	**0.082**	**0.082**	**0.082**
	0.198	0.107	0.083	0.078	0.078	0.075
	0.358	0.222	0.170	0.170	0.153	0.184
DP	**0.298**	**0.168**	**0.137**	**0.129**	**0.142**	**0.143**
	0.237	0.145	0.096	0.092	0.096	0.118
	0.553	0.407	0.414	0.444	0.159	0.122
WDP	**0.482**	**0.303**	**0.354**	**0.359**	**0.146**	**0.106**
	0.407	0.237	0.253	0.299	0.120	0.092

Table 6.5. *Results for sub-experiment N in the phoneme experiment. The center bold number gives the median error rate, the upper number gives the 75^{th} percentile, and the lower number gives the 25^{th} percentile.*

Algorithm	10	15	20	25	30	35
	0.755	0.798	0.763	0.786	0.761	0.768
STD	**0.705**	**0.740**	**0.702**	**0.727**	**0.730**	**0.711**
	0.658	0.702	0.634	0.686	0.682	0.643
	0.120	0.109	0.111	0.102	0.101	0.110
DDPPMF	**0.114**	**0.100**	**0.093**	**0.089**	**0.093**	**0.103**
	0.096	0.086	0.084	0.082	0.083	0.089
	0.120	0.103	0.098	0.107	0.102	0.113
DDPPSRE	**0.114**	**0.091**	**0.086**	**0.097**	**0.097**	**0.096**
	0.101	0.084	0.077	0.091	0.086	0.090
	0.102	0.103	0.115	0.099	0.104	0.107
KLT	**0.095**	**0.093**	**0.097**	**0.091**	**0.094**	**0.094**
	0.087	0.082	0.085	0.081	0.087	0.086
	0.116	0.100	0.107	0.106	0.108	0.114
KLTBB	**0.101**	**0.086**	**0.103**	**0.101**	**0.101**	**0.102**
	0.089	0.080	0.091	0.090	0.089	0.092
	0.138	0.109	0.115	0.112	0.105	0.103
KLTDPP	**0.127**	**0.099**	**0.105**	**0.100**	**0.098**	**0.094**
	0.115	0.086	0.096	0.094	0.090	0.087
	0.104	0.110	0.103	0.103	0.110	0.112
LDB	**0.097**	**0.098**	**0.094**	**0.093**	**0.097**	**0.096**
	0.091	0.085	0.080	0.081	0.090	0.089
	0.144	0.174	0.166	0.172	0.195	0.243
DP	**0.110**	**0.124**	**0.123**	**0.142**	**0.150**	**0.180**
	0.086	0.103	0.112	0.113	0.112	0.133
	0.562	0.577	0.478	0.433	0.328	0.348
WDP	**0.488**	**0.497**	**0.326**	**0.347**	**0.259**	**0.268**
	0.414	0.371	0.253	0.274	0.227	0.195

Table 6.6. *Results for sub-experiment Bases in the phoneme experiment. The center bold number gives the median error rate, the upper number gives the* 75^{th} *percentile, and the lower number gives the* 25^{th} *percentile.*

6.4 Conclusion

From the results in this chapter, the following conclusions can be drawn.

1. Adapted feature extraction techniques can reduce the error rate achievable using fixed 1/3 octave features by as much as 40% for noise monitoring data.
2. The SRE criterion appears to be slightly superior to the MF criterion for noise monitoring data but essentially equivalent for phoneme data. Since the MF criterion was found to be superior for the synthetic data studied in chapter 5, this suggest that the error rate achievable by a given criterion is strongly data dependent.
3. The DP algorithm performs poorly when the dataset has a combination of classes that are easy to discriminate between and classes that are hard to discrimate between since it keeps choosing features that separate the classes that are easy to separate (Buckheit and Donoho [13] observed this fact as well).
4. The last three conclusions made for the synthetic data in section 5.7 apply to the recorded data as well.

