Time-Frequency Features
For Speech Recognition

James G. Droppo III

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Washington

2000

Program Authorized to Offer Degree: Electrical Engineering

University of Washington

Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

James G. Droppo III

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of Supervisory Committee:

_____

Les E. Atlas

Reading Committee:

_____

X. D. Huang

_____

John Sahr

Date: _____

In presenting this dissertation in partial fulfillment of the requirements for the Doctorial degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to Bell and Howell Information Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature_____

Date_____

University of Washington

Abstract

Time-Frequency Features For Speech Recognition

by James G. Droppo III

Chair of Supervisory Committee

Professor Les E. Atlas
Electrical Engineering

Conventional speaker independent, continuous speech recognition systems are built upon assumptions that are, in general, not met. This dissertation focuses on one deficiency in particular, that the non-stationary speech signal is modeled as a single series of stationary spectral estimates.

Time-frequency representations (TFRs) have the potential to be powerful features for nonstationary signals. Whereas short-term spectral estimates must make implicit time and frequency resolution tradeoffs, a single TFR simultaneously contains both short-term and long-term spectral estimates. Unfortunately, the proper way to harness this power is still a matter of debate.

This dissertation proposes a class dependent time-frequency feature for speech recognition. The feature is automatically derived from time-frequency representations of speech signals by maximizing the discriminability within classes. A two-stage speech recognition system incorporating these representaions achieves 1.6% error rate. This is 39% lower than the best published result for the chosen task.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

The author wishes to express sincere appreciation to his advisor Les Atlas, who provided encouragement and insight, and to Microsoft Research, who provided raw computational power and access to a wealth of knowledgeable people on speech recognition. Additionally, acknowledgment should be made to the other members of the ISDL lab, including Siva Narayanan, Jack McLaughlin, Lane Owsley, Brad Gillespie, John Keane, and Mike Dougherty. Without their encouragement, insight, and occasional distractions, work on this dissertation would have most certainly stalled long ago.

DEDICATION

This dissertation is dedicated to my family, who understand.

Chapter 1

INTRODUCTION

Conventional speaker-independent, continuous speech recognition systems are built upon assumptions that are, in general, not met. This dissertation focuses on one deficiency in particular, that the non-stationary speech signal is modeled as a single series of stationary spectral estimates.

Time-frequency representations (TFR) have the potential to be powerful features for nonstationary signals. Whereas short-term spectral estimates must make implicit time and frequency resolution tradeoffs, a single TFR simultaneously contains both short-term and long-term spectral estimates. Unfortunately, the proper way to harness this power is still a matter of debate.

This dissertation proposes a class dependent time-frequency feature for speech recognition. The feature is automatically derived from time-frequency representations of speech signals by maximizing the discriminability within classes. It can be used to supplement the conventional spectral features in a speech recognition system to improve discrimination. For demonstration purposes, the technique is applied to a state-of-the-art isolated alphabet recognition system. By constructing a two-stage recognition system and using the time-frequency feature to refine classification on confusable sets, 34% error reduction is obtained over the best baseline by mel-frequency cepstral features. Overall, the two-stage system achieves 1.6% error rate. This is 39% lower than the best published result for this task.

1.1 Overview

The goal of this dissertation is to show that it is feasible to develop time-frequency representations that can be incorporated into a continuous speech recognition system that outperforms a baseline system.

This document can be divided into three distinct sections. The first section consists of Chapter 1, and serves as an introduction to the relevant background information in speech recognition and a motivation for the other two sections.

Chapter 2 through Chapter 5 form the second section of this dissertation. They contain the theoretical framework that was built for the task of using discrete time-frequency representations in continuous speech recognition. While these chapters were crafted with speech recognition in mind, the results are general enough to apply to diverse classification tasks.

The genesis of our time-frequency representations is to be found in the area of operator theory, which we use in Chapter 2 to establish a direct link between a discrete signal and its corresponding discrete time-frequency representation. We see that many similarities, but also some important differences, exist between the results of the continuous-time operator approach and our discrete one. The differences between the continuous representations and discrete ones transcend the simple sampling relationship that has often been assumed.

From this work, we obtain a concise, matrix-based expression for a discrete time-frequency representation which is simply the product of the kernel with another matrix. This simple expression opens up the possibility to optimize the kernel in a number of ways. We focus, of course, on optimizations most suitable for classification, and ultimately wind up with the class dependent kernel.

The remainder of the theoretical section is devoted to overcoming the high dimensionality of these time-frequency representations, which is the biggest obstacle to modeling in a practical system.

The statistical properties of this new discrete formulation are explored in Chapter 3. Although the discrete time-frequency representations exist in a very high dimensional space, it is possible to compute full rank covariance matrices with a small amount of training data. A good estimate of the covariance can be used to either model the data directly, or to serve as a starting point for reducing the feature's dimensionality.

Chapter 4 introduces a set of criteria for the design of kernels (generating functions) for discrete time-frequency representations. These criteria aim only to produce kernels (and thus, TFRs) which will enable more accurate classification. We refer to these kernels, which are optimized to discriminate among several classes of signals, as *signal class dependent kernels*, or simply *class dependent kernels*.

The last chapter in the theoretical section is Chapter 5. Although the discrete time-frequency representations exist in a high dimensional space, this might be just an illusion. The features themselves are a continuous function of the signal values in the original discrete time signal. If we start with N signal samples, the expectation is that the true feature exists as a continuous N dimensional surface embedded in the signal space. Chapter 5 explores this concept and its ramifications.

To conclude the dissertation, Chapter 6 presents practical results that show the utility of the new discrete time-frequency features coupled with appropriate class dependent kernels. Results are presented for simulated passive sonar transient signals, isolated phone recognition, and isolated alphabet recognition.

1.2 Background

Speech recognition systems typically rely on one kind of spectral feature, with one set of parameters, for recognizing all the sounds generated by the target language. These sounds include vowels, which can be stationary for over 100 ms, and stops, which last on the order of 10 ms, the timing of which can be important for recognition.

In the context of this dissertation, sounds in the English language can be categorized into classes according to the type of sound. One can make broad divisions such as voiced and unvoiced sound, or become more specific, such as front vowels, back vowels, semivowels, and so on.

Regardless of the exact categorization used, even a cursory examination of speech data shows that different sound classes have very different signal structures. In particular, they vary in their duration and relative degree of stationarity. Despite this variation, current speech recognition systems use a single model in which signals are assumed to be piecewise stationary. This assumtion is met to a varying degree, depending on the signal class. Stops, such as /p/ and /t/, have definite stationary states, but semivowels and glides change their spectrum continuously over time.

The most widely used feature representation for speech recognition is mel-frequency cepstral coefficients. These MFCC are cepstral features based on mel-scale filter banks. Even though MFCC are very good features, they have the same fixed time and frequency domain resolution for all sound classes, which may not be optimal. Time-frequency representations, on the other hand, contain simultaneous short-term and long-term information. Class-dependency makes it possible to optimize TFR features for a particular class, i.e. sound type, to maximize its discriminability.

These features can be used to enhance the performance of the traditional speech recognition systems based on MFCC features.

Section 1.3 provides an overview of how continuous speech recognition systems are built today. It includes a summary of the basic components used to construct the system, a review of modern training techniques, along with commentary about the strengths and weaknesses of the various components.

Section 1.4 introduces relevant research into the construction of more sophisticated hidden Markov model structures. Although these structures were in general developed to build noise robust recognition systems, we use them to improve recognition performance even in the absence of noise.

1.3 Continuous Speech Recognition

At the most general level, a continuous speech recognition system produces a transcription $\mathbf{W} = \{w_0, w_1, \ldots, w_{K-1}\}$ from a sequence of acoustic feature vectors (frames), $\mathbf{X} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{N-1}\}$. The transcription symbols could be words, or sub-word units such as phones, diphones, or triphones, depending on the design of the recognition system. For the purposes of this document, the symbols are assumed to belong to the set of phones listed in Table 1.1.

The conventional approach is to define one stochastic model, capable of producing acoustic features, for every symbol in the vocabulary. The model for an entire transcription is built by concatenating the appropriate models for each symbol, as shown in Figure 1.1. In this figure, each labeled circle represents the model for the corresponding phone.

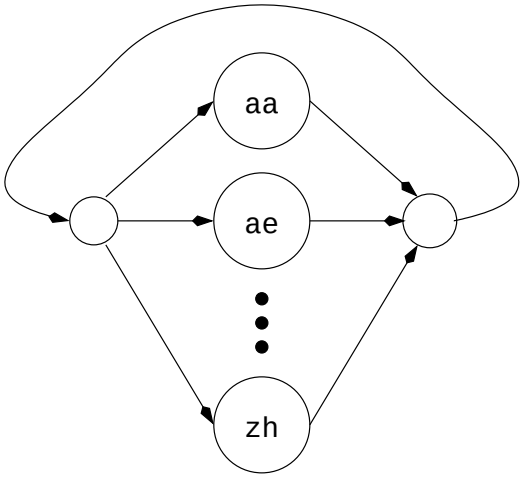

Figure 1.1: Transcription production model

This system is then used to discover which sequence of symbols is most likely to have produced the observed signal. The decoder with the least probability of error, the maximum *a posteriori* (MAP) decoder, is given by

Table 1.1: A set of English phonemes

Class	Symbol	Example
Stops	b	bee
	d	day
	g	gay
	p	pea
	t	tea
	k	key
Affricates	jh	joke
	ch	choke
Fricatives	s	sea
	sh	she
	z	zone
	zh	azure
	f	fin
	th	thin
	v	van
	dh	then
Semivowels and Glides	l	lay
	r	ray
	w	way
	y	yacht
	hh	hay
	el	bottle

Class	Symbol	Example
Nasals	m	mom
	n	noon
	ng	sing
	em	bottom
	en	button
Vowels	iy	beet
	ih	bit
	eh	bet
	ey	bait
	ae	bat
	aa	bott
	aw	bout
	ay	bite
	ah	but
	ao	bought
	oy	boy
	ow	boat
	uh	book
	uw	boot
	er	bird
	ax	about
	ix	debit
	axr	butter

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} P(\mathbf{W}|\mathbf{X}). \tag{1.1}$$

Computation of the *a posteriori* probability $P(\mathbf{W}|\mathbf{X})$ proceeds by an application of Bayes' rule, which rewrites this probability in terms of the conditional probability $P(\mathbf{X}|\mathbf{W})$, the probability that the observation sequence was produced from the the symbol sequence $\mathbf{W}$.

$$P(\mathbf{W}|\mathbf{X}) = \frac{P(\mathbf{X}|\mathbf{W})P(\mathbf{W})}{P(\mathbf{X})} \tag{1.2}$$

The term $P(\mathbf{W})$ is the probability that the correct transcription consists of the sequence of symbols $\mathbf{W}$. This probability is usually provided by a language model (Section 1.3.3), which defines the likely symbol sequences.

Since the signal representation remains constant for all frames, the term $P(\mathbf{X})$ is constant for a given observation sequence. It can safely be ignored when Equations 1.1 and 1.2 are combined for recognition.

To find $\hat{\mathbf{W}}$ in Equation 1.1, the recognition system must jointly optimize the symbol sequence and segmentation of the signal $\mathbf{X}$.

1.3.1 Hidden Markov Models

Hidden Markov models (HMMs) are a probabilistic tool that describe piecewise stationary signals, analogous to the way probability distribution functions describe stationary signals.

Although it can be argued that speech signals are not actually piecewise stationary, HMMs have so many other desirable qualities that they are hard to outperform in speech recognition systems.

Hidden Markov models are popular in speech recognition systems because they are simple enough to implement in a real time system, yet complex enough to capture the basic non-stationary structure of speech. Because their behavior can be described

with simple formulas, the full power of mathematics and probability theory can be brought to bear on the speech recognition problem. The definitive tutorial on the basic HMM formulations can be found in [43].

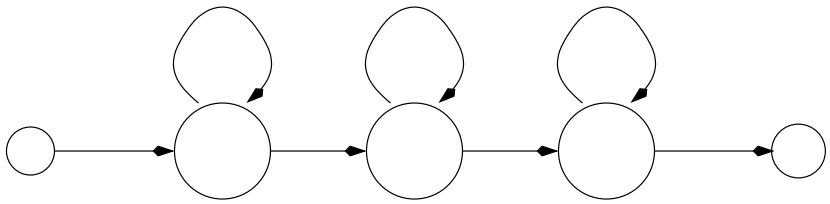

Figure 1.2: Conventional forward hidden Markov model

Figure 1.2 is a graph of a conventional, three state, forward hidden Markov model.

The three large circles correspond to states that the model can take on. While occupying a given state, the model takes on the properties of a stationary stochastic process. As such, each state is described by its probability distribution function (PDF) $f(x)$. This PDF can be modeled in one of several ways, including Gaussian mixture models (GMM), and discrete vector quantization (VQ) codebooks.

The two small circles at the beginning and end of the graph represent the entrance and exit states of the model. The model does not produce any output while in these states, their purpose is to make it more convenient to concatenate several models.

All of the arrows represent allowed transitions between states. At any discrete time index, the model occupies only one state and produces only one output. At each time increment, the model can follow one of the allowed transitions, according to a probability associated with each transition.

Individual HMMs can be chained together quite easily, and the search for the best transcription over all segmentations can be performed in one step. Individual symbol models are unified in a larger HMM structure, such as Figure 1.1, with transitions between the individual HMMs governed by the language model. Then, the Viterbi algorithm is used to find the single state sequence path with the highest probability.

Expectation-Maximization Training

Given an initial model and an observation sequence, a new HMM can be found that is more likely to have produced the observation sequence. This type of training, known as expectation-maximization (EM) training, is based on the classic work of Baum. Although it is quite popular, EM training is not optimal in terms of reducing the error rate for the system as a whole. There is no reason to believe that maximizing the probability of the data given the model will achieve the lowest possible error rate.

Discriminative Training

Discriminative training was introduced to correct some perceived deficiencies with EM training. In particular, it is not clear that increasing the *a posteriori* probability of the training data (Equation 1.2) necessarily reduces the error rate, even though it is the most common practical cost function assumed. Although it has been shown ([37]) that this will hold under certain assumptions, these assumptions are almost always violated in practical speech recognition systems.

One type of discriminative training, *corrective training*, tries to minimize the error rate directly. In corrective training, model parameters are re-estimated on observations that were misrecognized by the current model. The technique increases the *a posteriori* probability of the training data, given the correct transcription, and decreases the *a posteriori* probability of the training data, given the incorrect transcription. Examples and modifications to this basic idea can be found in [3, 4].

Another popular incarnation of discriminative training is maximum mutual information estimation (MMIE)[40]. In addition to maximizing the probability of the observation sequence given the correct transcriptions, MMIE simultaneously minimizes the probability of the observation sequence given all possible incorrect transcriptions. Equivalently, it attempts to increase the *a posteriori* probability of the model corresponding to the training data, given the training data (Equation 1.1).

Initially, there were no algorithms proven to be convergent for discriminative training, and researchers resorted to gradient descent and hill climbing techniques that converged slowly and gave little improvement over EM training.

Generalizations of the classical results of Baum and Eagon[20, 21] have been used to generate an algorithm that has been proven to converge for Markov models with discrete output distributions [39, 40, 29]. This algorithm has also been extended to be used for Markov models with continuous output distributions, but is no longer guaranteed to converge [39]. In practice, it usually converges to a local maxima, and convergence occurs faster than with the old gradient descent techniques.

Significant improvements over EM training, and various improvements to the basic MMIE formulation, have been reported in [39, 35, 9, 29].

1.3.2 Features

Stationary spectral features have been used in speech recognition systems for many years.

The rate at which these features are generated is called the frame rate. The frame rate is typically between 100 and 200 frames per second, wich corresponds to a frame every 5 to 10 ms.

Since the spectral estimate assumes a stationary signal, only a small amount of data is used for each estimate. The amount of data is usually referred to as the window length. Window lengths are typically around 20 to 25 ms.

A qualified study of the evolution of speech recognition features contains insight to what may make a good speech feature. In this context, a good feature is one that represents the entire 25 ms frame in a low-dimensional space. Within this space, each phone should form a cluster that is discriminable from the others.

Short Time Fourier Transform

Figure 1.3 shows a plot of the magnitude of the Fourier transform for a 25 ms segment of speech. The overall envelope of the spectrum is imposed by the vocal tract, and contains information specific to the phone.

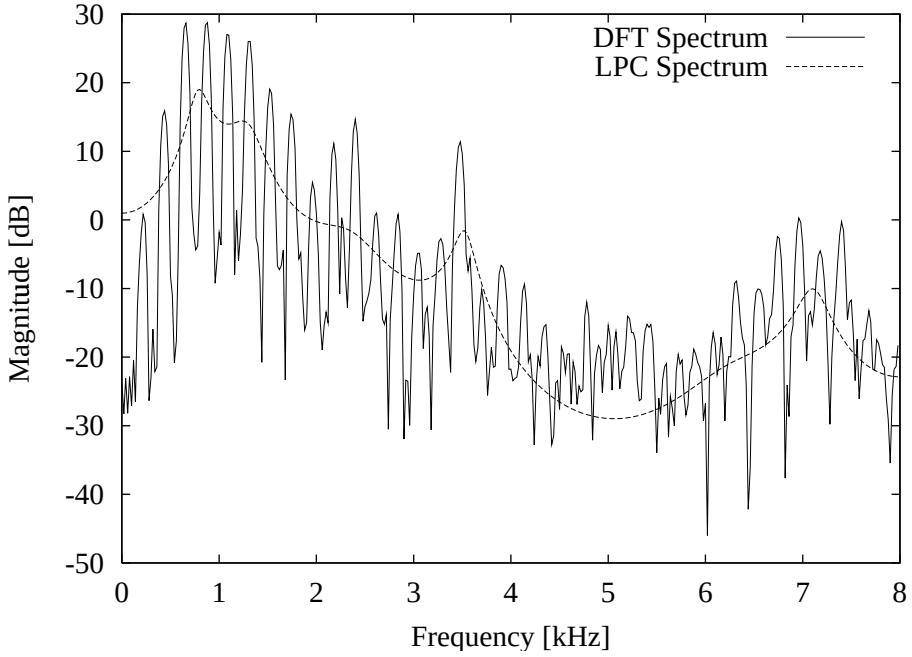

Figure 1.3: Raw and smoothed (LPC) spectrum for /aa/

In addition, the spectrum contains periodicities on the order of 100 Hz due to the underlying pitch produced by the glottis, which provides the forcing function for voiced speech. These periodic variations in the spectrum indicate whether the phone is voiced or unvoiced, but do not contain any other phone specific information.