Chapter 7

Conclusion

7.1 Summary

This thesis developed the discriminant dictionary projection pursuit (DDPP) algorithm which was able to take advantage of the powerful mathematics of wavelet packet signal processing to efficiently extract useful features from sampled acoustic spectra for the purpose of discriminating between different classes of sounds. In a series of extensive classification experiments on real and synthetic data it was shown that the DDPP algorithm with an appropriate projection criterion was able to perform as well or better than i) traditional feature extraction techniques such as the Karhunen-Loève (KL) transform and ii) other wavelet packet feature extraction techniques such as Saito and Coifman's local discriminant bases [108, 109, 110] and Buckheit and Donoho's discriminant pursuit [13].

The DDPP algorithm (section 4.6) is a special case of the more general dictionary projection pursuit (DPP) algorithm (section 4.4) which in turn is an adaptation of the projection pursuit (PP) algorithm [61]. The only advantage of using the DPP algorithm over the PP algorithm is computational efficiency, but for high dimensional data, this can be significant, allowing the DPP algorithm to be used when the PP algorithm is simply too slow. It should be emphasized that the DPP algorithm is best suited for applications where the multi-dimensional feature vectors are samples from an underlying continuous signal. In cases where the elements of the feature vector have complicated relationships such as temperature, humidity, pressure, etc., there is no reason to expect that the wavelet packet basis functions would provide interesting or useful projections. However, the DPP algorithm could still be used cautiously to reduce the dimensionality of the feature space allowing the PP algorithm to be

applied more efficiently. This concept is discussed further in section 7.3.

The most traditional form of feature extraction uses the Karhunen-Loève (KL) transform. As discussed in section 4.7.4, there are some potential drawbacks to this method, but as exemplified in the experiments of chapters 5 and 6, this method still performs well for most feature extraction problems. For all the problems studied in this thesis, the dimensionality of the feature vector was ≤ 256 which made it possible to use the KL transform with reasonable computational efficiency. However, as discussed in section 4.7.4, the number of operations required to compute the KL transform is propotional to $O(NM^2 + M^3)$, where N is the number of signals in the ensemble and M is the dimensionality of the feature vector; increasing the dimensionality much past 256 made this technique unacceptably slow. The number of operations required for the DPP approximate KL transform developed in this thesis is proportional to $O((N + \hat{M})M \log M + \hat{M}^3)$, where N and M are defined above and $\hat{M}$ is the number of basis functions (*i.e.*, features) that are kept. Therefore, this algorithm scales very easily to higher dimensions, and as shown in the experiments of chapters 5 and 6 it performs nearly as well as the KL feature extraction method. These results provide strong support for using the DPP approximate KL transform in high dimensional spaces for feature extraction or simply as a means of dimensionality reduction.

When the projection criterion function of the DPP algorithm is a discriminant criterion function, then the algorithm is called discriminant dictionary projection pursuit (DDPP). In this form, the algorithm can be used to select spectral features for discriminating between different classes of sounds by using several example spectra from each class. This is similar to the KL methods discussed above except that by using a discriminant projection criterion it ensures that basis functions are chosen which separate the classes in a statistical sense rather than just choosing basis functions that have large variance. It was shown in section 4.6.2 that the basis functions that best separate phonemes are *different* from the basis functions that best separate noise monitoring sounds such as trains, planes and cars. This is perhaps not surprising, but it does emphasize that each sound recognition task is unique and thus it is important to choose features independently for each problem. It is also interesting that the DDPP algorithm found basis functions for the phoneme recognition

problem which are very similar to the basis functions used by the speech recognition community which were found by many years of trial and error.