Equation 1.3 shows how a small segment is extracted from the input signal. The window length is controlled by the parameter L, and the parameter S controls the frame rate.

$$x_n[a] = \begin{cases} x[nS + a]w[a] & \text{if } 0 \leq a < L \\ 0 & \text{otherwise} \end{cases} \tag{1.3}$$

It is assumed that the signal is stationary over the interval defined by L, and the window helps ameliorate the effects of extracting a finite length of samples. The window function used throughout this document is the Hamming window,

$$w[a] = 0.54 - 0.46\cos(\frac{2\pi a}{L}). \tag{1.4}$$

Linear Prediction Coefficients

Linear Prediction Coefficients (LPC) can parameterize the speech spectrum quite well. LPC assume an all-pole speech production model, as shown in Equation 1.5. In this equation, $X(z)$ is the spectrum of the speech signal and $G(z)$ is the spectrum of the glottal excitation, which is assumed to be white. $1/A(z)$ is the spectrum of the vocal tract, where $A(z)$ is modeled as a polynomial function of z.

$$X(z) = G(z)\frac{1}{A(z)} = G(z)\frac{1}{1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_P z^{-P}} \tag{1.5}$$

The LPC are the coefficients $\{a_1, a_2, \ldots, a_P\}$, estimated from the current frame of data, given the speech production model.

An LPC sequence can be computed directly from the corresponding correlation sequence of equal length. A length twelve LPC vector can be derived from a length twelve autocorrelation, which in turn is interpreted as a smoothed Fourier spectrum.

The dashed line in Figure 1.3 shows an example of one spectrum estimated from 25 ms of data, using twelve LPC coefficients. The same data was used to generate both the LPC spectrum and the DFT spectrum in this figure.

Cepstral Coefficients

Cepstral coefficient vectors are the feature of choice for conventional speech recognition systems. An incoming time-domain signal is decomposed into a series of vectors, and each vector contains information about the stationary spectrum of the time domain signal.

The introduction of cepstral coefficients dramatically reduces the error rate in speech recognition systems. The cepstrum of a time series has several desirable properties. It maps convolution onto addition, compresses the range of the magnitude spectrum, and reduces correlation between coefficients.

Mapping convolution onto addition allows easy separation of the glottal excitation and the vocal tract. An incoming signal, composed of the convolution of two signals, produces a cepstrum that is the sum of the cepstra of the two component signals.

Compressing the range of the magnitude spectrum makes peaks and valleys in the spectrum less pronounced, and therefore easier to represent with fewer coefficients.

There are several variants of cepstral coefficients in use. Two of the most common are linear predictive cepstral coefficients (LPCC) and Mel-frequency cepstral coefficients (MFCC).

A mapping exists between a finite sequence of linear predictive coefficients (LPC) and an infinite sequence of LPCC.

The first step, therefore, in calculating an LPCC vector, is to estimate a set of LPC from the data. The LPC are then transformed into LPCC. Unfortunately, it is not practical to calculate an infinite sequence of LPCC, so the sequence is truncated at some point.

This truncated LPCC series has the general advantages of cepstral coefficients, that is, they are less correlated along their dimensions and the spectral envelope has undergone a logarithmic compression. The dashed line in Figure 1.4 represents the spectrum of a length twelve LPCC vector, and the solid line represents the spectrum

represented by a length twelve LPC vector. Both spectra were calculated from the same data.

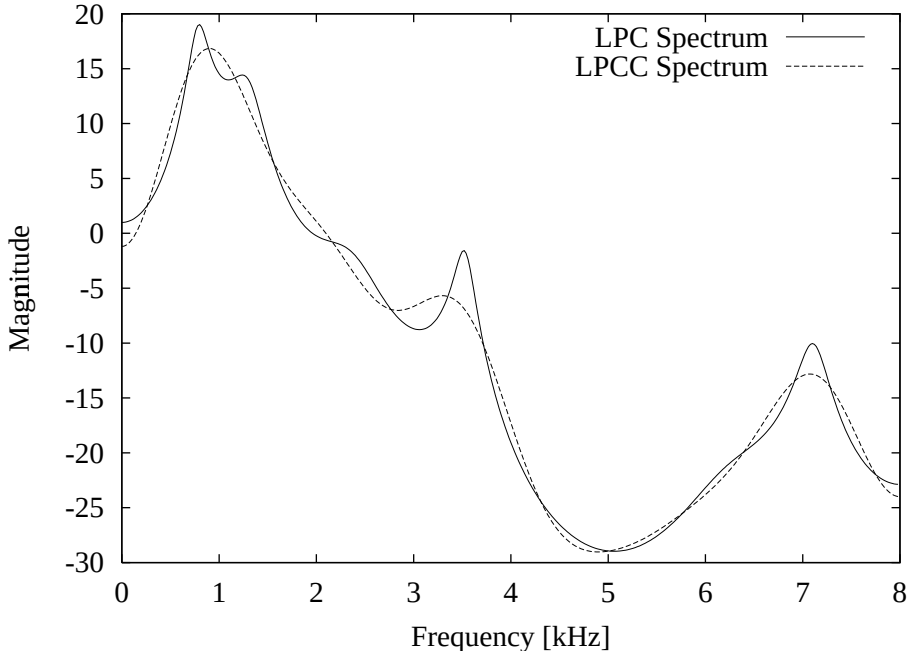

Figure 1.4: LPCC and LPC spectrum for /aa/

Unfortunately, the truncation of the LPCC series introduces a smoothing of the spectrum. The first two resonances have been blurred together in the LPCC spectrum.

Mel-frequency cepstral coefficients warp the Fourier frequency axis in addition to the magnitude during computation. The warping function (Equation 1.6) is usually implemented as a filterbank with linear frequency spacing at low frequencies (less than 1 kHz) and logarithmic spacing at higher frequencies.

$$F_{mel}(f) = \begin{cases} f & \text{if } 0 \leq f < 1 \text{ kHz} \\ \exp\left(\frac{(f-8)\ln(1)-(f-1)\ln(8)}{1-8}\right) & 1 \text{ kHz} \leq f < 8 \text{ kHz} \end{cases} \tag{1.6}$$

The shape of the warping function tends to resolve spectral peaks in speech with fewer coefficients. For example, the MFCC and LPC spectra shown in Figure 1.5 were

calculated from the same data. The peaks in the two curves are not quite aligned because the filterbank does not do a perfect job of implementing equation 1.6. The warping function separated the first two peaks so that both are resolved, without increasing the dimensionality of the representation.

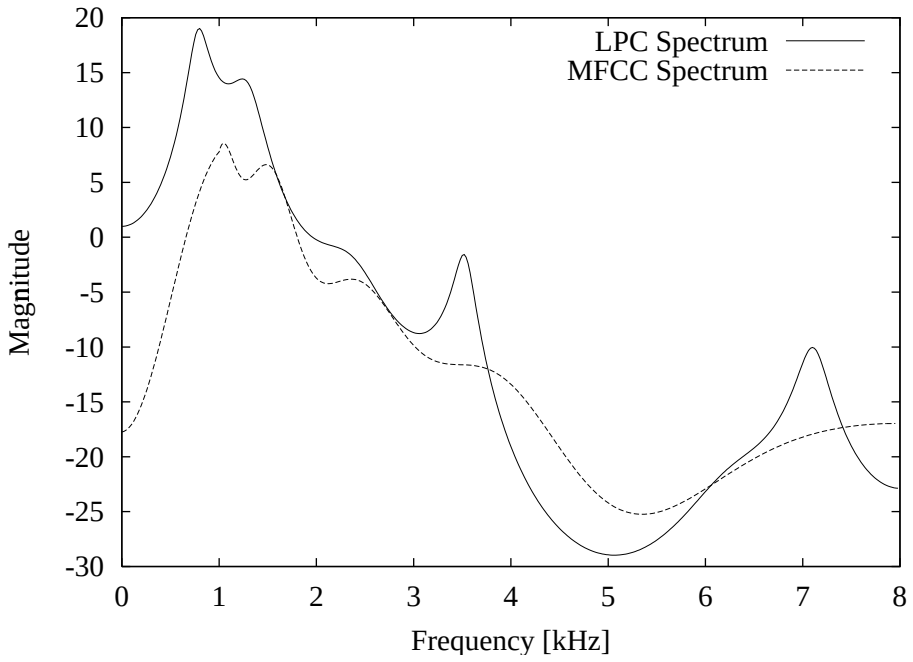

Figure 1.5: MFCC and LPC spectrum for /aa/

In addition to the static feature vectors, it is well known within the speech recognition community that introducing the first and second difference of the static feature vector improves performance significantly.

1.3.3 Language Models

Speech is not a stochastic process whose sounds can come in any order, it contains structure on several levels. Vowels are usually separated by consonants, some pairs of phones are more common than others. An utterance is more likely to contain a sequence of valid words than a nonsensical sequence of sounds. A person is usually

talking about something, trying to convey information using structures like subject-action, prepositional phrases, and the like.

All of these structural clues can be useful in the recognition process. Typically, the phone structure is limited by a fixed vocabulary. Instead of arbitrary sequences, the recognition system limits its search to sequences of phones that form sequences of words taken from within the vocabulary of the recognition system.

Additionally, recognition systems form a stochastic model of word occurrence. Common forms are the unigram, which only considers individual word frequencies, the bigram, which models word to word transition probabilities, and the trigram model, which computes the likelihood of any word based on the words immediately before and after it in the transcription.

1.4 Alternative Markov Model Structures

As mentioned previously, conventional recognition systems are limited to one type of input feature for all models of all phones. Unfortunately, using multiple input streams presents a problem for the traditional HMM framework.

The inputs may be correlated, violating any assumption to the contrary. The dynamic range of the input probability density functions may be significantly different, causing one input to swamp the others. A feature stream may be very good at discriminating certain classes of phones, but terrible at others, but weighted the same regardless of the task.

Attempts to circumvent some or all of these problems can be grouped into three general categories: recombination, feature concatenation, and model expansion.

A handful of researchers are currently working on methods that extract cepstral features in parallel from a filterbank [48, 28, 8, 42, 47, 7]. As Figure 1.6 shows, instead of generating one set of cepstral features for each signal $x[n]$, multiple streams of features are generated, one for each bandpass filter in the filterbank.

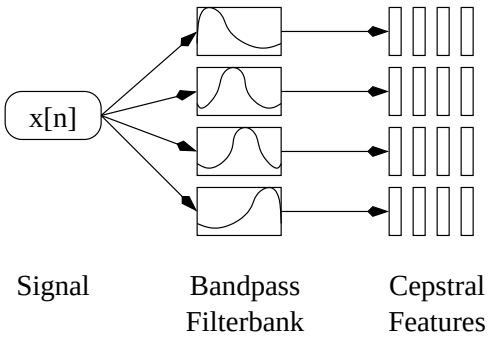

Figure 1.6: Multiple features from one speech signal

Systems built in this way have been shown to be comparable to more traditional systems in the absence of noise, but more robust to some types of additive noise.

It is not the sub-band features themselves which are of interest in this context, but the ways in which these possibly correlated features are combined into a recognition system. Typically, several recognizers are trained and run in parallel, and their outputs are joined at an intermediate level [6, 7, 23, 47].

1.4.1 Likelihood Recombination

Researchers have tried separately training one model for each input stream, and then forcing alignment between the streams at the frame, phone, syllable, or word level [6, 7, 23, 47]. Alternatively, a full (unconstrained) recognition can be done on all bands, and then post-processed to remove inconsistencies [8].

Unfortunately, some streams may do poorly on certain phones, leading to poor recognition results. A weighting scheme for the different streams has been proposed [7], as well as non-linear techniques [47], but reliable estimation of the parameters remains an open problem.

1.4.2 Feature Concatenation

These systems have the same structure as the conventional system, but their feature vectors contain a concatenation of all of the input streams. This method was successfully applied to multi-band speech recognition in [42], and shown to be more effective than the likelihood recombination. Furthermore, the technique was used with multi-resolution cepstral features in [33] and shown to improve speaker-independent continuous phone recognition with the TIMIT database.

Although streams with poor recognition performance play less of a role, the curse of dimensionality[5] may come into play. Even if a useless feature stream adds nothing on average to the output probabilities of the model, it still adds noise which corrupts the recognition accuracy. As more features are included in the concatenation, one must increase the complexity of the model. If the training set size remains constant, the model quickly becomes under-trained.

Because we must be open to the possibility of strong correlation between feature streams, as well as mismatches in dynamic range and usefulness of individual streams, feature concatenation is not a good choice for this task.

1.4.3 Model Expansion

With the model expansion approach, the topology of the Markov model is changed to allow for simultaneous use of data from one or more streams.

Figure 1.7 shows a modified forward HMM for use with two streams. States in the upper path model data from stream one, and states in the lower path model data from stream two. The HMM is allowed to transition between paths, depending on which better describes the current input.

With this approach, the representation of the signal is changing according to the current state of the model. To make such a system work, we can no longer ignore the $P(\mathbf{X})$ term in Equation 1.2. Re-normalization of the output probabilities will be

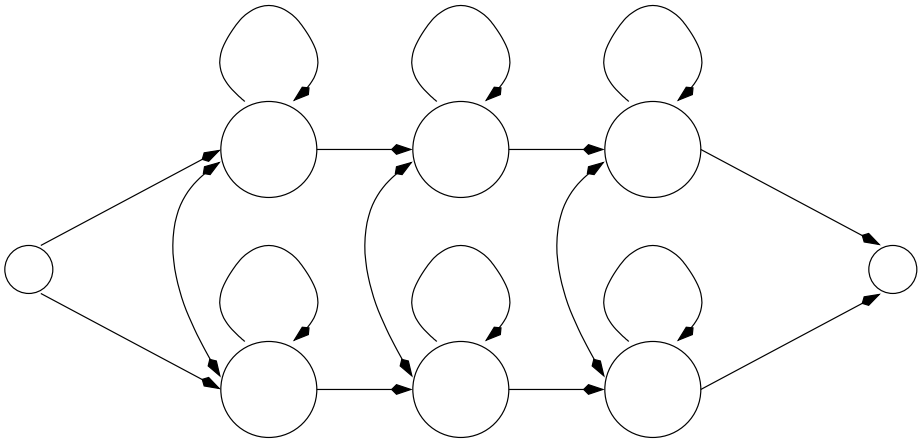

Figure 1.7: Expanded six state hidden Markov model

necessary to make the recognition work correctly.

This method provides a great flexibility to choose among the feature streams available, on a frame by frame basis. With this flexibility, however, there is also a downside. The complexity of the model increases greatly. In addition to increasing the number of parameters in the model, the state information necessary for a proper normalization during recognition grows quite large. For these reasons, it was decided that this topology is not appropriate for the task at hand.

1.4.4 Parallel Models

An alternative to the model expansion would be to disallow transitions between the rows of states shown in Figure 1.7. This would be equivalent to having one simple model for each stream, forced alignment at the phone level, and a simple likelihood recombination scheme.

A simple approach for approximating $P(\mathbf{X})$ for each feature stream $\mathbf{X^{(i)}}$ is to constrain the model to use only one stream during training. The data would be set aside, and the process repeated for each stream. During the final recognition stage,

these probabilities can be used to normalize the likelihoods of each stream.

During training, one set of models is independently trained for each feature stream. Ideally, in the decoder, the search space is augmented with an additional feature stream dimension. The decoder considers all the features and finds a word sequence with maximum likelihood over all the models, states and feature streams. Such a system is described in [27].

1.5 Conclusion

Stationary spectral features, together with Mel frequency cepstral coefficients, do an adequate job of modeling the time varying nature of speech, but are not a perfect fit to the task.

A new feature, that models the time varying spectrum of speech directly, has the potential of supplementing or replacing these stationary spectral features. A theory of fully discrete time-frequency features is introduced and developed in the next four chapters.

In Chapter 6, it will be shown that these new features can be used to supplement the Mel frequency cepstral coefficients in a parallel hidden Markov model structure. The end result is a significant reduction in error rate.

Chapter 2

QUADRATIC TIME-FREQUENCY REPRESENTATIONS

Quadratic time-frequency representations have all of the descriptive power of conventional spectral features, together with a powerful framework that describes how the spectrum evolves over time.

This chapter begins with an overview of that portion of previous work with continuous time-frequency representations that is relevant to the present discussion.

This overview is followed by our derivation for discrete time-frequency representations. Some of the more interesting properties of these representations are presented, as well as a short discussion of how to read and interpret these representations.

Finally, analogies are made to more conventional features that represent a subset of the descriptive space of discrete time-frequency representations.

2.1 Introduction

Leon Cohen developed a systematic way of obtaining bilinear, joint bivariate densities of time and frequency using operator theory[11].

More recently [12], he presented a generalized approach for obtaining joint representations for arbitrary quantities (not just time and frequency) using the characteristic function operator methods of Moyal [36] and Ville [49].

In both papers, there is an emphasis on relating the time-frequency representations to probability distributions of energy in two variables, most commonly time and frequency. Although it is nice to think of the TFR as describing the signal's energy distribution, such a mindset leads one to distracting questions. Why is this distribu-

tion complex valued, and should I force it to be real? Can a signal have any energy at the instant it crosses through zero? Should a distribution of the sum of two sines be limited to have two components, or is three satisfactory? In this dissertation, I choose to use the term "representation" in place of "distribution" to avoid traps posed by unnecessary semantic ties.

2.2 Continuous Time, Continuous Frequency

To obtain a continuous time, continuous frequency representation via Cohen's method, we must first define what we mean by time and frequency. This is accomplished by defining operators $\mathcal{T}$ and $\mathcal{W}$, in terms of how they affect a continuous time signal $x(t)$. Since this theory, in general, works with complex signals $x(t)$, the notation $x^*(t)$ indicates its complex conjugate.

$$\int x^*(t)\mathcal{W}x(t)dt = \int X^*(\omega)\omega X(\omega)d\omega \tag{2.1}$$

$$\mathcal{T}x(t) = tx(t) \tag{2.2}$$

2.2.1 Derivation

In the continuous time case, operators for time and frequency can be used together to produce time-frequency representations. A continuous distribution in one variable, $P(a)$, with its corresponding operator $\mathcal{A}$, can be calculated through its characteristic function $M(\alpha)$.

$$M(\alpha) = \langle \exp(j\alpha\mathcal{A})\rangle = \int s^*(t)\exp(j\alpha\mathcal{A})s(t)dt \tag{2.3}$$

$$P(a) = \int M(\alpha)\exp(ja\alpha)d\alpha \tag{2.4}$$

Here, the operator function $\exp(j\alpha\mathcal{A})s(t)$ can be directly computed from the definition of the operator $\mathcal{A}$

For a continuous joint time-frequency representation, $P(t,\omega)$, the characteristic function is defined as

$$M(\eta,\tau) \;=\; \langle \exp(j\mathcal{T}\eta + j\mathcal{W}\tau)\rangle\,. \tag{2.5}$$

Unlike the case for the single operator, in equation 2.5 there are two operators within a single exponential function. Since $\mathcal{T}$ and $\mathcal{W}$ are operators, they may or may not commute. As such, they cannot be manipulated as if they were variables. In fact, each possible reordering of the operators yields a unique time-frequency representation.