Extensive experimentation was done in chapters 5 and 6 to compare the DDPP algorithm with two different discriminant criterion functions to six other feature extraction algorithms. The algorithms were evaluated based on classification results using a common classifier (Fisher's LDA) in a variety of different situations. The most consistent performer in these experiments was the DDPP algorithm with the modified Fisher criterion (both developed in this thesis). This is particularly impressive since the experiments were designed before the DDPP algorithm was implemented. However, excitement about this result should be contained since other algorithms also performed well in these experiments such as the KLT algorithm and Saito and Coifman's LDB algorithm. The main advantage of the the DDPP algorithm over the KLT algorithm is that it is computationally more efficient and it produces significantly better results when the discriminant information in a problem is highest in a low variance subspace (see section 5.5). The main advantage of the DDPP algorithm over the LDB algorithm is that DDPP can optimize criterion functions like the modified Fisher criterion which LDB cannot. This is significant since in many cases, the modified Fisher (MF) criterion seems to be superior to the symmetric relative entropy (SRE) criterion (*e.g.* sections 5.5 and 5.6). This is not a universal trend though since SRE seems to be a better criterion for the noise monitoring data in section 6.2.

One of the goals of this thesis was to show that the classification performance of noise monitoring data can be improved by using adapted feature extraction techniques rather than fixed 1/3 octave features which is currently the standard in the field. This objective was achieved since the median error rate obtained using adapted feature extraction were up to 40 % lower than the 1/3 octave error rates (section 6.2).

In summary, the dictionary projection pursuit (DPP) algorithm developed in this thesis, and in particular the discriminant version of it (DDPP) performs very well as a feature extraction tool for sound recognition tasks. Using wavelet packet signal processing techniques allowed this algorithm to be implemented very efficiently.

7.2 Original Contribution

The original contributions in this thesis can be succintly stated as follows:

1. The dictionary projection pursuit algorithm (DPP in section 4.4) and its discriminant version (DDPP in section 4.6).
2. The modified Fisher discriminant criterion (section 4.6.1.2).
3. The signal model (section 5.2).
4. The Monte Carlo method for estimating the Bayes error rate (section 5.2.5).
5. The synthetic data classification results (chapter 5).
6. The recorded data classification results (chapter 6).

7.3 Future Work

The most exciting part about this thesis and the development of the DPP algorithm is the number of possible applications that are yet to be investigated. This section describes just a few of the possibilities.

1. Probably the most obvious next step is to apply the DDPP algorithm to classification problems with higher dimensional data such as images or multiframe acoustic data.
2. There are many other application areas that could also take advantage of the DDPP algorithm (*e.g.* classifying chemical or astronomical spectra).
3. There are many other possible applications for the DPP algorithm other than feature extraction, many of which are described by Huber [61]. One of the most interesting would be to use DPP as a fast and automatic method of finding outliers in a dataset of sampled waveforms (*i.e.*, finding projections where most of the waveforms are clustered nicely but a few waveforms have radically different values). These outliers are usually caused by measurement errors of some sort and often cause major problems with automatic data processing systems. Being able to eliminate outliers in data automatically with the DPP algorithm would be very advantageous.

4. It is well known that the wavelet packet transform is not shift invariant. That is, if the input signal is shifted by a few pixels, the resulting projection coefficients can be significantly different. The DDPP algorithm could thus be improved by using a shift invariant form of the wavelet packet transform [32]. This simply amounts to having more basis functions to choose from which are slightly shifted versions of the current basis functions at the cost of higher computational complexity.
5. Finally, another method of possibly improving the DPP algorithm is to use it as a preprocessor for the PP algorithm, similar to the way that the DPP algorithm was used as a preprocessor for the KLT algorithm in section 4.7.6. In this way, DPP is used as a coarse method of optimizing the projection criterion index and PP is used to fine tune the result.

Obviously, there is a significant amount of future work to be done with this algorithm and hopefully the results will be as successful as the results obtained in this thesis.

Bibliography

[1] A Antoniou. *Digital Filters: Analysis, Design, and Applications.* McGraw-Hill Book Company, New York, 1993.

[2] G. Arfken. *Mathematical Methods for Physicists (Third Edition).* Academic Press, Inc., 1985.

[3] T. Ashiya and M. Nakagawa. A proposal of a recognition system for environmental sound. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E76-A(10):1858–1860, October 1993.