In [12], Cohen shows how these different orderings are equivalent to introducing a multiplicative kernel function $\phi(\eta,\tau)$ into Equation 2.5 while holding the ordering constant. This is known as the correspondence rule. The set of all distributions that can be generated from Equations 2.5 and 2.7 is referred to as Cohen's class of quadratic time-frequency representations. The characteristic function can then expressed as

$$M(\eta,\tau) \;=\; \phi(\eta,\tau)\int s^*(t)\exp(j\mathcal{T}\eta)\exp(j\mathcal{W}\tau)s(t),\ \text{and} \tag{2.6}$$

$$P(t,\omega) \;=\; \int\int M(\eta,\tau)\exp(-j\eta t)\exp(-j\tau\omega)d\eta d\tau. \tag{2.7}$$

In this way, each signal is associated with not one, but a multitude of time-frequency distributions, each one uniquely specified by the choice of a root representation and a kernel function.

The choice of the root representation is critical. If it is chosen correctly, then all quadratic TFRs associated with the signal can be generated by using the appropriate kernel function.

The root representation typically has a lot of detail in both time and frequency, and the kernel function selects some details to emphasize and other details to smooth. Traditionally, kernels are chosen to impart specific properties to the resulting distributions, such as satisfying the marginals or reducing cross-terms.

2.3 Discrete Time, Discrete Frequency

Although Cohen's class of continuous quadratic TFRs are backed by rich theory, they have to undergo discrete sampling, resulting in discrete-time, discrete-frequency functions, before they can be implemented in any digital system and be of practical use. A considerable amount of work has gone into the study of sampling the continuous-time TFRs and ameliorating the ill effects introduced by the process, e.g. [41].

Our approach to formulating discrete-time, discrete-frequency representations[38] provides a direct theoretical link between the discrete time sequence $x[n]$ and its discrete time-frequency representation $P[n, k]$.

This theoretical link is provided by discrete operator theory. A similar method of deriving discrete time-frequency representations, based on group theory, has been presented by Richman, *et al.* in [44]. One notable difference between the two is that whereas Richman's technique incorporates different calculations for even and odd length signals, our technique is the same for all signal lengths.

As in the continuous case, each discrete time sequence is associated with not one, but a multitude of time-frequency distributions, each one uniquely specified by the root distribution and a kernel function.

2.3.1 Derivation

We begin by assuming that we have is a discrete, periodic sequence $x[n]$ with period N. The discrete Fourier transform is given by $X[k]$. We then work towards obtaining $P[n, k]$ by developing discrete (matrix) operators corresponding to the discrete time and frequency variables. From here, we can calculate our distribution $P[n, k]$. Important differences exist between the set of sampled continuous-time quadratic TFRs and the set of discrete TFRs obtained through discrete operator theory. Through operators, we are able to obtain additional insights into existing TFRs as well as establish a framework for formulating new distributions.

To develop a fully discrete theory, we formulated the discrete counterparts of the time and frequency operators. This yields a theory that is distinct from sampling a distribution produced with the continuous time and frequency operators. There are problems inherent in sampling the continuous distribution that the discrete theory circumvents, by mapping directly from a discrete signal to a discrete time-frequency representation.

The continuous time and frequency operators are defined by Equations 2.1 and 2.2. When these equations are reformulated in discrete time, with a discrete frequency operator, $\mathcal{K}$, and a discrete time operator, $\mathcal{L}$, we get

$$\sum_{n=0}^{N-1} x^*[n]\mathcal{K}x[n] = \sum_{k=0}^{N-1} X^*[k]kX[k], \text{ and} \tag{2.8}$$

$$\mathcal{L}x[n] = nx[n]. \tag{2.9}$$

Since the operators $\mathcal{K}$ and $\mathcal{L}$ are discrete and linear, their operation on discrete sequences can be compactly represented as a matrix-vector product. The sequence $\{x[0], x[1], \ldots, x[N-1]\}$ is represented by the vector $\mathbf{x}$. The Hermitian transpose of $\mathbf{x}$ is represented by the notation $\mathbf{x}^*$. The operators $\mathcal{K}$ and $\mathcal{L}$ are represented as matrices $\mathbf{K}$ and $\mathbf{L}$, with scalar elements K_{ij} at row i and column j.

In this form, the solution to Equation 2.9 is a diagonal matrix with values $\{0, 1, \ldots, N-1\}$ along its diagonal.

The solution to Equation 2.8 is

$$\mathcal{K}x[n] = o[n] \otimes x[n], \text{where} \tag{2.10}$$

$$o[n] = \begin{cases} \frac{(j/2)(-1)^n}{\sin(\pi n/N)} & n \neq 0 \\ 0 & n = 0 \end{cases}. \tag{2.11}$$

The symbol $\otimes$ indicates circular convolution, and the matrix $\mathbf{K}$ is therefore a circulant matrix with the sequence $o[n]$ populating its first column.

Our discrete operators can now be used to form a discrete-time version of the characteristic function.

$$M[\eta,\tau] = \langle \exp(j2\pi\mathbf{L}\eta + j2\pi\mathbf{K}\tau)\rangle \tag{2.12}$$

Just as in the continuous case, we can take advantage of the correspondence rule to represent permutations of the operators in Equation 2.12 as a multiplicative kernel function $\phi[\eta,\tau]$.

Using the characteristic function in Equation 2.12 above, and absorbing the constants into the operators for convenience,

$$\begin{aligned} M[\eta,\tau] &= \langle \exp(j\eta L)\exp(j\tau K)\rangle \\ &= \sum_n x^*[n]\exp(j\eta L)\exp(j\tau K)x[n+\tau] \qquad (2.13) \\ M[\eta,\tau] &= \sum_n x^*[n]\exp(j\eta L)x[n+\tau] \qquad (2.14) \end{aligned}$$

using the shift property. A two dimensional discrete Fourier transform gives us the time-frequency representation,

$$P[n,k] = \sum_\tau \sum_\eta M[\eta,\tau]\exp(-j\tau k)\exp(-j\eta n).$$

Substituting Equation 2.14 for the auto-ambiguity function $M[\eta,\tau]$,

$$P[n,k] = \sum_\tau \sum_\eta \left(\sum_u x^*[u]\exp(j\eta L)x[u+\tau]\right)\exp(-j\tau k)\exp(-j\eta n).$$

Since $\exp(j\eta L) = \exp(j\eta n)$ in the time domain, we get

$$\begin{aligned} P[n,k] &= \sum_\tau \sum_\eta \left(\sum_u x^*[u]\exp(j\eta u)x[u+\tau]\right)\exp(-j\tau k)\exp(-j\eta n) \\ &= \sum_\tau \sum_u x^*[n]x[u+\tau]\delta[n-u]\exp(-j\tau k) \\ &= \sum_\tau x^*[n]x[n+\tau]\exp(-j\tau k) \\ &= x^*[n]\exp(jnk)X[k], \qquad (2.15) \end{aligned}$$

which is the discrete version of the well-known Rihaczek TFR[45]. This result is analogous to the continuous-time result which can be derived in a very similar fashion.

Including the kernel function $\phi[\eta, \tau]$, the discrete time-frequency representation in the auto-ambiguity plane is given by

$$M[\eta, \tau] = \phi[\eta, \tau] \sum_{n=0}^{N-1} x^*[n-\tau]x[n] \exp\left(\frac{j2\pi\eta n}{N}\right). \quad (2.16)$$

$$P[n, k] = \sum_{\eta=0}^{N-1} \exp(\frac{-j2\pi\eta n}{N}) \sum_{\tau=0}^{N-1} \exp(\frac{-j2\pi\tau k}{N}) M[\eta, \tau] \quad (2.17)$$

Members of this new class of discrete time-frequency representations include the discrete Rihaczek TFR [45], the discrete Margeneau-Hill, and the spectrogram.

Given a discrete time signal $x[n]$, operator theory [38] dictates that an acceptable root distribution is the discrete time, discrete frequency Rihaczek representation[45],

$$P_R[n, k] = x^*[n]X[k] \exp(\frac{j2\pi nk}{N}). \quad (2.18)$$

By taking the two dimensional discrete Fourier transform, this is expressed in the ambiguity plane as

$$M_R[\eta, \tau] = \sum_n x^*[n]x[n+\tau] \exp(\frac{j2\pi n\eta}{N}). \quad (2.19)$$

Can a root representation other than the Rihaczek be used? Yes. Since one TFR can be derived from any other with application of the appropriate transforming kernel, any TFR related to the Rihaczek through an invertible kernel may serve as an initial, base representation in our method. The optimal discriminating kernel will vary with the base TFR chosen, however, due to the varying amounts of time-frequency similarity between the signal classes.

The representation described by Equation 2.19 contains a lot of information about the signal $x[n]$. If $\eta = 0$, then $M_R[\eta, \tau]$ becomes the stationary autocorrelation of the signal $x[n]$, with τ taking its familiar role as the time lag variable. If $\tau = 0$, then $M_R[\eta, \tau]$ becomes the spectrum of the instantaneous energy of the signal $x[n]$. This leads us to refer to η as the modulation frequency variable, because it relates to how quickly the envelope of the signal is changing. In general, a point (η, τ) refers to how quickly a specific correlation coefficient is being modulated.

2.3.2 Interpretation

Time-frequency representations promise to capture both static spectral information and evolutionary spectral information in a single feature. Although representations in time and frequency usually make intuitive sense to the untrained viewer, representations in η and τ, as in Equation 2.16, tend to be less intuitive.

In general, stationary noise has no extent in η, white noise has no extent in τ, stationary white noise is concentrated at the origin, and more interesting structures appear when $\eta \neq 0$ and $\tau \neq 0$. Time modulations increase a signal's extent in η, and colored spectra are responsible for extent in τ.

The lag variable, τ, is related to time correlations in the signal. In fact, along $\phi[0, \tau] = 1$, Equation 2.16 reduces to the stationary autocorrelation of the signal, which is related to the squared magnitude stationary spectrum of the signal through a Fourier transform.

$$A[0, \tau] = \sum_{n=0}^{N-1} x^*[n - \tau]x[n] \tag{2.20}$$

As a result, values along the $\eta = 0$ axis are always interpreted as the autocorrelation of the signal $x[n]$. Conversely, any stationary signal will concentrate its energy within this region. Figure 2.1 is one example of a signal with this behavior. The signal $x[n]$ was generated by passing stationary white Gaussian noise through an FIR filter to color its spectrum. The filtering operation does not change the stationarity of the signal, although it may change its amplitude.

The modulation frequency variable, η, is related to time modulations of the signal. Along $\phi[\eta, 0]$, Equation 2.16 reduces to the spectrum of the squared magnitude of the original signal.

$$A[\eta, 0] = \sum_{n=0}^{N-1} |x[n]|^2 \exp\left(\frac{j2\pi\eta n}{N}\right) \tag{2.21}$$

As a result, values along the $\tau = 0$ axis show how the modulation "envelope"

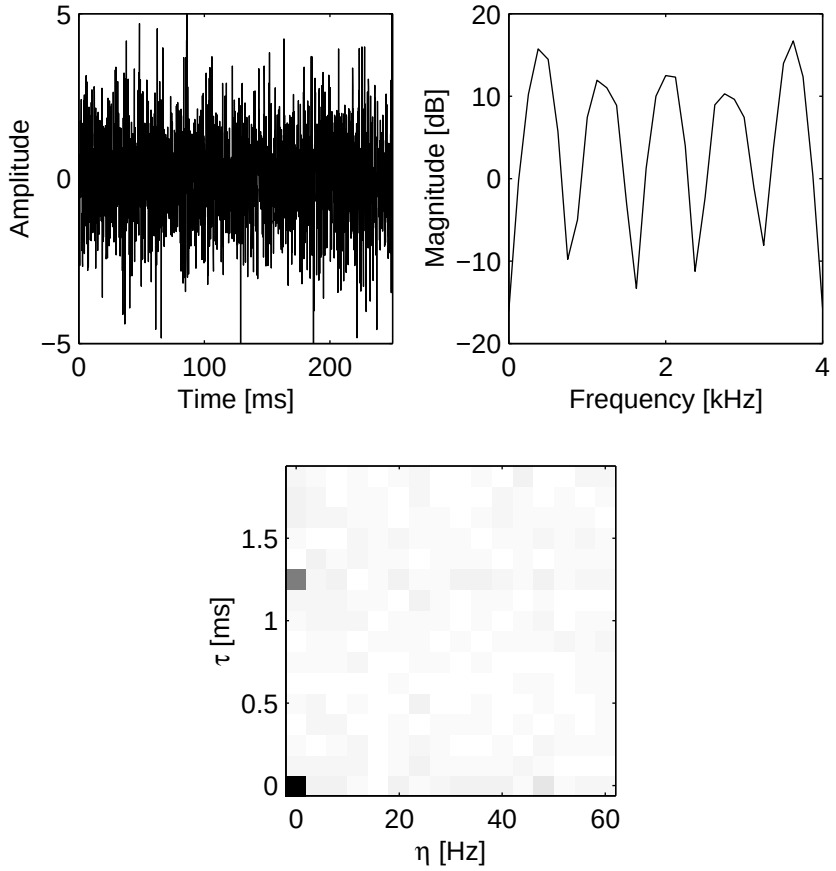

Figure 2.1: Colored Gaussian noise

of the signal $x[n]$ is changing with time. A signal with little correlation between its samples will concentrate its energy in this region. Figure 2.2 is one example of a signal with this behavior. The signal was generated by modulating stationary white Gaussian noise with a $\cos(t)$ function. This process does not change the correlation of the individual samples, but does make the signal quite non-stationary.

If a signal is both stationary and white (uncorrelated), such as stationary white Gaussian noise, it will concentrate all of its energy at the origin. Such a case is shown in Figure 2.3.

More generally, one can interpret each sample in the $(\eta \times \tau)$ plane as the output

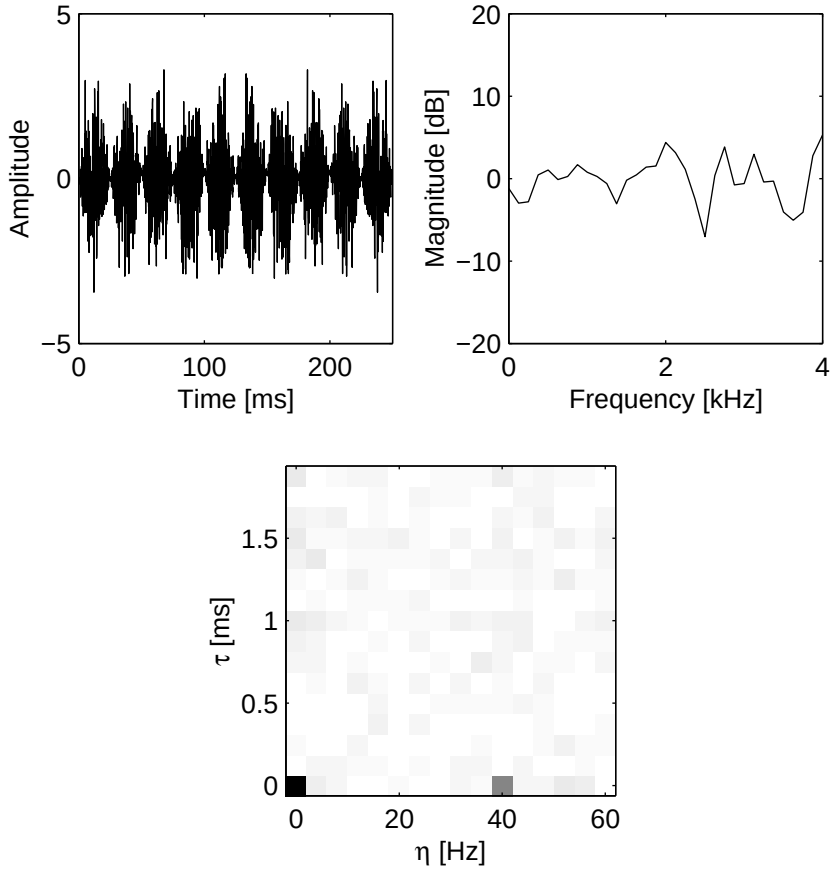

Figure 2.2: Modulated white Gaussian noise

of a non-stationary sinusoidal filterbank, where distance from the origin is analogous to bandwidth, and the angle from the η axis represents the linear chirp rate of the filterbank. To see this, consider a kernel function

$$\phi[\eta, \tau] = \delta[\eta - a]\delta[\tau - b], \text{ where} \tag{2.22}$$

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2.23}$$

In the $(n \times k)$ plane, this kernel becomes a complex exponential, with parameters controlled by a and b.

$$\Phi[n, k] = \exp\left(-j2\pi(an + bk)\right) \tag{2.24}$$

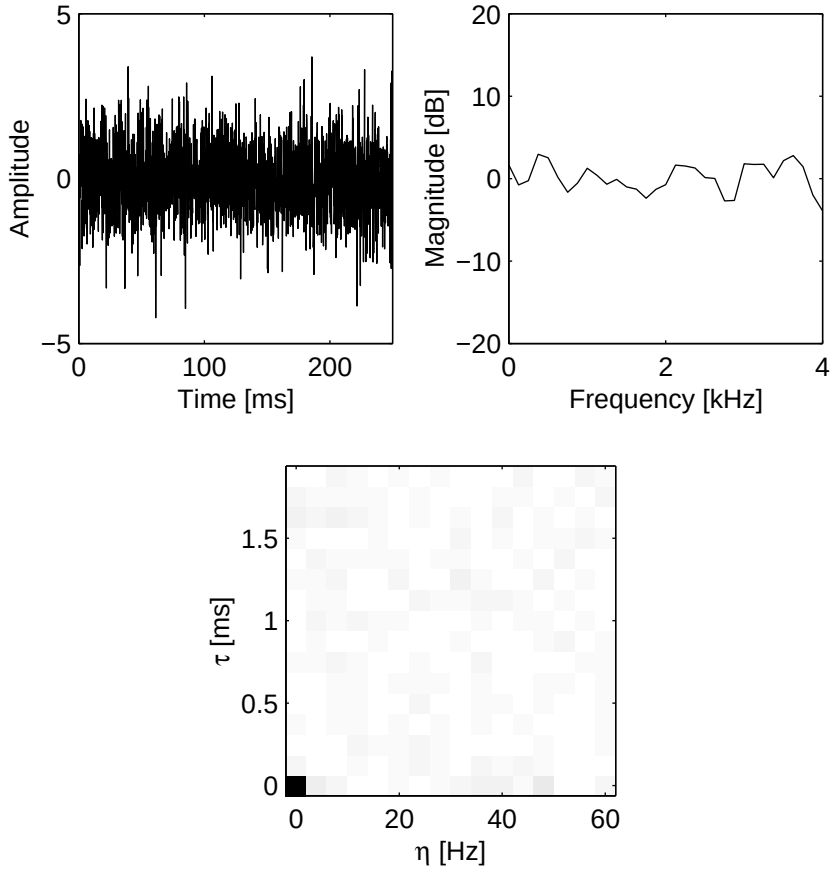

Figure 2.3: White Gaussian noise

2.4 Unleashing the Power of Time-Frequency Representations

Initial work into time-frequency representations focused on generating images that humans could examine and extract relevant information. This research tended to focus on developing representations that have desirable properties, where "desirable" may be a subjective term.

These properties include, but are not limited to, satisfying marginal equations, being strictly positive or real, and having reduced cross-terms[12], resulting in representations such as the Rihaczek, Wigner-Ville, Choi-Williams, and Zhao-Atlas-Marks (cone kernel), among others [12]. Generally, these properties are chosen because they

produce more visually appealing representations, or TFRs that preserve all of the information in the original signal.

The traditional disadvantage to using these representations as features is the deluge of information. Whereas stationary spectral features produce vectors with at most a few dozen dimensions, a real length N signal will always produce a representation with $N(N+1)/2$ linearly independent dimensions. For a 250 ms signal sampled at 16 kHz, this amounts to no less than eight million dimensions.

There have been several attempts in the literature to reduce this information into salient features, including taking autocorrelation coefficients of the representation in time or frequency to achieve a time-shift or frequency-shift invariant representation of the data [55], and finding two dimensional moments of the time-frequency representation [46].

Although some of these representations may offer advantages in classification for certain types of signals, they cannot hope to offer improved performance for all signals because discrimination is not one of the goals of the feature design and dimensionality reduction.

Given the apparently immense dimensionality of the feature space, one might assume that extracting useful information in an automated and efficient way would be impossible. This is not the case. The next chapters present a new way of looking at these representations.

Chapter 3 shows an efficient method for computing a low noise, full covariance estimate with limited data, even when the dimensionality of the feature space is quite high.

Additionally, Chapter 4 uncovers the structure of the space of valid discrete time-frequency representations. This space is an N dimensional surface embedded in an $N(N+1)/2$ dimensional feature space.

A method that reduces the dimensionality of the feature space, and improves classification accuracy at the same time, is explored in detail in Chapter 5.

Chapter 3

STATISTICAL PROPERTIES OF QUADRATIC TIME-FREQUENCY REPRESENTATIONS

A fully discrete theory was developed in Chapter 2, that maps a discrete signal directly into a discrete time-frequency representation. These representations exist in very high dimensional spaces, so nonstandard techniques must be employed if we are to model them. This chapter outlines one technique that has proved quite valuable in uncovering the covariance of a set of time-frequency representations.

First, a method is developed that generates a low variance estimate of the covariance of the feature distributions. Given the high dimensionality of our time-frequency features, this method is critical to implement a recognition system using a reasonable set of training data.

The within-class distribution of feature data is influenced by two factors. First, the parameters of the source may be changing with time or between sources. This is typically dealt with by introducing multiple mixtures into a Gaussian model, and is not the subject of this dissertation. Standard techniques, such as using mixtures of distributions, should adequately model this variance.

The second source of randomness in the features is that they are being estimated from a finite amount of data output from a stochastic process. This chapter seeks to evaluate the latter cause, and leverage this advantage when building models for the high dimensional time-frequency features.

3.1 Overview

Signal models are typically used to prescribe parameters that describe the system of interest. The front end of a classification system then estimates these parameters, and produces a feature vector. The back end of a classification system knows the probability distribution functions of the feature vectors produced by each class of signals. This is commonly done with Gaussian distributions.

Unfortunately, feature distributions are not guaranteed to be Gaussian. Even if they were, the feature space is usually large enough to disrupt a full covariance analysis. Traditionally, decorrelating transforms such as the discrete cosine transform (DCT) are used to overcome the need for a full covariance estimate, and Gaussian mixtures are used to model the non-Gaussian distribution.

This chapter shows how, for the case of our quadratic time-frequency representations, the model that defines the parameters can be used to predict their covariance structure.

3.2 Signal Model

In the following sections, a signal model will prove useful. The model we adopted is that of stationary noise through a time-varying linear filter. This class of signals can be expressed in the form

$$y[n] = \sum_m h[n,m]x[m], \text{ where } x \text{ is } N(\mu_x, \sigma_x) \tag{3.1}$$

In Equation 3.1, it is assumed that the input to the filter, $x[n]$, is a stationary Gaussian process, with a mean μ_x and variance σ_x. The function $h[n,m]$ represents the time-varying filter, which takes on values at the time n and lag m.

Applying this model to the formula for the auto-ambiguity function in the frequency-correlation plane, we can see that it can be re-written in terms of the filter parameters $h[n,m]$.

$$
\begin{aligned}
A[\eta,k] &= Y[k]Y^*[\eta+k] \\
&= \sum_a \left(\sum_m h[a,m]x[m]W_N^{-ak} \right) \sum_b \left(\sum_n h[b,n]x[n]W_N^{-b(\eta+k)} \right)^* \\
&= \sum_{a,b} \sum_{m,n} h[a,m]h^*[b,n]x[m]x^*[n]W_N^{-ak}W_N^{b(\eta+k)} \\
&= \sigma_x^2 \sum_n H[k,n]H^*[\eta+k,n]
\end{aligned}
$$

For the case of a linear time-invariant system, the values of $h[m,n]$ in Equation 3.1 are independent of n. This case is considered first, and then extended to include the general case where the filter can be time-varying and the entire auto-ambiguity plane is significant.

In the next section, the theory will be built up and demonstrated for stationary signals. Their simple and non-ambiguous nature make it easy to illustrate the basic concepts. Then, the theory will be extended to the general case of nonstationary signals.

3.3 Estimation of Autocorrelation Coefficients

Consider the class of signals $\mathcal{X}$ whose members $x[n]$ are created by filtering stationary white Gaussian noise through a linear time invariant (LTI) filter. The variance of the noise, together with the filter parameters, entirely describe the signal class $\mathcal{X}$.

Both the expected value of either the power spectrum spectrum $|X(e^{j\omega})|^2$ and the autocorrelation coefficients, $c[\tau]$ also unambiguously describe this class of signals.

The signals $x[n]$ are drawn from an ergodic process, so we may estimate the stationary autocorrelation coefficients $c[\tau]$ with a time average of the instantaneous autocorrelation estimate $\hat{r}[n,\tau]$.

$$\hat{r}[n,\tau] = x[n]x^*[n+\tau] \tag{3.2}$$

$$\hat{c}[\tau] = \frac{1}{N}\sum_n \hat{r}[n,\tau] = \frac{1}{N}\sum_n x[n]x^*[n+\tau]$$

Since $x[n]$ is a sequence of random variables with known properties, the estimators $\hat{r}[n,\tau]$ and $\hat{c}[\tau]$ are also random variables, and we can derive their properties.

3.3.1 Instantaneous autocorrelation estimate

First, consider the instantaneous autocorrelation estimate in Equation 3.2. This case is interesting because both the stationary autocorrelation and the entire auto-ambiguity functions are built from linear combinations of these estimates.

Here, $\hat{r}[n,\tau]$, is the product of two correlated Gaussian random variables. Although the sum of two Gaussian random variables is itself Gaussian, the result does not extend to the product. To derive the true distribution, first consider the joint Gaussian distribution of $\mathbf{u} = x[n]$ and $\mathbf{v} = x[n+\tau]$.

$$f_{\mathbf{u},\mathbf{v}}(u,v) = \exp\left(\frac{c[0]u^2 - 2c[\tau]uv + c[0]v^2}{-2\left(c^2[0] - c^2[\tau]\right)}\right)$$

The distribution of the product $\mathbf{w} = \mathbf{uv}$ can be found from

$$\begin{aligned} f_{\mathbf{w}}(w) &= \int_{-\infty}^{\infty} \frac{1}{|z|} f_{\mathbf{u},\mathbf{v}}(z, w/z)dz \\ &= \frac{1}{\pi\sqrt{c^2[0] - c^2[\tau]}} \exp\left(\frac{-wc[\tau]}{c^2[0] - c^2[\tau]}\right) K_0\left(\frac{|w|\,c[0]}{c^2[0] - c^2[\tau]}\right) \end{aligned}$$

Figure 3.1 plots this distribution, where one random variable has a correlation of 0.5, and the other has a correlation of -0.5.

The Bessel function $K_0(t)$ has a singularity at $t = 0$, and both it and the exponential term have very long "tails" which preclude any hope of fitting a single Gaussian to this distribution.

Figure 3.2 shows the mismatch if one were to try and fit a Gaussian distribution to the product PDF. The Gaussian distribution is parabolic in this Figure, and it is obvious that the product distribution is not even nearly Gaussian.

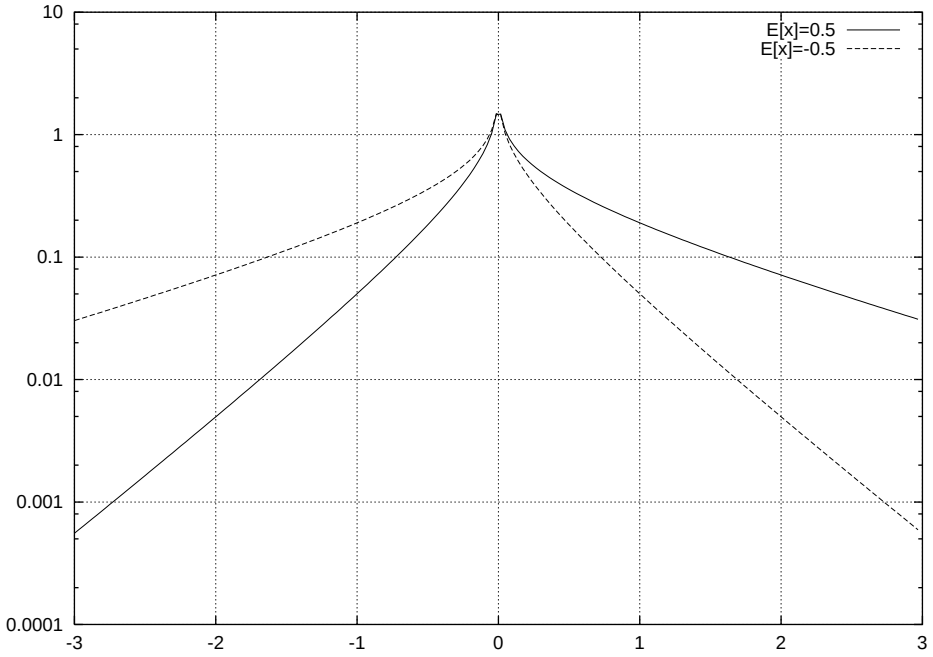

Figure 3.1: K_0 function

One would expect this instantaneous autocorrelation estimate to have an expected value equal to the stationary autocorrelation and a large variance, which is indeed the case.

The expected value of the instantaneous autocorrelation is

$$E\left[\hat{r}[n,\tau]\right] = E\left[x[n]x[n+\tau]\right] = c[\tau]$$

The variance of this estimate is

$$E[(\hat{r}[n,\tau])^2] - E[\hat{r}[n,\tau]]^2 = E\left[x^2[n]x^2[n+\tau]\right] - c[\tau]^2 = c^4[0] + c^2[\tau]$$

3.3.2 Stationary Autocorrelation Estimate

The stationary autocorrelation estimate in Equation 3.3 is an average of several instantaneous autocorrelation estimates. Thanks to the central limit theorem, the distribution becomes more and more Gaussian as more points are added.

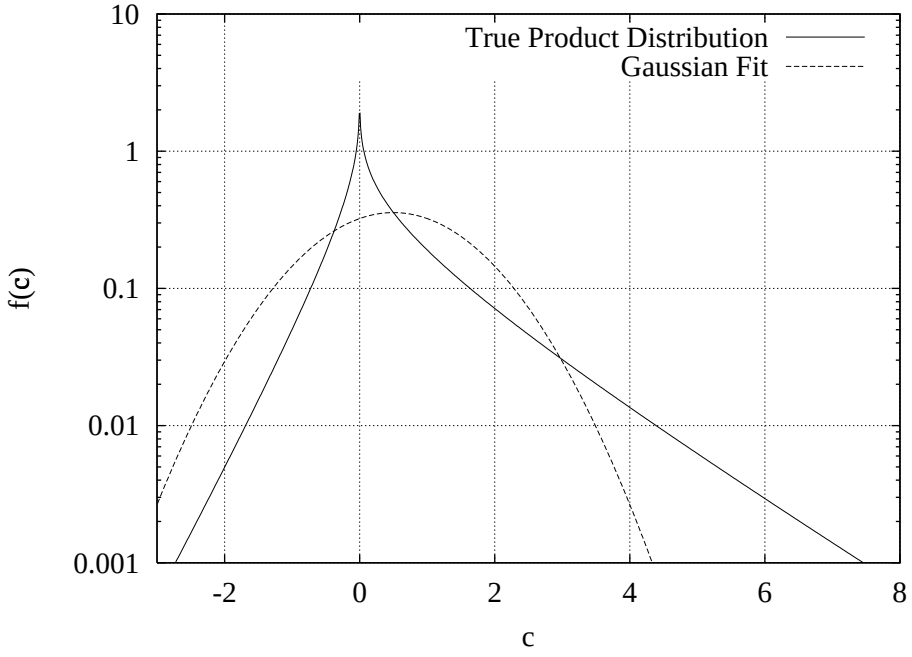

Figure 3.2: Probability distribution function for a product of two Gaussian random variables, together with its best Gaussian fit

Figure 3.3 shows the PDF for the case when sixteen independent instantaneous autocorrelation estimates are used to form one stationary autocorrelation estimate. This distribution, generated from synthetic data, is more nearly Gaussian. This result is not quite as promising as it might first seem, because in practice the instantaneous autocorrelation estimates from any given signal are correlated in time. To achieve a similar result, you would need data equivalent not to sixteen time samples of the signal, but sixteen correlation lengths of the signal.

It is currently unknown how many points are required in general to meet the Gaussian assumption, or how correlation between the instantaneous autocorrelation estimates affects the shape of the estimator's PDF. For real speech samples, it does not appear that 25ms of data is enough to get true Gaussian distributions. Figure 3.4 shows a typical distribution for real speech data. Although the data appears uni-

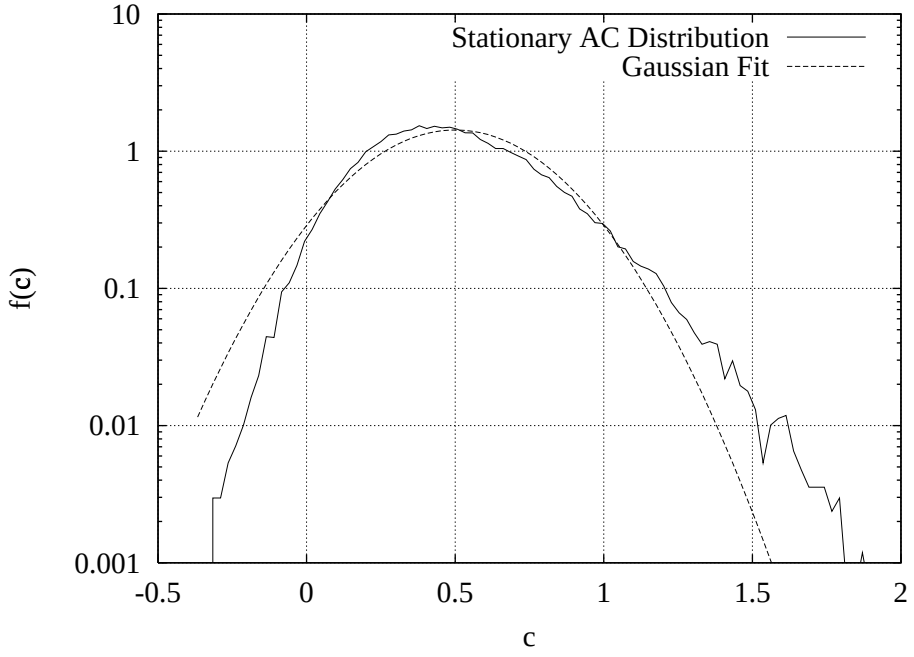

Figure 3.3: Probability distribution function stationary autocorrelation estimate, together with its best Gaussian fit

modal, as expected, it has long non-Gaussian tails, and the tail pointed away from zero is heavier than its counterpart.

As in the previous section, the expected value of this estimate is the true autocorrelation coefficient.

$$E\left[\hat{c}[\tau]\right] = E\left[\frac{1}{N}\sum_n \hat{r}[n,\tau]\right] = E\left[\frac{1}{N}\sum_n x[n]x^*[n+\tau]\right] = c[\tau]$$

The variance of this estimate can be represented as

$$\begin{aligned} E\left[\hat{c}[\tau_1]\hat{c}[\tau_2]\right] &= E\left[\frac{1}{N^2}\sum_n x[n]x[n+\tau_1]\sum_m x[m]x[m+\tau_2]\right] \\ &= \frac{1}{N^2}\sum_{m,n} E\left[x[n]x[n+\tau_1]x[m]x[m+\tau_2]\right] \\ &= \frac{1}{N^2}\sum_{m,n} c[\tau_1]c[\tau_2] + \\ &\quad \frac{1}{N^2}\sum_{m,n} c[m-n]c[m+\tau_2-n-\tau_1] + \end{aligned}$$

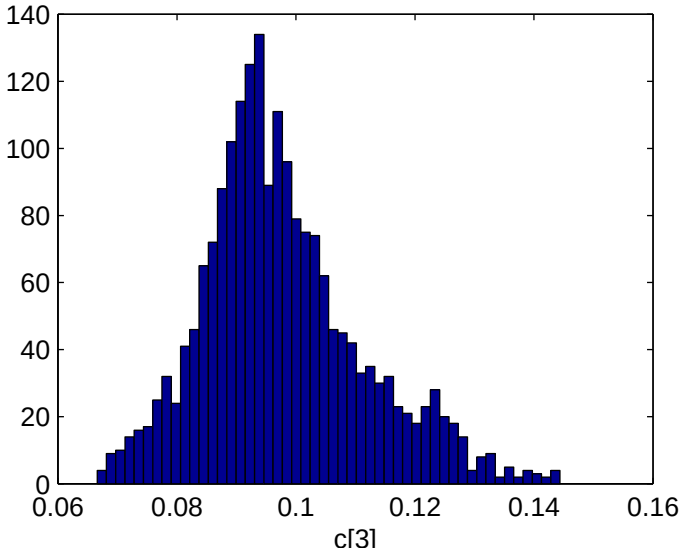

Figure 3.4: Histogram for $\hat{c}[3]$, for two seconds of the author saying `/AA/`

$$\begin{aligned} & \frac{1}{N^2}\sum_{m,n} c[m+\tau_2-n]c[m-n-\tau_1] \\ = \; & c[\tau_1]c[\tau_2] + \frac{1}{N}\sum_{m} c[m+\tau_1]\left(c[m+\tau_2)] + c[m-\tau_2]\right) \end{aligned} \tag{3.3}$$

That is, the variance of the stationary autocorrelation estimate contains the expected factor $c[\tau_1]c[\tau_2]$ and an additional structure imposed by the true autocorrelation of the signal.