[4] A. Averbuch, L. Bahl, R. Bakis, P. Brown, A. Cole, S. Daggett, G. Das, K. Davies, S. De Gennaro, P. de Souza, E. Epstein, D. Fraleigh, F. Jelinek, S. Katz, B. Lewis, R. Mercer, A. Nadas, D. Nahamoo, M. Picheny, G. Shichman, and P. Spinelli. An IBM-PC based large-vocabulary isolated utterance speech recognizer. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 53–56, April 1986. Tokyo, Japan.

[5] R.E. Bellman. *Adaptive Control Processes.* Princeton University Press, 1961.

[6] A.H. Benade. *Fundamentals of Musical Acoustics (2nd Revised Edition).* Dover Publications Inc., 31 East 2nd St., Mineola, N.Y., 11501, 1990.

[7] R.N. Bracewell. *The Fourier Transform and Its Applications 2nd Ed.* McGraw-Hill Book Company, New York, 1986.

[8] A.S. Bregman. *Auditory Scene Analysis: the perceptual organization of sound.* MIT Press, Cambridge, MA, 1990.

[9] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees.* Wadsworth International Group, 1984.

[10] G.J Brown. *Computational Auditory Scene Analysis: A Representational Approach.* PhD thesis, University of Sheffield, 1992.

[11] R.G. Brown and P.Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering.* John Wiley & Sons, New York, 1997.

[12] J.G. Bryan. The generalized discriminant function: Mathematical foundations and computational routine. *Harvard Educational Review*, 21:90–95, 1951.

[13] J. Buckheit and D. Donoho. Improved linear discrimination using time-frequency dictionaries. In *Proceedings of SPIE Wavelet Applications in Signal and Image Processing III Vol 2569*, pages 540–551, July 1995.

[14] S. Chen and D.L. Donoho. Examples of basis pursuit. In A.F. Laine, M.A. Unser, and M.V. Wickerhauser, editors, *Proceedings of SPIE: Wavelet Applications in Signal and Image Processing III Volume 2569*, pages 564–574, July 1995.

[15] S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[16] V. Cherkassky and F. Mulier. *Learning From Data: Concepts, Theory, and Methods.* John Wiley and Sons, Inc., 1998.

[17] C.K. Chui. *An Introduction to Wavelets, Wavelet Analysis and its Applications*, volume 1. Academic Press, New York, 1992.

[18] A. Cohen, I. Daubechies, and P. Vial. Wavelets on the interval and fast wavelet transforms. *Applied and Computational Harmonic Analysis*, 1:54–81, December 1993.

[19] A. Cohen and J. Kovačević. Wavelets: The mathematical background. *Proceedings of the IEEE*, 84(4):514–522, April 1996.

[20] R.P. Cohn. Robust Voiced/Unvoiced speech classification using a neural net. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 437–440, May 1991.

[21] R. Coifman and V. Wickerhauser. Entropy-based algorithms for best-basis selection. *IEEE Transactions on Information Theory*, 38:713–718, 1992.

[22] M. Cooke. *Modelling Auditory Processing and Organisation.* PhD thesis, University of Sheffield, 1993.

[23] J.W. Cooley and J.W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematical Computations*, 19:297–301, 1965.

[24] S. Coren, L.M. Ward, and J.T. Enns. *Sensation and Perception.* Harcourt Brace College Publishers, 4th edition, 1994.

[25] C. Couvreur. *Environmental Sound Recognition: A Statistical Approach.* PhD thesis, Faculté Polytechnique de Mons, 1997.

[26] C. Couvreur. Personal communication, 1998.

[27] C. Couvreur and Y. Bresler. A statistical pattern recognition framework for noise recognition in an intelligent noise monitoring system. In *Proceedings EURO-NOISE '95*, volume 3, pages 1007–1012, Lyon, France, Mar. 1995.

[28] B.V. Dasarathy. *Nearest Neighbour (NN) Norms: NN Pattern Classification Techniques.* IEEE Computer Society Press, 1991.

[29] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, XLI:901–996, 1988.

[30] I. Daubechies. *Ten Lectures on Wavelets.* Philadelphia: Society of Industrial and Applied Mathematics, 1992.

[31] I. Daubechies. Where do wavelets come from?— a personal point of view. *Proceedings of the IEEE*, 84(4):510–513, April 1996.

[32] S. Del Marco and J. Weiss. Improved transient signal detection using a wavepacket-based detector with an extended translation-invariant wavelet transform. *IEEE Transactions on Signal Processing*, 45(4), April 1997.

[33] M. Desai and D. Shazeer. Acoustic transient analysis using wavelet decomposition. In *Proceedings of the IEEE conference on neural networks for ocean engineering*, Aug. 1991.

[34] P.A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach.* Prentice-Hall International, Inc., 1982.

[35] G.R. Doddington. Speaker recognition - identifying people by their voices. *Proceedings of the IEEE*, 73(11):1651–1664, November 1985.

[36] D.L. Donoho. De-noising by soft thresholding. *IEEE Transactions on Information Theory*, 41:613–627, 1995.

[37] E. Dorken, E.E. Milios, and S.H. Nawab. Knowledge-based signal processing applications. In A.V. Oppenheim and S.H. Nawab, editors, *Symbolic and Knowledge-Based Signal Processing*, chapter 9, pages 303–330. Prentice Hall, Englewood-CLiff, NJ, 1992.

[38] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons, Inc., 1973.

[39] D.P.W. Ellis. *Prediction-Driven Computational Auditory Scene Analysis.* PhD thesis, Massachusetts Institute of Technology, 1996.

[40] D. Estaban and C. Galand. Application of quadrature mirror filters to split-band voice coding schemes. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 191–195, 1977.

[41] F.A. Everest. *The Master Handbook of Acoustics.* TAB Books, 3rd edition, 1994.

[42] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[43] B. Flury. *A First Course in Multivariate Statistics.* Springer-Verlag, 1997.

[44] J.E. Freund. *Mathematical Statistics.* Prentice-Hall, Inc., 5th edition, 1992.

[45] J.H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175, March 1989. Theory and Methods.

[46] J.H. Friedman. An overview of predictive learning and function approximation. In V. Cherkassky, J.H. Friedman, and H. Wechsler, editors, *From Statistics to*

Neural Networks: Theory and Pattern Recognition Applications, pages 1–61. Springer-Verlag, 1994.

[47] J.H. Friedman and W. Stuetzle. Projection pursuit regression. *Journal of the American Statistical Associtation*, 76(376):817–823, December 1981. Theory and Methods Section.

[48] J.H. Friedman and J.W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, c-23(9):881–890, September 1974.

[49] K.S. Fu. *Syntactic Pattern Recognition and Applications.* Prentice-Hall International, Inc., 1982.

[50] W.A. Gardner. *Statistical Spectral analysis.* Prentice-Hall Inc., Englewood Cliffs, New Jersey 07632, 1988.

[51] A. Gersho and R.M. Gray. *Vector Quantization and Signal Compression.* Kluwer Academic, 1992.

[52] R. Goldhor. Recognition of environmental sounds. In *Proceedings of the IEEE International Conference on Acoustics, Speech and S Ignal Processing*, pages I–149–I–152, 1993.

[53] G.H. Golub and C.F. Van Loan. *Matrix Computations.* The John Hopkins University Press, 2nd edition, 1989.

[54] P. Goupillaud, A. Grossman, and J. Morlet. Cycle-octave and related transforms in seismic signal analysis. *Geoexploration*, 23:85–102, 1984.

[55] A. Graps. An introduction to wavelets. *IEEE Computational Sciences and Engineering*, 2(2):50–61, Summer 1995.

[56] A. Grossmann and J. Morlet. Decomposition of hardy functions into square integrable wavelets of constant shape. *SIAM Journal of Mathematical Analysis*, 15(4):723–736, 1984.

[57] D.E. Hall. *Musical Acoustics: An Introduction.* Wadsworth Publishing Company, 1980.

[58] T. Hastie, A. Buja, and R. Tibshirani. Penalized discriminant analysis. *Annals of Statistics*, 23(1):73–102, February 1995.

[59] W. Hess. *Pitch Determination of Speech Signals.* Springer-Verlag, 1983.