Since $c[n]$ is a decreasing function of n, we can expect that the variance of the estimator $\hat{c}[n]$ decreases as the window size N increases. This is a well known result. As more data is used, the estimator becomes better. Increasing the window length without bound is not an option, because the $c[n]$ calculation expects the signal to be stationary within the window. The goal of having a low variance estimate is diametrically opposed to ensuring stationarity within the window.

Unfortunately, we are limited in using this particular estimator to a contiguous segment of speech, that is short enough to guarantee stationarity over the duration of the segment.

As mentioned in Section 3.3, the estimators $\hat{c}[\tau]$ are themselves random variables, whose properties will yield to a little bit of analysis.

The general method to form a model of a set of signals, is first to define the model, and then to find the maximum likelihood set of parameters for the model, given the training data. The maximum likelihood estimate is consistent and asymptotically efficient. This means that as the size of the training set increases, the parameter estimate is guaranteed to converge in probability to the true value of the parameter, and that the variance of the estimate approaches the Cramér-Rao lower bound.

If we assume a general Gaussian model for the feature set, its likelihood function and log likelihood function are given in Equation 3.4 and 3.5. The random variable $\mathbf{r}$ represents the stochastic feature set, and r_{in} is the nth sample from the ith member of the training set. The vector Θ is the mean of the distribution, and Λ is the inverse covariance matrix.

$$p_{\mathbf{r}|\theta}(\mathbf{R}|\Theta) = \prod_{i=0}^{N-1} \frac{1}{(2\pi)^{L/2}|\Lambda|^{1/2}} \exp\left(\frac{1}{2}\sum_{m=0}^{L-1}\sum_{n=0}^{L-1}(r_{im}-\theta_m)\lambda_{mn}(r_{in}-\theta_n)\right) \quad (3.4)$$

$$\ln p_{\mathbf{r}|\theta}(\mathbf{R}|\Theta) = \ln \frac{N}{(2\pi)^{L/2}|\Lambda|^{1/2}} - \frac{1}{2}\sum_{i=0}^{N-1}\sum_{m=0}^{L-1}\sum_{n=0}^{L-1}(r_{im}-\theta_m)\lambda_{mn}(r_{in}-\theta_n) \quad (3.5)$$

A maximum likelihood estimate of the model parameters given the training set is found by taking derivatives of Equation 3.5 with respect to the model parameters and equating to zero. For this Gaussian model, the well-known answer is

$$\hat{\Theta} = \frac{1}{N}\sum_{i=0}^{N-1}(r_i) \quad (3.6)$$

$$\hat{\Lambda}^{-1} = \frac{1}{N}\sum_{i=0}^{N-1}\left(r_i-\hat{\Theta}\right)\left(r_i-\hat{\Theta}\right)^T \quad (3.7)$$

In accordance with the maximum likelihood estimate, we average the feature $\hat{c}[\tau]$ over the set of training data to find the mean value for our model.

$$\mu[\tau] = \frac{1}{I}\sum_{i=0}^{I-1}\hat{c}^{(i)}[\tau]$$

The average outer product of our feature set is used to find the covariance of our model.

$$\Sigma[\tau_1, \tau_2] = \frac{1}{I}\sum_{i=0}^{I-1} \hat{c}^{(i)}[\tau_1]\hat{c}^{(i)}[\tau_2] - \mu[\tau_1]\mu[\tau_2]$$

This estimate is well known, and gives us an asymptotically unbiased estimate of the true mean and covariance of the feature set. But it is possible to have a better estimate than this that takes advantage of the structure imposed on the covariance matrix by the features, to reduce the degrees of freedom in the estimation problem.

Inspired by Equation 3.3, where the covariance of the autocorrelation features is a function of the autocorrelation of the signal, we can form an alternative estimate of the covariance matrix directly from the maximum likelihood estimate of the signal's autocorrelation.

The estimates $\mu[\tau] \approx c[\tau]$ are used to approximate the covariance matrix for our unimodal Gaussian model.

$$\begin{aligned}
\Sigma[\tau_1, \tau_2] &= \frac{1}{N}\sum_m c[m+\tau_1]\left(c[m+\tau_2] + c[m-\tau_2]\right) \\
&\approx \frac{1}{N}\sum_m \mu[m+\tau_1]\left(\mu[m+\tau_2] + \mu[m-\tau_2]\right)
\end{aligned}$$

The expected value of the estimate is given by

$$\begin{aligned}
& E\left[\frac{1}{N}\sum_m \mu[m+\tau_1]\left(\mu[m+\tau_2)] + \mu[m-\tau_2]\right)\right] \\
= \; & \frac{1}{N^3 I^2}\sum_{m,i,j,a,b} E\left[x^{(i)}[a]x^{(i)}[a+m+\tau_1]x^{(j)}[b]x^{(j)}[b+m+\tau_2]\right] + \\
& \frac{1}{N^3 I^2}\sum_{m,i,j,a,b} E\left[x^{(i)}[a]x^{(i)}[a+m+\tau_1]x^{(j)}[b]x^{(j)}[b+m-\tau_2]\right] \\
= \; & \frac{1}{N^3 I^2}\sum_{m,i,j,a,b} c[m+\tau_1]\left(c[m+\tau_2] + c[m-\tau_2]\right) + \\
& \frac{1}{N^3 I^2}\sum_{m,i,j,a,b} \delta[i-j]c[b-a]c[b-a+\tau_2-\tau_1] + \\
& \frac{1}{N^3 I^2}\sum_{m,i,j,a,b} \delta[i-j]c[b-a]c[b-a-\tau_2-\tau_1] +
\end{aligned}$$

$$
\begin{aligned}
& \frac{1}{N^3 I^2} \sum_{m,i,j,a,b} \delta[i-j] c[b-a+m+\tau_2] c[b-a+-m-\tau_1] + \\
& \frac{1}{N^3 I^2} \sum_{m,i,j,a,b} \delta[i-j] c[b-a+m-\tau_2] c[b-a+-m-\tau_1] \\
= \quad & \frac{1}{N} \sum_{m} c[m+\tau_1] \left(c[m+\tau_2] + c[m-\tau_2] \right) + \\
& \frac{1}{IN} \sum_{a,m} c[m] \left(c[m+\tau_2-\tau_1] + c[m-\tau_2-\tau_1] \right) + \\
& \frac{1}{IN} \sum_{a,m} c[a-m-\tau_1] \left(c[a+m+\tau_2] + c[a+m-\tau_2] \right) \\
= \quad & \frac{I+1}{I} \frac{1}{N} \sum_{m} c[m+\tau_1] \left(c[m+\tau_2] + c[m-\tau_2] \right) + \\
& \frac{1}{IN} \left(C^2[0] + 2C^2[N/2](-1)^{\tau_1+\tau_2} \right)
\end{aligned}
$$

Where $C[k]$ is the power spectrum of the signal, and $C[N/2]$ is defined as zero for odd signal lengths N.

$$C[k] = \sum_{\tau} c[\tau] \exp\left(\frac{2j\pi\tau k}{N} \right)$$

As $I \to \infty$, the expected value of the new, parametric, estimate approaches the correct value. Thus, it too is an asymptotically unbiased estimator. In practice, the total error in this estimate of the variance is lower than that given by the maximum likelihood approach. By incorporating knowledge of the structure of the autocorrelation matrix, we are able to dramatically improve upon the maximum likelihood estimate derived from a general Gaussian model.

Figure 3.5 shows how the parametric covariance estimate is closer to the true covariance estimate for limited training set size.

Data was generated by passing white Gaussian noise through a known fourth order autoregressive filter. For each training set size I, several estimates of the non-parametric and parametric estimate of the covariance were made. The Frobenius norm (Equation 3.8) between the true and estimated covariance matrices were calculated

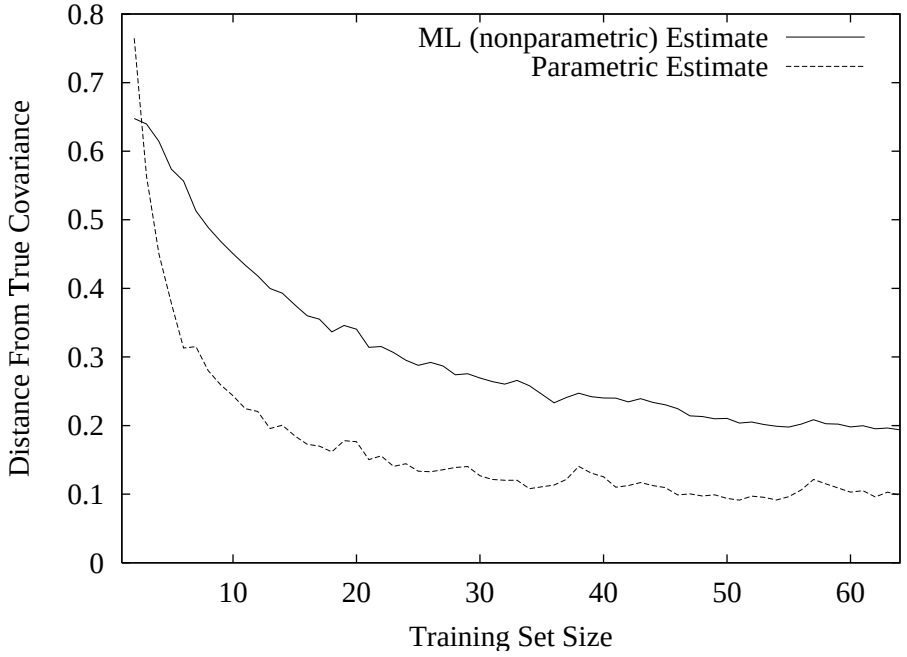

Figure 3.5: Frobenius norm between true and estimated autocorrelation covariance matrices

and averaged over each training set size.

$$D_F[A, B] = \left(\sum_{i,j} (a_{ij} - b_{ij})^2 \right)^{1/2} \tag{3.8}$$

There are two features of importance in Figure 3.5. First, and most noticeably, for $I > 2$, the parametric estimate is consistently closer to the true covariance matrix by up to a factor of two. This is not an artifact of the filter chosen to shape the spectrum of the data, but a general property exhibited by the parametric estimate.

The second important feature in Figure 3.5 is that the parametric estimate was able to perform on a training set with only one element. For this case, the non-parametric estimate can not be used, because it estimates the covariance to be a matrix full of zeros.

In general, The parametric estimate produces a full rank estimate of the covariance. For any given training set size $I > 1$, the non-parametric covariance estimate

is limited to a rank of at most I. This becomes a problem when trying to estimate covariances of features with thousands or millions of dimensions.

Figure 3.6 shows examples of parametric and non-parametric covariance estimates for one synthetic data set. The top row corresponds with the new, parametric, estimate with sixteen training examples. With this small amount of training data, a reasonably good estimate is achieved. The covariance has full rank, and the first three eigenvectors are smooth.

The next three rows in Figure 3.6 illustrate the non-parametric covariance estimate for training set sizes 16, 32, and 48. None of these covariance matrices have full rank, and the eigenvectors do not resemble those produced by the parametric estimate until at least three times the data is included in the training set. Even the final row, generated from 1000 training examples, is just approaching the quality of the parametric estimate.

3.3.3 Nonstationary Autocorrelation Estimate

Parallel to the stationary autocorrelation features, the auto-ambiguity plane can be interpreted as a nonstationary autocorrelation estimate. For samples where $\eta = 0$, we have the stationary autocorrelation estimate, and everywhere else we have the estimate of that portion of the autocorrelation that is being modulated at a rate of η.

Just as the covariance of the autocorrelation feature was a function of the true covariance of the signal, it turns out that the covariance of the auto-ambiguity feature is a function of the true auto-ambiguity of the signal.

In particular, the expected value of the autocorrelation of the auto-ambiguity function is

$$\begin{aligned}\Sigma[\eta_1,\tau_1,\eta_2,\tau_2] &= E\left[\left(\sum_n x[n]x^*[n+\tau_1]W_N^{\eta_1 n}\right)^*\left(\sum_m x[m]x^*[m+\tau_2]W_N^{\eta_2 m}\right)\right] \\ &= \sum_{m,n} E\left[x^*[n]x[n+\tau_1]x[m]x^*[m+\tau_2]\right]W_N^{\eta_2 m-\eta_1 n}\end{aligned}$$

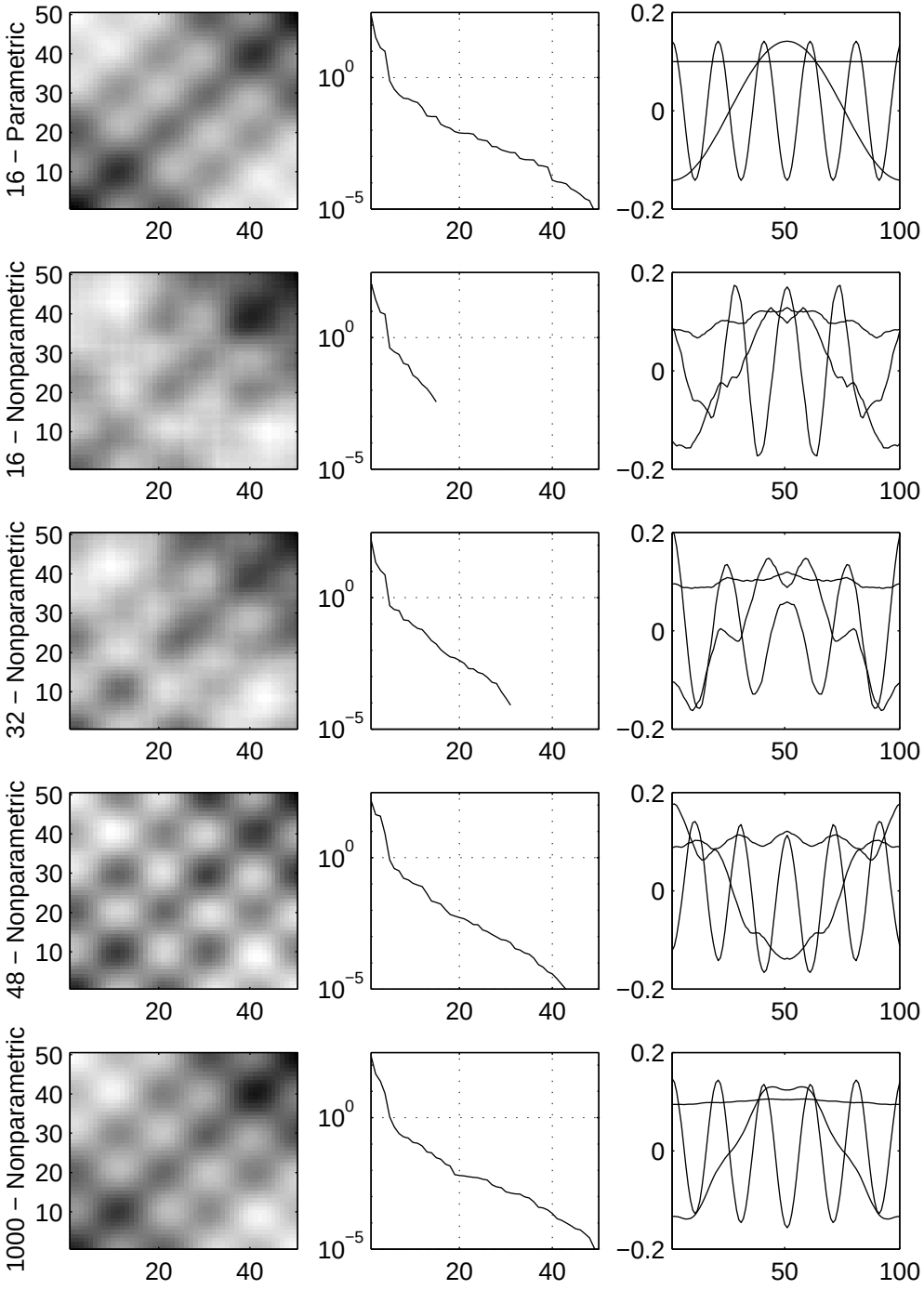

Figure 3.6: Comparison of parametric and non-parametric covariance estimation for autocorrelation features

$$
\begin{aligned}
= \quad & A^*[\eta_1, \tau_1] A[\eta_2, \tau_2] \\
& + \frac{1}{N} \sum_{a,n} A[a + \eta_2, n + \tau_2] A^*[a + \eta_1, n + \tau_1] W_N^{a\tau_2 - \eta_1 n} \\
& + \frac{1}{N} \sum_{a,n} A[a + \eta_2, n - \tau_2] A^*[a + \eta_1, n + \tau_1] W_N^{-a\tau_2 - \eta_1 n} W_N^{-\eta_2 \tau_2}
\end{aligned}
$$

We can therefore justify using the estimate of the mean of the feature to estimate the covariance of the feature. This gives us an unbiased estimate of the covariance of the auto-ambiguity function, with major advantages over the maximum likelihood estimate. The proof parallels that in the previous section, but is long and complex, and included in Appendix B.

The first advantage is that the parametric estimate of the covariance can achieve a large rank more quickly than the maximum likelihood estimate. If one has I examples in a training set, the maximum likelihood model will be doomed to live in an I dimensional subspace of the size N^2 feature space. The parametric model is not limited in this way, and tends to fill a large subspace with only a few training examples.

A second advantage that the parametric estimate has is a lower estimation error. Analogous to the autocorrelation case, this estimate is often much closer to the true covariance than the maximum likelihood estimate.