[60] H. Hotelling. Analysis of a complex of statistical variables into principle components. *Journal of Educational Psychology*, pages 498–520, 1933.

[61] P.J. Huber. Projection pursuit (with discussion). *Annals of Statistics*, 13(2):435–525, 1985.

[62] Q.Q. Huynh, L.N. Cooper, N. Intrator, and H. Shouval. Classification of underwater mammals using feature extraction based on time-frequency analysis and BCM theory. *IEEE Transactions on Signal Processing*, 46(5):1202–1207, 1998.

[63] N. Intrator and L.N. Cooper. Objective function formulation of the BCM theory of visual cortical plasticity: Statistical connections, stability considerations. *Neural Networks*, 5:3–17, 1992.

[64] M.C. Jones and R. Sibson. What is projection pursuit? (with discussion). *Journal of the Royal Statistical Association*, 150(1):1–36, 1987.

[65] Pierce J.R. *The Science of Musical Sound (Revised Edition)*. Scientific American Books, 1992.

[66] A. Khotanzad, J.H. Lu, and M.D. Srinath. Target detection using a neural network based passive sonar system. In *Proceedings of the IEEE International Joint Conference on Neural Networks*. IEEE, 1989.

[67] F. Klassner. *Data Reprocessing in Signal Understanding Systems*. PhD thesis, University of Massachusetts Amherst, 1996.

[68] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, 2nd extended edition, 1997.

[69] J.B. Kruskal. Toward a practical method which helps unccover the structure of a set of multivariate observations by finding the linear transformation which optimizes a new 'index of condensation'. In R.C. Milton and J.A. Nelder, editors, *Statistical Computation*, pages 427–440, New York, 1969. Academic Press.

[70] S. Kullback. *Information Theory and Statistics*. John Wiley, New York, 1959.

[71] R.E. Learned and A.S. Willsky. A wavelet packet approach to transient signal classification. *Applied and Computational Harmonic Analysis*, 2:265, 1995.

[72] K. Lee. *Automatic Speech Recognition: The Development of the SPHINX System*. Kluwer Academics Publishers, 1989.

[73] V. Lesser, S.H. Nawab, I. Gallastegi, and F. Klassner. Ipus: an architecture for integrated signal processing and signal interpretation in complex environments. In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93)*, Jul. 1993.

[74] V. Lesser, S.H. Nawab, and F. Klassner. IPUS: An architecture for the integrated processing and understanding of signals. *Artificial Intelligence Journal*, 77(1), Aug. 1995.

[75] R.P. Lippman. Pattern classification using neural networks. *IEEE Communications Magazine*, pages 47–64, November 1989.

[76] S. Mallat. Multifrequency channel decomposition of images and wavelet models. *IEEE Transactions on ASSP*, pages 2091–2110, Dec 1989.

[77] S. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.

[78] S. Mallat and W.L. Hwang. Singularity detection and processing with wavelets. *IEEE Trans. on Information Theory*, 38:617–643, 1992.

[79] S.G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, December 1993.

[80] R.J. Mammone, X. Zhang, and R.P. Ramachandran. Robust speaker recognition: A feature-based approach. *IEEE Signal Processing Magazine*, pages 58–71, September 1996.

[81] D. Marr. *Vision.* W.H. Freeman and Company, San Francisco, 1982.

[82] G. McLachlan. *Discriminant Analysis and Statistical Pattern Recogntion.* John Wiley and Sons, Inc., 1992.

[83] D.K. Mellinger. *Event Formation and Separation in Musical Sound.* PhD thesis, Stanford University, 1991.

[84] Y. Meyer. *Wavelets: Algorithms and Applications.* SIAM, Philedelphia, 1993.

[85] Y. Meyer. *Wavelets and Operators.* Cambridge Studies in Advanced Mathematics 37. Cambridge University Press, New York, 1993. Translation from the French version, Paris, 1990, by D.H. Salinger.

[86] J. Moody and C.J. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1:281–294, 1989.

[87] J. Morlet, G. Arens, E. Fourgeau, and D. Giard. Wave propogation and sampling theory - part i: Complex signal and scattering in multilayered media. *Geophysics*, 47(2):203–221, 1982.

[88] J. Morlet, G. Arens, E. Fourgeau, and D. Giard. Wave propogation and sampling theory - part 2: Sampling theory and complex waves. *Geophysics*, 47(2):222–236, 1982.

[89] P. Moukas, J. Simson, and L. Norton-Wayne. Automatic identification of noise pollution sources. *IEEE Transactions on Systems, Man and Cybernetics*, 12(5):622–634, Sept. 1982.

[90] S.H. Nawab and V. Lesser. Integrated processing and understanding of signals. In A.V. Oppenheim and S.H. Nawab, editors, *Symbolic and Knowledge-Based Signal Processing*, chapter 7, pages 286–302. Prentice-Hall, Englewood-Cliff, NJ, 1992.

[91] E. Oja. *Subspace Methods of Pattern Recognition.* Pattern Recognition and Image Processing Series. Research Studies Press Ltd. John Wiley and Sons Inc., 1983.

[92] H.F. Olson. *Music, Physics, and Engineering.* Dover Publications, Inc., 2nd edition, 1967.

[93] A.V. Oppenheim and R.W. Schafer. *Digital Signal Processing.* Prentice-Hall Inc, Englewood Cliffs, New Jersey, 1975.

[94] A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill Book Company, New York, 1984.

[95] M.J. Paradie and S.H. Nawab. The classification of ringing sounds. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Prcessing (ICASSP)*, pages 2435–2438, Albuquerque, NM, April 1990.

[96] R.D. Patterson and J. Holdsworth. A functional model of neural activity patterns and auditory images. In W.A. Ainsworth, editor, *Advances in Speech, Hearing and Language Processing Vol. 3.* JAI Press, 1990.

[97] R.D. Patterson and B.C.J. Moore. Auditory filters and excitation patterns as representations of frequency resolution. In B.C.J. Moore, editor, *Frequency Selectivity in Hearing.* Academic, 1986.

[98] J. Picone. Continuous speech recognition using hidden markov models. *IEEE Acoustics, Speech and Signal Processing Magazine*, pages 26–41, July 1990.

[99] B. Pinkowski. Lpc spectral moments for clustering acoustic transients. *IEEE Transactions on Speech and Audio Processing*, 1(3), Jul. 1993.

[100] B. Pinkowski. Robust fourier descriptors for characterizing amplitude modulated waveform shapes. *Journal of the Acoustical Society of America*, 95(6):3419–3423, June 1994.

[101] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C.* Cambridge University Press, 40 West 20th Street, New York, NY 10011 -4211, USA, 1992.

[102] L. Rabiner and B.H. Juang, editors. *Fundamentals of Speech Recognition.* Prentice-Hall, Englewood-Cliff, NJ, 1993.

[103] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[104] L.R. Rabiner and B.H. Juang. An introduction to hidden markov models. *IEEE Signal Processing Magazine*, 3(1):4–16, 1986.

[105] M.D. Richard and R.P. Lippman. Neural network classifiers estimate bayesian *a posteriori* probabilities. *Neural Computation*, 3:461–483, 1991.

[106] J. Rissanen. Modeling by shortest description length. *Automatica*, 14:465–471, 1978.

[107] D.F. Rosenthal and H.G. Okuno, editors. *Computational Auditory Scene Analysis*, 10 Industrial Ave, Mahwah, New Jersey 07430, 1998. Lawrence Erlbaum Associates.

[108] N. Saito. *Local Feature Extraction and It's Applications Using a Library of Bases.* PhD thesis, Department of Mathematics, Yale University, New Haven CT, USA, December 1994.

[109] N. Saito and R. Coifman. Local discriminant bases and their applications. *Journal of Mathematical Imaging and Vision*, 5(4):337–358, 1995.

[110] N. Saito and R. Coifman. Improved local discriminant bases using empirical probability density estimation. In *American Statistical Association's Proceedings of Statistcal Computing*, 1996.