Chapter 4

DISTANCE METRICS FOR QUADRATIC TIME-FREQUENCY REPRESENTATIONS

This chapter addresses problems related to the sparsity of the discrete time-frequency representation. Although each real signal $x[n]$ has only N free parameters, its representation has N^2 elements, along $N(N+1)/2$ independent dimensions. This leads to problems in modeling, classification, and recognition tasks.

Traditional features do not suffer from any of these problems, having few enough dimensions so that a full coverage of the space is possible.

4.1 Introduction

It is not advisable to use standard models directly on the complete time-frequency representation. Techniques such as Gaussian mixture models, VQ, or k-nearest neighbor can break down when applied to very sparse data in a very high dimensional feature space. In general, there are three problems with this approach.

The first problem is data sparsity. For real signals with length N, the time-frequency representation consists of $N(N+1)/2$ linearly independent dimensions. To model this feature accurately using conventional methods, on the order of N^2 independent training examples would be needed, which is impractical.

The second problem is that not all points in the $N(N+1)/2$ dimensional feature space are valid time-frequency representations. In particular, it is shown that the set of all valid time-frequency representations is a continuous N dimensional surface embedded in the feature space. As a result, at every point in the feature space, a

distribution has zero extent in at least $N^2/2$ dimensions.

The third problem with modeling with traditional techniques is that the mean feature may not lie near any valid representations. That is, for a given set of data, the mean of the data does not correspond with an actual signal. In other words, the mapping from the representation to the signal is overdetermined.

As an example of the problems that may be encountered, consider the case of a set of three dimensional data restricted to lie on the surface of a cone. While it may take many mixtures of full covariance Gaussians to model the data accurately, there is a simple underlying structure that should be exploited.

Section 4.2 describes what can be expected of the set of valid time-frequency representations. The dimensionality of the representations is explored, together with the gross shape of the set.

Section 4.3 continues the thread by introducing a new distance metric that leverages the information from Section 4.2.

Section 4.4 illustrates these concepts in a low-dimensional signal space, where they are easy to picture.

Section 4.5 shows that classifiers based on this new distance metric can improve accuracy over blindly training models in the representation space.

4.2 The Set of Valid Time-Frequency Representations

Consider the discrete-time signal $x[n]$, with length N. The quadratic time-frequency representation $A[\eta, k]$ associated with $x[n]$ consists of N^2 complex numbers, and is contained in the set of length N^2 complex vectors.

$$A[\eta, \tau] = \sum_n x^*[n]x[n+\tau] \exp\left(2\pi\eta n/N\right).$$

This, however, is somewhat misleading as to the true size of the feature space. A discrete Fourier transform from η into n yields,

$$A[n, \tau] = x^*[n]x[n+\tau].$$

Clearly, there are $N(N-1)/2$ unique products when $\tau \neq 0$, and N magnitudes when $\tau = 0$. For complex signals, this yields a feature space with N^2 linearly independent dimensions, and for real signals, the feature space has $N(N+1)/2$ linearly independent dimensions.

The mapping between the signal $x[n]$ and its representation $A[n,k]$ is continuous, and therefore the set of representations that correspond with actual signals forms a continuous subspace.

For real signals with $N = 2$, the set of valid time-frequency representations is a 90 degree cone embedded in a four-dimensional space, with extent only in three dimensions. This agrees with the predicted value, since the expected dimensionality of the cone is $2(2+1)/2 = 3$. This case is examined in detail in Section 4.4.

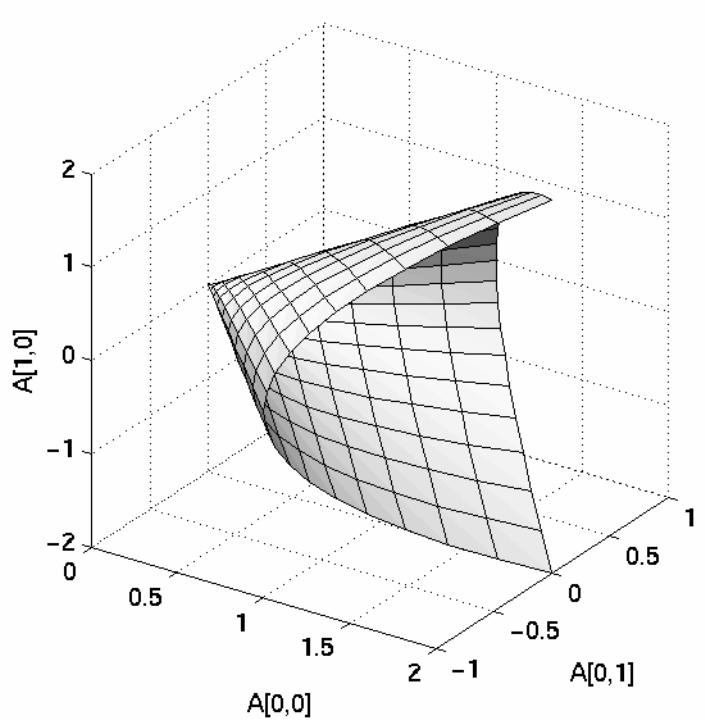

Figure 4.1: Conical structure of valid TFR

For larger values of N, the surface can not be named, but it can be functionally

described. The set of valid discrete time-frequency representations always lies on a conical structure, regardless of N. The axis of the cone corresponds to the energy of the signal $x[n]$, which is stored at the origin of the (η, τ) plane, $A[0,0]$. For a fixed energy, the valid representations lie on a hyper-sphere with a radius proportional to $A[0,0]$.

$$\begin{aligned} \sum_{m=0}^{L-1}\sum_{n=0}^{L-1} |A[m,n]|^2 &= \sum_{m=0}^{L-1}\sum_{n=0}^{L-1} |x[m]x^*[n]|^2 \\ &= \sum_{m=0}^{L-1} |x[m]|^2 \sum_{n=0}^{L-1} |x[n]|^2 \\ &= |A[0,0]|^2 \end{aligned} \tag{4.1}$$

Although all valid representations lie on this conical structure, not all points on this structure are valid representations. That is, meeting Equation 4.1 is a necessary, but not sufficient test for valid time-frequency representations.

4.3 A Geodesic Distance Measure

Measuring the distance between New York and San Francisco by drawing a straight line through three dimensional space would not seem to be an acceptable measure. Similarly, measuring the distance along a line connecting two valid representations, but lying entirely outside of the surface of valid representations, may not be the best solution. This is exactly what is going on with a Euclidean distance metric in the time frequency representation space.

The structure of the surface of valid time frequency representations, along with the calculus of variations, allows us to find the shortest line connecting two representations, that remains entirely within the set of representations of true signals. This line is a Geodesic, and its length is the Geodesic distance between the endpoints.

This line can be used to morph one signal into another, along the shortest path in quadratic time-frequency space. That is, given two signals, it specifies a continuous

path along which one signal slowly changes into the other. This idea is explored in Section 7.1. The line can also be used to find the "true" distance between distributions, which would not necessarily be a straight line in the feature space.

Without explicitly solving the geodesic problem, approximate solutions can be found. Two solutions for the geodesic distance are presented, an algorithmic approximation and a numerical approximation.

4.3.1 Algorithmic Approximation

To find the shortest distance between two points, lying entirely along valid time-frequency representations, an initial solution is posed, and then iteratively re-estimated.

The initial estimate is to form a set of points equally spaced in the signal space, with the proper endpoints. If one endpoint is the signal $a[n]$, and the second endpoint is $b[n]$, the initial set of points in the signal space would be,

$$x_i[n] = a[n] + (b[n] - a[n])\frac{i}{M-1}, \text{ for } 0 \leq i < M.$$

The re-estimation assumes that links between neighboring signals are elastic, and tries to pull each signal towards its neighbors in the time-frequency representation space.

First, a direction is computed that will move a signal closer to its neighbors in the time-frequency representation space. If the representation for $y[n]$ moves in this direction, the distance to the representations $x[n]$ and $z[n]$ tends to decrease.

$$v[n,\tau] = \frac{1}{2}(x[n]x[\tau] + z[n]z[\tau]) - y[n]y[\tau]$$

This vector is then mapped to an equivalent direction in signal space. The derivatives of the representation with respect to the original signal is,

$$\frac{d}{y[a]}y[n]y[\tau] = \delta[n-a]y[\tau] + \delta[\tau-a]y[n]$$

Now, this is a scalar-valued function of n, τ, and a. Unwrap it into a matrix, whose columns are a, and whose rows consist of all combinations of n and τ. Call

this matrix V. Its columns represent the directions the representation will move with changes in the signal values.

Now, all that remains is mapping the desired vector v, which exists in the representation space, onto the available directions contained in V. This is a simple matrix operation, and consists of the minimum square solution to the linear equation,

$$\begin{aligned} Vd &= v \\ d &= (V^H V)^{-1} V^H v \end{aligned}$$

If the signal y is moved in the direction d, the distances between the representations for y and z, and y and x, is reduced. If the representations of x, y, and z are sufficiently close, the linear distance and the geodesic distances should be approximately equal.

4.3.2 Numerical Approximation

The numerical approximation to the geodesic distance consists of first defining a path in the representation space, and then precisely computing its length.

The surface of valid time-frequency representations consists of N^2 dimensions y_{ij}, where $0 \leq \{i, j\} < N$. This surface is parameterized in terms of the independent variables x_i, where $0 \leq i < N$, which are the discrete signal samples.

$$y_{ij} = x_i x_j$$

The first step is to assume the signal values x_i are functions of a single independent variable, t. As t varies from 0 to 1, a path is traced through both the signal space and the representation space.

Let $\hat{u}_{ij}$ be a unit vector in the N^2 time-frequency representation space such that $\langle \hat{u}_{ab}, \hat{u}_{cd} \rangle = \delta[a-c]\delta[b-d]$. The derivative of the path with respect to the independent variable is given by

$$\frac{d}{dt}\hat{s} = \sum_{i,j} \frac{d}{dt} y_{ij} \hat{u}_{ij}$$

$$
\begin{aligned}
&= \sum_{i,j} \frac{d}{dt} x_i(t) x_j(t) \hat{u}_{ij} \\
&= \sum_{i,j} \left(x_j(t) \dot{x}_i(t) + x_i(t) \dot{x}_j(t) \right) \hat{u}_{ij}
\end{aligned}
$$

To find the distance along the path, integrate the function $f(t)$, where

$$
\begin{aligned}
f(t) &= \left| \frac{d\hat{s}}{dt} \right| \\
&= \left(\sum_{i,j} \left(x_j(t) \dot{x}_i(t) + x_i(t) \dot{x}_j(t) \right)^2 \right)^{\frac{1}{2}}
\end{aligned}
$$

In many dimensions, it is difficult to solve this equation exactly for the functions $x_i(t)$, $0 \leq i < N$, but an approximate solution can be found by assuming a linear form for the functions. The slope and offset for each function are entirely specified by the initial value, $x_i(0)$, and the final value, $x_i(1)$.

$$
\begin{aligned}
x_i(t) &= x_i(0) + \left(x_i(1) - x_i(0) \right) t \\
\dot{x}_i(t) &= x_i(1) - x_i(0)
\end{aligned}
$$

Once the endpoints for the calculation are known, the function $f(t)$ takes the form of a square root of a quadratic function. This can be quickly estimated with great precision by calculating the integral

$$
d = \int_0^1 f(t) dt.
$$

This approximation is equivalent to the starting point of the iterative solution presented previously. Although this approximation is rather gross, in practice the differences between the two solutions do not amount to much. That is, if you draw a straight line between two signals in the signal space, the corresponding line in the time-frequency representation space is close to optimal.

4.4 Length Two Signals

This section deals with the case of real signals with $N = 2$. As mentioned previously, the set of valid representations for this case is a three dimensional cone embedded in

a four dimensional space.

For this special case, an exact solution for the geodesic path has been found using the calculus of variations.

For convenience, let a and b represent the two samples of the signal $x[n]$. In the (η, τ) plane, this signal's representation only has three non-zero values.

$$A[0,1] = 2ab \qquad A[1,1] = 0$$
$$A[0,0] = a^2 + b^2 \quad A[1,0] = a^2 - b^2$$

A simple change of variables reveals the traditional formula for a cone,

$$x = A[0,1] = u\sin v \quad y = A[1,0] = u\cos v \quad z = A[0,0] = u, \text{ where } u \geq 0$$

4.4.1 Geodesic Distance on a Cone

To determine the distance between two representations, we measure the shortest line along the surface that joins the two points. This is exactly the type of problem that the calculus of variations is meant to solve.

The problem is posed as one of minimizing the line integral between the two points. The calculus of variations tells us that the geodesic function $v(u)$ is given by

$$v = c_1 \int \frac{\sqrt{P}}{\sqrt{R(R - c_1^2)}} du,$$

where

$$P = \left(\frac{\delta}{\delta u}x\right)^2 + \left(\frac{\delta}{\delta u}y\right)^2 + \left(\frac{\delta}{\delta u}z\right)^2$$
$$Q = \left(\frac{\delta}{\delta u}x\right)\left(\frac{\delta}{\delta v}x\right) + \left(\frac{\delta}{\delta u}y\right)\left(\frac{\delta}{\delta v}y\right) + \left(\frac{\delta}{\delta u}z\right)\left(\frac{\delta}{\delta v}z\right)$$
$$R = \left(\frac{\delta}{\delta v}x\right)^2 + \left(\frac{\delta}{\delta v}y\right)^2 + \left(\frac{\delta}{\delta v}z\right)^2$$

and

$$\tfrac{\delta}{\delta u}x(u,v) = \sin(v) \qquad \tfrac{\delta}{\delta u}y(u,v) = \cos(v) \qquad \tfrac{\delta}{\delta u}z(u,v) = 1$$
$$\tfrac{\delta}{\delta v}x(u,v) = u\cos(v) \quad \tfrac{\delta}{\delta v}y(u,v) = -u\sin(v) \quad \tfrac{\delta}{\delta v}z(u,v) = 0$$

so

$$P = 2 \quad Q = 0 \quad R = u^2$$

Combining these equations, the result is

$$\begin{aligned} v &= c_1 \int \frac{\sqrt{2}}{\sqrt{(u^2)(u^2 - c_1^2)}} du \\ &= c_1 \sqrt{2} \int \frac{1}{u\sqrt{u^2 - c_1^2}} du \\ v &= \sqrt{2} \cos^{-1} \left| \frac{c_1}{u} \right| + c_2 \end{aligned}$$

The constants c_1 and c_2 are dependent on the desired endpoints of the geodesic, (u_1, v_1) and (u_2, v_2).

$$\begin{aligned} c_1 &= \pm u \cos(\frac{v - c_2}{\sqrt{2}}) \\ c_2 &= v - \sqrt{2} \cos^{-1} \left| \frac{c_1}{u} \right| \\ c_2 &= -\sqrt{2} \tan^{-1} \left(\frac{u_1 \cos \frac{v_1}{\sqrt{2}} - u_2 \cos \frac{v_2}{\sqrt{2}}}{u_1 \sin \frac{v_1}{\sqrt{2}} - u_2 \sin \frac{v_2}{\sqrt{2}}} \right) \end{aligned}$$

4.5 Classification Improvement

Figure 4.2 shows the performance of three distance metrics for classification in additive white Gaussian noise. The "L2-Signal" metric is equivalent to the matched filter solution, and serves as an upper bound on performance. The "Geodesic Approximation" corresponds to a classifier using the numerical approximation to the geodesic distance. The "L2-Representation" classifier uses the distance from the mean in representation space.

The geodesic approximation does much better than computing distances directly in the representation space. It comes close to being the optimal detector, although it is just an approximation to the true geodesic distance.

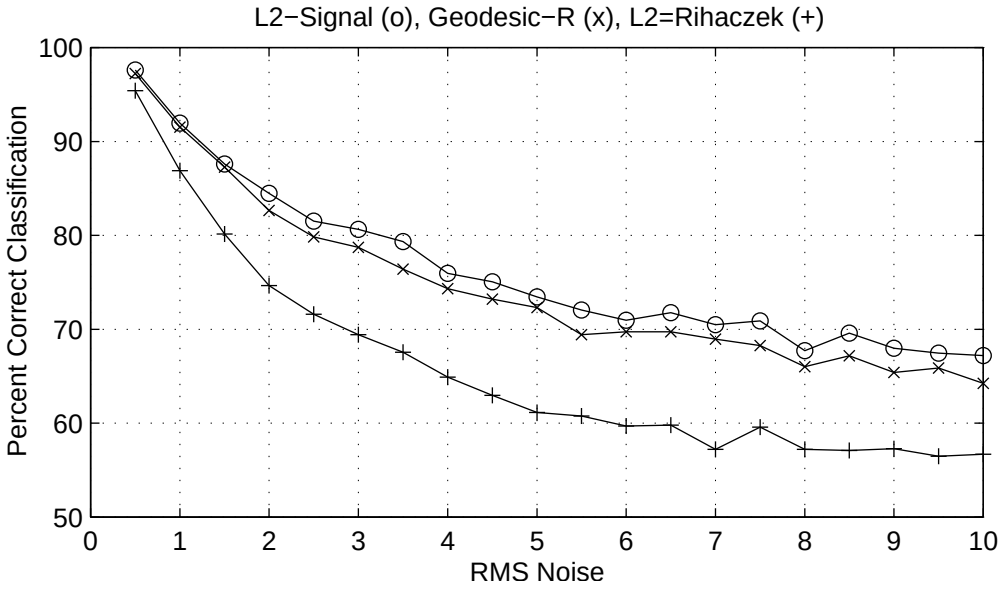

Figure 4.2: Classification in the presence of additive noise

4.6 Least Square Representation Inversion

Of course the geodesic measure only works when the reference corresponds to a known signal. If we just form an average of the training data, it will almost certainly not fall within the proper subspace.

Finding the centroid of real data is a problem. The mean of the training data is almost guaranteed not to lie on a valid signal. As a result, attempts at modeling distributions around this mean value are intrinsically flawed. Additionally, to use the geodesic distance measure, both endpoints of the distance must be valid representations.

One route around this problem is to compute the arithmetic mean of the features, and then substitute a mean value that does exist and is close to the arithmetic mean.

Define the distance between the arithmetic mean Q and the closest valid repre-

sentation to be the error, E.

$$E = \sum_{k,\eta} |X[k]X^*[k+\eta] - Q[k,\eta]|^2 \tag{4.2}$$

Now, expand E in terms of the real and imaginary parts of Q and X.

$$E = \sum_{k,\eta} \left(\begin{array}{l} \left(X_r[k]X_r[k+\eta] + X_i[k]X_i[k+\eta] - Q_r[k,\eta]\right)^2 \\ + \left(X_i[k]X_r[k+\eta] - X_r[k]X_i[k+\eta] - Q_i[k,\eta]\right)^2 \end{array} \right) \tag{4.3}$$

To find the X that gives minimum error, differentiate E with respect to the real and imaginary parts of X.

$$\frac{\delta E}{\delta X_r[k]} = 2 \sum_{\eta} \left(\begin{array}{l} +X_r[k]\left(X_r^2[k+\eta] + X_i^2[k+\eta]\right) \\ -X_r[k+\eta]Q_r[k,\eta] + X_i[k+\eta]Q_i[k,\eta] \\ +X_r[k-\eta]\left(X_r^2[k] + X_i^2[k]\right) \\ -X_r[k]Q_r[k-\eta,\eta] + X_i[k]Q_i[k-\eta,\eta] \end{array} \right) \tag{4.4}$$

$$\frac{\delta E}{\delta X_i[k]} = 2 \sum_{\eta} \left(\begin{array}{l} X_i[k]\left(X_r^2[k+\eta] + X_i^2[k+\eta]\right) \\ -X_i[k+\eta]Q_r[k,\eta] - X_r[k+\eta]Q_i[k,\eta] \\ X_i[k-\eta]\left(X_r^2[k] + X_i^2[k]\right) \\ -X_i[k]Q_r[k-\eta,\eta] - X_r[k]Q_i[k-\eta,\eta] \end{array} \right) \tag{4.5}$$

These two equations can be combined, giving the function,

$$\frac{\delta E}{\delta X[k]} = 2 \sum_{\eta} \left(\begin{array}{l} +X[k]\left|X[k+\eta]\right|^2 - X[k+\eta]Q[k,\eta] \\ +X[k-\eta]\left|X[k]\right|^2 - X[k]Q[k-\eta,\eta]. \end{array} \right) \tag{4.6}$$

Now, traditional gradient descent techniques can be used to find the solution.

Chapter 5

CLASS DEPENDENT TIME-FREQUENCY REPRESENTATIONS

Class dependent time-frequency representations combine the rich theory of truly discrete, quadratic, time-frequency representations presented in Chapters 2 and 3 with a discriminative approach to kernel design.

Traditionally, kernel design and selection has been guided by the desire to have the resulting time-frequency representation (TFR) satisfy one or more established properties each of which is motivated by certain physical or mathematical considerations[10]. For example, we may wish for our TFR to be consistent, in some sense, with the time series and/or the Fourier transform of the time series. We may want our TFR to have the characteristics of an energy distribution. Or we may want our TFR to be shift-invariant, *e.g.* if we shift the time series, we observe a similar shift in the TFR. Not all TFR kernels will preserve all of these properties, but what all previously-proposed kernels lack are properties which relate directly to an end objective of signal classification.

In contrast, the design goal for a class dependent kernel is maximum separability of the classes, and higher classification accuracy. Classification performance becomes the primary concern in the kernel design procedure.

If TFRs such as those alluded to above are used, we have to hope that if our signal is well represented, then it will be well classified. Though this seems like a reasonable proposition, a representation may well bear a great deal of information which *all* signals under study share as well as information unique to each individual

example signal. Such information is irrelevant to the classification, but most past approaches leave it up to a subsequent classifier to "screen out" these details.

The class dependent kernel method isolates that portion of the time-frequency structure that is useful for classification. That is, when possible, a closed-form solution is derived for an optimally discriminating kernel. This signal is not signal dependent, but *signal class* dependent.

Other laboratories are using this idea to generate time-frequency representations, and excitement is building as experiments continue to show promising results. Unfortunately, those few methods that propose to optimize the kernel for classification constrain the form of the kernel to Gaussian functions with symmetries which may not be germane to detection or classification [15, 22].

This chapter presents an overview of the commonalities among different methods which have been tested. Following this is one section for each method, starting with the most basic L_2 inspired method, and concluding with an information-theoretic measure for kernel design.

5.1 Class Dependent Kernel Method

Given a labeled set of data from several classes, we seek to generate a kernel function that emphasizes classification relevant details present in the representation.

Given a discrete-time signal $x[n]$, the general discrete time-frequency representation is

$$M[\eta, \tau] = \phi[\eta, \tau] \sum_n x^*[n] x[n + \tau] \exp(\frac{j2\pi n\eta}{N}). \tag{5.1}$$

In previous chapters, an information preserving kernel, $\phi[\eta, \tau] = 1$ was chosen. The question of how best to deal with these high-resolution features was addressed. In this chapter, the design freedom inherent in the kernel formulation is leveraged to reduce the dimensionality of the representation, without degrading classification performance.

5.1.1 Kernel Design Intuition

One can get an intuition about kernel design by studying the properties of traditional kernels. Among the set of traditional non-information preserving kernel functions are the spectrogram kernel and the cone kernel.

The spectrogram kernel is dependent on a window function $w[n]$. The kernel is the conjugate of the base TFR of the window function.

$$\phi[\eta, \tau] = \sum_n w[n] w^*[n + \tau] \exp(\frac{-j2\pi n\eta}{N})$$

A plot of this kernel for two different windows is shown in Figure 5.1. In general, this is a lowpass kernel. That is, it tends to concentrate its energy near the origin, and has limited extent in either η or τ.

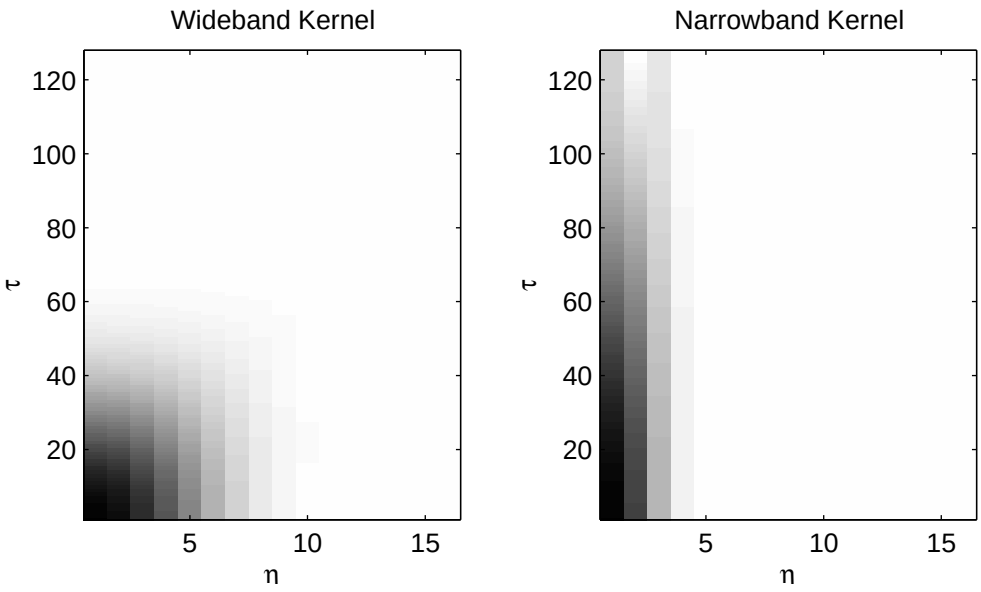

Figure 5.1: Spectrogram kernels

Furthermore, the extent in η is proportional to the inverse of the extent in τ. This tradeoff is the infamous time-frequency resolution tradeoff inherent in the spectrogram and is governed by the choice of window functions.

A larger extent in τ corresponds to a narrow-band spectrogram. Fine frequency resolution is achieved, at the expense of the time resolution.

Conversely, a larger extent in η corresponds to a narrow-band spectrogram. Fine time resolution results, but frequency resolution is sacrificed.

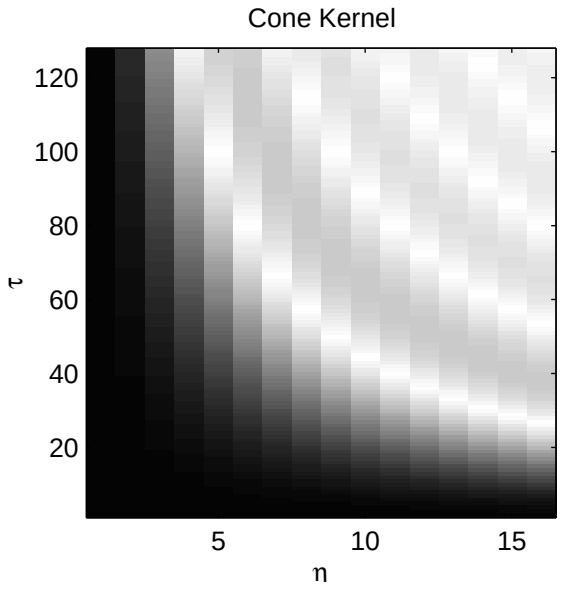

Figure 5.2: Cone (Zhao-Atlas-Marks) kernel

The cone kernel[56], Figure 5.2, was formulated to overcome some of the problems in choosing a global time-frequency tradeoff. The extent in η is a function of the extent in τ. This means that features that rely on the fine time structure of the signal are estimated with greater precision in time, and features that correspond with longer lags are averaged over more data.

The cone kernel shares one important property with the spectrogram kernel. It, too, is lowpass. In fact, the great majority of kernel design procedures produce kernels that are lowpass. In retrospect, this makes sense. The kernel is smoothing the representation. The information you would want to keep is the lowpass, gross structure of the representation, and not the fine structure, which may contain noise.

Another feature common to the cone kernel and spectrogram kernel is that neither one is invertible.

5.2 Kernel Design With Euclidean Distances

Our first attempt focused on finding the kernel that maximally separates the representations $A_1[\eta, \tau]$ and $A_2[\eta, \tau]$. We did this by maximizing the L_2 norm between the two representations. In the $(\eta \times \tau)$ plane, this goal simplifies to

$$\hat{\phi}[\eta, \tau] = \arg\max_{\phi[\eta,\tau]} \sum_{\eta,\tau} \left|\phi[\eta, \tau] \left(A_1[\eta, \tau] - A_2[\eta, \tau]\right)\right|^2 . \tag{5.2}$$

For Equation 5.2 to have a unique solution, the kernel is constrained to have unit energy.

$$\sum_{\eta,\tau} \left|\phi[\eta, \tau]\right|^2 = 1 \tag{5.3}$$

In this case, the problem collapses to that of maximizing a real symmetric quadratic form subject to the geometric constraint 5.3. The eigenvectors of this problem are

$$\phi^{(i,j)}[\eta, \tau] = \delta[\eta - i]\delta[\tau - j],$$

with associated eigenvalues

$$\lambda^{(i,j)} = \left|A_1[i, j] - A_2[i, j]\right|^2 .$$

The best kernel is the eigenvector that is associated with the maximal eigenvalue. That is, the kernel consists of one nonzero point, at the location where A_1 and A_2 are maximally separated.

Extensions have been proposed to this simple technique to make it more robust and to discriminate two classes of signals instead of a single pair of signals. To make it more robust, we tried choosing not just the single point where A_1 and A_2 were maximally separated, but also the second or third top contenders. To handle classes of signals, the mean of each class was substituted for the representations A_1 and A_2.

Both of these ideas were applied to data in [2]. The data was acoustic recordings of a hammer tap on a glass bottle, suspended by a string, and partially filled with water. The data was meant to simulate one type of underwater transient. The kernel produced chose only points along the $\eta = 0$ axis, indicating that spectral information alone was important for classification.

Although neither of the practical extensions extensions were theoretically pleasing, we were able to capture one key idea from this method. Instead of classifying on the entire time-frequency representation, we could do the classification in a subspace of the entire $(\eta \times \tau)$ plane. This subspace could be chosen so that it was good for discrimination.

5.3 Kernel Design With Fisher's Discriminant Metric

The design goal of the class dependent kernel became to find a subspace of the entire time-frequency representation space, in which the distribution of the training data is suitable for classification. Given a set of training examples $\{A_i^{(c)}\}$, indicating training example number i from class c, this subspace is identified with a discrimination metric.

The classifier defined by Equation 5.2 implicitly chooses its subspace using the discrimination metric

$$D[\eta,\tau] = \left| \frac{1}{N_1} \sum_{i=0}^{N_1-1} A_i^{(1)}[\eta,\tau] - \frac{1}{N_2} \sum_{i=0}^{N_2-1} A_i^{(2)}[\eta,\tau] \right|^2 . \tag{5.4}$$

In its place, a linear Fisher's discriminant can be used to identify dimensions suitable for classification.[2, 34].

$$D[\eta,\tau] = \frac{\sum_{c=0}^{C-1} \sum_{d=c+1}^{C-1} \left| \frac{1}{N_c} \sum_{i=0}^{N_c-1} A_i^{(c)}[\eta,\tau] - \frac{1}{N_d} \sum_{j=0}^{N_d-1} A_j^{(d)}[\eta,\tau] \right|^2}{\sum_{c=0}^{C-1} \left(\frac{1}{N_c} \sum_{i=0}^{N_c-1} \left| A_i^{(c)}[\eta,\tau] \right|^2 - \left| \frac{1}{N_c} \sum_{i=0}^{N_c-1} A_i^{(c)}[\eta,\tau] \right|^2 \right)} \tag{5.5}$$

The numerator of Equation 5.5 is similar to Equation 5.4, and measures the variance of the data between classes. The denominator of Equation 5.5 measures the

variance of the data within each class. Large discriminant values indicate dimensions along which the training data has a large variance between classes, and a small variance within each class.

Once a task-dependent subspace has been defined, a simple classifier can be built within that subspace.

This approach has been successfully applied to radar transmitter identification [18], helicopter gear box fault classification, and speech phoneme discrimination [16].

For N classes, and for each unique (η, τ) point, a Fisher's linear discriminant is computed. This computation is based on the mean, $\mu_i^{(\eta,\tau)}$, and variance, $\sigma_i^{(\eta,\tau)}$, of each class i and each coordinate (η, τ).

$$D[\eta, \tau] = \frac{\sum_{i=1}^{N_c} \sum_{j=i+1}^{N_c} \left(\mu_i^{(\eta,\tau)} - \mu_j^{(\eta,\tau)}\right)^2}{\sum_{i=1}^{N_c} \left(\sigma_i^{(\eta,\tau)}\right)^2} \tag{5.6}$$

This $D[\eta, \tau]$ value is then used to rank-order the points, from highest to lowest. In theory, large values will occur at coordinates where classes have both a large separation in the mean and a low within-class variance.

As more points are used, the classes are modeled in a larger space. As a result, they are allowed to become further apart. Unfortunately, as more points are added, the distributions of the classes are allowed to overlap more and more. With the first few points, the former effect dominates. As more points are added, the rate at which new information arrives decreases.

The former effect dominates as the first few dimensions are added, and then the latter effect dominates as more points are added. The key here is to add dimensions that increase the separation of the data, while minimizing the overlap, or variance, of the classes.

A kernel function can be specified in either plane. $\Phi[n, k]$ operates convolutionally on P_R in the time-frequency plane, and $\phi[\eta, \tau]$ operates as a multiplicative mask on $A[\eta, \tau]$ in the auto-ambiguity plane.

5.4 Discussion

5.4.1 Problems with Fisher's discriminant metric

One problem observed in the use of the Fisher's discriminant metric, is that it does not fully capture the merit of the individual dimensions chosen. Figure 5.3 illustrates the probability of error of choosing between two Gaussian random variables. The first is assumed to have zero mean and unit variance, without a loss of generality because if this were not the case, a linear scale and time shift would make it so. The figure shows the probability of error, given that the observation was equally likely to have come from either source, and that the second random variable has a mean μ_2 and a variance σ_2.

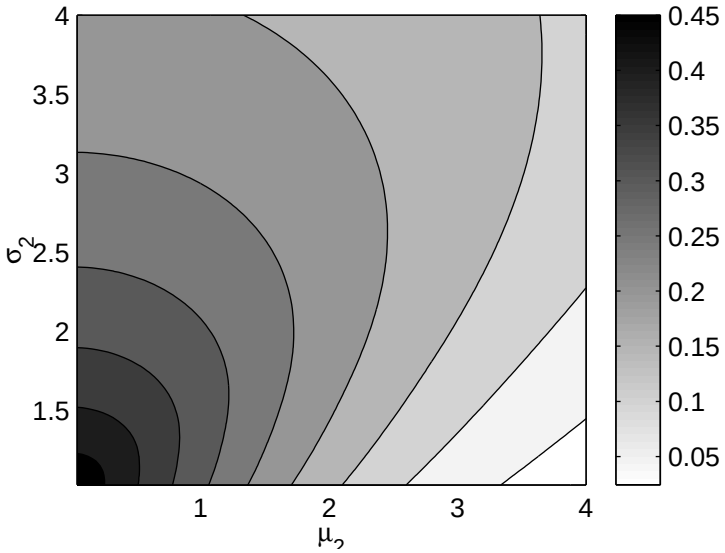

Figure 5.3: Pairwise Gaussian probability of error

This figure can be interpreted as one of merit for choosing the (x,y) pair to discriminate the classes. Regions of low probability are to be preferred over regions of higher probability.

If all pairs of variables fall such that $\mu > \sigma$, the Fishers metric will rank them

similarly to using Figure 5.3. However, the two metrics disagree in the region $\mu < \sigma$. In particular, two random variables with the same mean can be more discriminable than two whose means differ. The Fisher's discriminant metric ignores this effect.

5.4.2 Correlation Modeling

The feature correlation estimation techniques of the previous chapter can be directly applied. Instead of finding the sample covariance, the covariance model of the features is used. For reasonable training set sizes, this should result in a more accurate estimation of the covariance, as seen in the Chapter 3.

Chapter 6

EXPERIMENTAL RESULTS

This chapter illustrates how the theoretical results presented earlier can be applied to acoustic signal classification in general, and to speech recognition in particular.

6.1 Introduction

Three tasks are presented, with each new task being more difficult than the last.

Section 6.2 focuses on underwater transient identification. Although it was originally believed that it would be difficult to classify the chosen data, it turns out that most reasonable classifier designs yield 100 percent accuracy. It does still provide an excellent example of the class dependent kernel design procedure. The automatically designed kernel is easily interpretable at passing only those structures in the data that improve classification.

In Section 6.3, some results are presented on isolated phone recognition. The classifier's task is to discriminate among several sets of phones, extracted from continuously spoken English. The time-frequency features do not perform as well as traditional MFCC features, but when they are combined, the overall classification rate is better than either system alone.

Section 6.4 extends this result to classification of spoken letters taken from the American English alphabet. This task is more difficult than isolated phone recognition, because phone boundaries are unknown in the test set, and a more sophisticated method of combining the features must be employed.

6.2 Underwater Transient Identification

The first tests of our time-frequency representation in a classification system was with simulated underwater transients. The data was intended to represent short duration, transient-like passive sonar acoustic events, with both time and frequency structure.