[111] E. Scheirer and M. Slaney. Construction and evaluation of a robust multifeature music/speech discriminator. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 1331–1334. IEEE Press, April 1997.

[112] E.D. Scheirer. *Using Musical Knowledge to Extract Expressive Performance Information from Audio Recordings*, chapter 24, pages 361–380. Lawrence Erlbaum Associates Publishers, 1998.

[113] J. Schürmann. *Pattern Classification: A Unified View of Statistical and Neural Approaches.* John Wiley & Sons, Inc., New York, 1996.

[114] J.M. Shapiro. Embedded image coding using zerotrees of the wavelet coefficients. *IEEE Transactions on Signal Processing*, 41:3445–3462, 1993.

[115] M. Slaney. An efficient implementation of the patterson-holdsworth auditory filter bank. Technical Report No. 35, Apple Computer Co., 1993.

[116] M.J.T. Smith and T.P. Barnwell III. A procedure for designing exact reconstruction filter banks for tree-structured sub-band coders. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, San Diego, CA, March 1984.

[117] M.J.T. Smith and T.P. Barnwell III. Exact reconstruction for tree-structured sub-band coders. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34(3):431–441, June 1986.

[118] G. Strang. *Linear Algebra and its Applications.* Academic Press Inc., 111 Fifth Ave, New York, NY 10003, 1976.

[119] G. Strang. Wavelets. *American Scientist*, 82:250–255, May-June 1994.

[120] G. Strang and K. Borre. *Linear Algebra, Geodesy and GPS.* Wellesley-Cambridge Press, 1997.

[121] G. Strang and T. Nguyen. *Wavelets and Filter Banks.* Wellesley-Cambridge Press, Box 812060 Wellesley MA 02181 USA, 1996.

[122] D.W. Thomas. Vehicle sounds and recognition (chapter 13). In B.G. Batchelor, editor, *Pattern Recognition: Ideas in Practise*, pages 333–361, London, 1978. Plenum Press.

[123] J.T. Tou and R.C. Gonzalez. *Pattern Recognition Principles.* Addison-Wesley Publishing Company, 1974.

[124] S.R. Turajlić and Z.M. Šarić. Sequential speech segmentation based on the spectral ARMA transition measure. *Circuits Systems Signal Process*, 15(1):71–92, 1996.

[125] P.P. Vaidyanathan. Multirate digital filters, filter banks, polyphase networks, and applications: A tutorial. *Proceedings of the IEEE*, 78:56–93, January 1990.

[126] P.P. Vaidyanathan. *Multirate Systems and Filter Banks.* Prentice-Hall Inc., Englewood Cliffs, New Jersey 07632, 1993.

[127] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer Verlag, New York, 1995.

[128] M. Vetterli and J. Kovačević. *Wavelets and Subband Coding.* Prentice-Hall Inc., Englewood Cliffs, New Jersey 07632, 1995.

[129] D.S. Watkins. *Fundamentals of Matrix Computations.* John Wiley and Sons, 1991.

[130] M.V. Wickerhauser. *Adapted Wavelet Analysis from Theory to Software.* A.K. Peters, Ltd, 1994.

[131] J.M. Winograd and S.H. Nawab. A c++ software environment for the development of embedded signal processing systems. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2715–2718, Detroit, MI, May 1995.

[132] R.E. Wohlford. A comparison of four techniques for automatic speaker recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Singal Processing*, pages 908–911, April 1980.

[133] E. Wold, T. Blum, D. Keislar, and J. Wheaton. Content-based classification, search, and retrieval of audio. *IEEE Multimedia*, 3(3):27–36, 1996.

[134] E. Wold, T. Blum, D. Keislar, and J. Wheaton. Classification, search and retrieval of audio. In *CRC Handbook of Multimedia Computing*, 1999.

[135] J.P. Woodard. Modeling and classification of natural sounds by product code

hidden markov models. *IEEE Transactions on Signal Processing*, 40(7):1833–1835, July 1992.