Since the data was generated under controlled conditions, we were able to experiment with several distinct, well labeled classes.

6.2.1 The Data

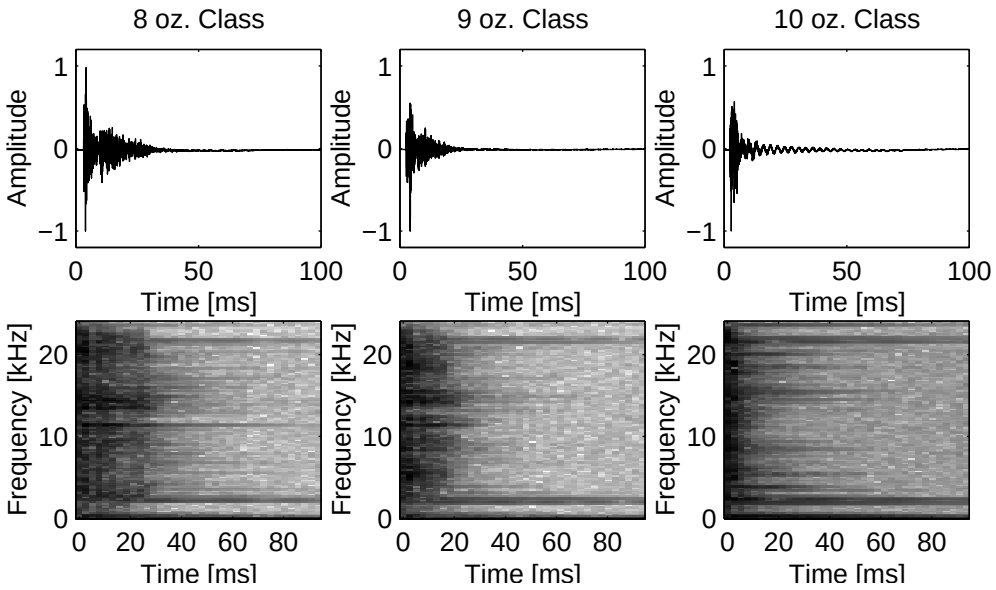

Figure 6.1: One example from each class, time series and spectrogram.

Figure 6.1 shows three typical time series from our data set. Each signal was generated by suspending a 12-ounce glass bottle from the ceiling, and tapping it with a small hammer. Several classes of data were generated by filling the bottle with water in thirteen steps, adding about one ounce at each step. As the water level increased, it was hoped that the changing resonance and impulse response of the bottle would make classification possible.

Many signals from each class were recorded on a digital audio tape. Later, the data was converted to analog and re-digitized at a sampling rate of 48 kHz. Thirty consecutive seconds of data from each class was digitized. All complete transients from each class were extracted with a 100ms window. This window was placed to contain 10ms of relative silence followed by the entire transient.

When played at the original sampling rate of 48 kHz, it is possible for a human listener to make rough classification estimates. When played at a rate of 8 kHz, a human listener can discriminate between adjacent water levels (classes), but not with complete accuracy.

Three data sets were chosen for automatic classification, representing approximately eight, nine, and ten ounces of water in the bottle. Sixteen examples were available for each class. Half were reserved for testing data, and half were used as training data.

It was believed that these three classes would pose a significant problem for the classification algorithm.

6.2.2 The Classifiers

Two classifiers were designed. The first classifier was based on the method of linear predictive coefficients (LPC) or, equivalently, an autoregressive model. The second classifier was based on our class dependent kernel, described previously.

6.2.3 LPC Method

LPC are good at capturing the spectral content of a signal, but ignore any time-domain information that may be present. LPC assume that the observed signal was generated by passing either white noise or a periodic impulse train through a purely recursive discrete-time linear filter. The LPC are an approximation of the coefficients of this filter.

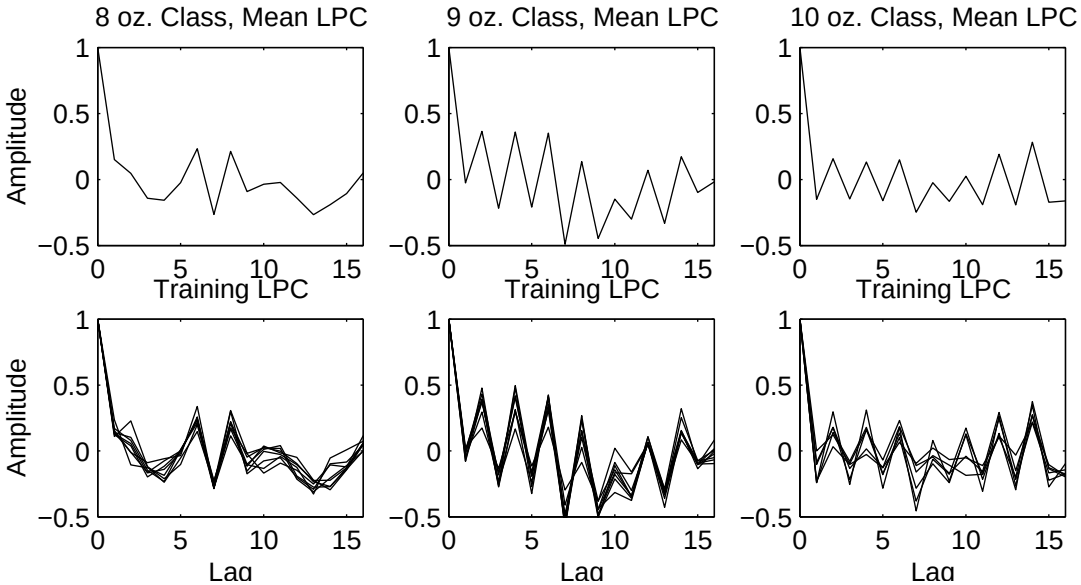

Figure 6.2: Linear predictive coefficients (AR(15))

For each training signal in each class, the first sixteen LPC were generated. Each class is then represented by its average LPC vector. Figure 6.2 shows the mean LPC vector for each of the three classes, along with plots of the vectors used for training.

Each signal in the test set was classified into one of the three classes by forming a new LPC feature, and determining which class mean was closest (in the Euclidean sense) to the test data.

The LPC classifier was able to perfectly classify all of the test data. We interpret this result as indicating that time-domain information was not relevant in the task of discriminating among our chosen classes. The spectral information is sufficient for classification in this way.

6.2.4 Class Dependent Kernel Method

Theoretically, our class dependent kernel should be able to find any time-variant behavior in the signals important for classification. The LPC contain only time-

invariant spectral information, and do not have this luxury.

A kernel was generated to maximally separate the classes from one another. That is, we would expect our class-dependent kernel to generate distributions for each class which are far apart according to our Euclidean distance measure.

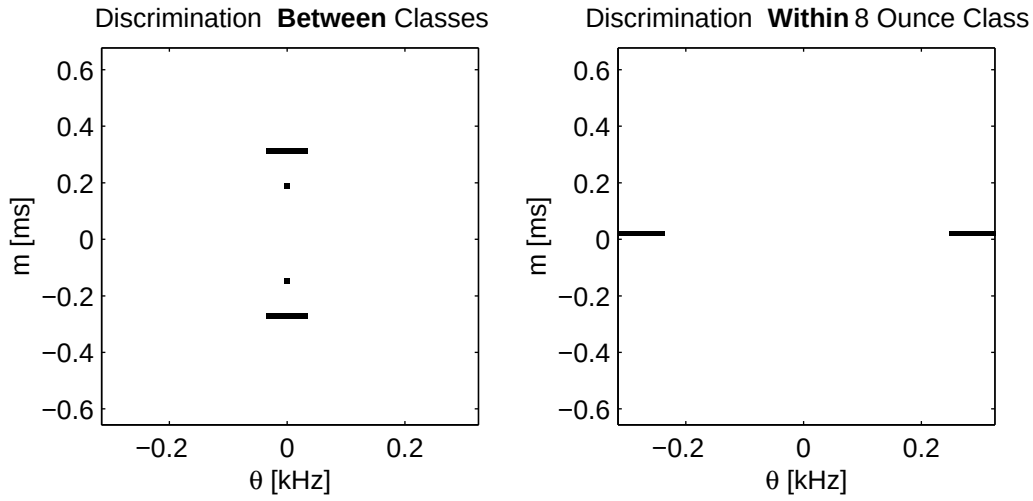

Figure 6.3: Class dependent kernels

The kernel found by our procedure is shown in the left half of Figure 6.3. By design, the kernel only takes on values of 1.0 and 0.0. The dark areas in the figure represent the only non-zero regions in the kernel.

This kernel focuses most of its energy along the $\eta = 0$ line in the ambiguity plane. Such a kernel tends to emphasize the spectral content of a signal. This indicates that, for this classification task, frequency information is more important. This agrees with our intuition that the difference between the classes is in the resonance of the bottle.

Figure 6.4 shows examples of time-frequency representations before and after smoothing with a class dependent kernel, for one signal in each class. From these images, it is apparent that the kernel emphasizes the frequency differences while at the same time smoothes the signals along the time axis.

Our class-dependent kernel method was also used to generate a kernel that would

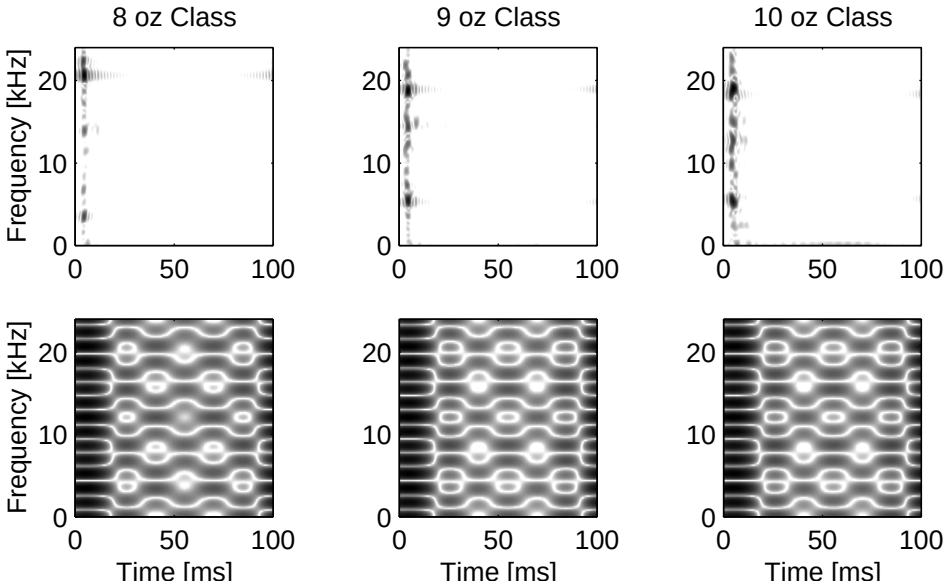

Figure 6.4: Example Rihaczek (top) and class dependent (bottom) distributions.

optimally discriminate between signals within a single class. That is, the training data consisted of each signal from the training data for one class, separately.

The resulting kernel is reproduced in the right half of Figure 6.3. It has a concentration of energy along the $\tau = 0$ axis. In contrast to our previous example, this kernel tends to emphasize the time domain information in the signal. This indicates that within the class, the spectral information was uniform and not useful for classification. The relevant information exists mainly in the time domain. To summarize, the only within-class differences were the way the bottle was tapped from signal to signal, a difference which class-dependent kernels could, if desired, be designed to be sensitive to.

6.3 Isolated Phone Recognition

For the isolated phone recognition task, individual phones were extracted from continuous speech, and subsequently classified. The phones were grouped into several

confusable sets, where within a set phone-based speech recognition systems usually experience confusion. This allowed us to experiment with small data sets and classification problems, fairly well labeled, that were known to cause problems to conventional systems.

6.3.1 The Data

All of the phone recognition experiments were performed on data from the TIMIT corpus. This corpus contains continuous speech from 460 native English speakers, organized into eight dialects. There are ten recorded utterances for each speaker, including two SA sentences, five SX sentences, and three SI sentences. Both SA sentences, SA1 and SA2, are the same for each speaker. Each SI sentence only occurs once in the dataset, and each SX sentence occurs six times. Including the SA sentences in the testing or training sets would bias the results, so they are usually excluded, leaving only eight utterances per speaker.

In the TIMIT corpus, the core test set consists of twenty-four speakers, two males and one female from each of the eight dialect regions. The full test set consists of the core test set, plus any speaker who shares an SX utterance with a core test set speaker. There are a total of 168 speakers in the full test set.

All speakers that are not in the full test set are placed in the training set. There are a total of 462 speakers in the training set.

6.3.2 MFCC/HMM Phone Recognition System

A conventional speaker-independent speech recognition system was constructed with Entropic Software's HTK modeling toolkit. This system provided a baseline for all improvements and modifications.

This baseline system uses Mel-frequency cepstral coefficients (MFCC). The coefficients are generated with a window length of 25ms and a window skip of 5ms.

The static MFCC and energy terms are supplemented with delta and delta-delta coefficients, for a total of 36 coefficients.

The model set consists of the forty-six context-independent phones shown in Table 1.1, and a vocabulary of 6200 words, although no language model is used.

Three-state hidden Markov models (HMM) are used to classify the phonemes. Each HMM state contains an eight mixture diagonal covariance continuous density Gaussian probability distribution function.

Initial HMM parameters are estimated from a subset of the training data using the tool `HInit`. Each state initially consists of one Gaussian mixture with a diagonal covariance.

Embedded training is performed with the tool `HERest`. The entire set of training data updates the set of models at once using Baum-Welch re-estimation. New Gaussian mixtures are added one at a time, followed by two passes of embedded training.

Using a fully trained model, isolated phone recognition can be performed either on segments extracted from the full test set, or on the continuous speech.

The phone, word, and sentence-level recognition rate for the system is comparable to similar systems referenced in the literature. Lee and Hon [31] reported a phone recognition rate of 58.77% using context-independent phones and no language model. Our baseline system, with the parameters cited above, has a phone recognition rate of 52.64%.

Table 6.1 and Table 6.2 show the confusion matrices for two confusable test sets. Here, a set of vowels and a set of consonants were extracted from the test set. A fully trained model was used to perform isolated recognition on each set, constrained to choose within the confusable set.

Each row of the tables corresponds with the correct transcription, and each column corresponds with the recognizer output. For example, twenty-nine utterances of /aa/ were misrecognized as /ae/, for example, and the overall error rate for that phone

Table 6.1: Vowel performance of the HMM recognition system

	aa	ae	ah	ao	ax	ay	Error Rate
aa	504	29	42	116	5	150	40.4%
ae	20	634	26	2	0	90	17.9%
ah	143	102	305	47	36	227	64.5%
ao	125	1	10	581	0	44	23.6%
ax	89	44	292	82	770	158	46.3%
ay	44	13	18	2	0	609	11.2%

was 40.4%.

As expected, this baseline system already performs reasonably well. The overall error rate within this set is 36.5%. Among the vowels chosen, the pair {/aa/,/ao} is the most troublesome.

The average error rate within the set of consonants, Table 6.2, is 24.6%, considerably lower than for the vowels. It is reasonable to assume that this is due, at least in part, to the fit of the model to the data. The consonants generally exhibit piecewise stationary behavior, whereas the vowel sounds can continuously change over time.

6.3.3 Time-Frequency Isolated Phone Recognition

As a first step towards building a full recognition system, the class dependent technique was explored on a less complex task. The recognition task was reduced to isolated recognition of phones extracted from continuous speech, so that it would be easier to examine the behavior of the new techniques.

The class dependent phone recognition system, by itself, did not produce stellar results. However, in the next section, it will be shown that the information in the class dependent time-frequency representation can be used to supplement the MFCC

Table 6.2: Consonant performance of the HMM recognition system

	b	d	g	k	p	t	Error Rate
b	557	33	47	3	218	28	37.1%
d	24	404	80	45	39	249	52.0%
g	10	18	276	120	12	16	38.9%
k	3	6	37	1022	42	94	15.1%
p	25	5	0	17	840	70	12.2%
t	1	59	6	41	54	1206	11.8%

features to build a recognition system that outperforms one built on either feature set in isolation.

The class dependent system was built using custom software, written in the "C" programming language, glued together with Matlab script files. The only features used in this system were time-frequency representations. These representations are generated with a window length of 250ms and a window skip of 50ms. Increasing the window length and skip by a factor of ten was possible, and desirable, because of the nature of the feature set. A subset (mask) of the ambiguity plane representation for a particular task is chosen as described previously, using the class-conditional method.

A single mixture Gaussian is used to classify the data. The covariance is a full, smoothed, covariance, estimated using the method presented in Chapter 3. It should be possible to extend these methods to use mixtures of smoothed full covariance mixtures.

It is notable that we get good results even when there is no state information present in the classifier. We would expect the quality of results to be directly related to the stationarity of the actual phone.

Recognition performance was tested on several separate discrimination tasks, sum-

Table 6.3: Confusable acoustic classes for phone classification

Class	Members
Vowels	/aa/, /ae/, /ah/, /ao/, /ax/, /ay/
Glides	/l/, /r/, /w/
Consonants	/b/, /d/, /g/, /k/, /p/, /t/
Consonant-Vowel	/ae/, /n/, /ae/-/n/, /n/-/ae/

Table 6.4: Isolated phone error rate (95% confidence)

	HMM Recognizer	Class Dependent TFR
Vowels	36.9%(±0.9%)	49.7%(±1.0%)
Glides	15.5%(±1.0%)	25.9%(±1.2%)
C-V Pair	36.3%(±2.8%)	31.5%(±2.7%)

marized in Table 6.3. The goal was to show that the class dependent kernel function did appropriate things when presented with actual speech data, and that the time-frequency representations contained enough information to build a speech recognition system.

A comparison of performance for three of the classes is presented in Table 6.4. The HMM recognition system outperformed the system built upon class dependent time-frequency representations in two out of the three cases.

For the "vowels" test, the class dependent kernel only chose autocorrelation coefficients along the $\eta = 0$ axis. This is a confirmation of the knowledge that these sounds only differ in their stationary spectra, but was found automatically. The class dependent result, therefore, is identical to training on four fixed lags of the autocor-

relation function. The HMM Recognizer produces approximately 26% fewer errors than the class dependent TFR.

For the "Glides" test, the class dependent kernel chose both stationary autocorrelation coefficients and points along $\eta = 1$, which correspond to modulations around 6.7 Hz. The class dependent kernel classifies the data not only according to its stationary spectral information, but how that spectrum changes with time. For this test, the HMM Recognizer produces approximately 40% fewer errors than the class dependent TFR.

In the third test, discrimination among {/ae/, /n/,/ae-n/, /n-ae/}, the class dependent kernel did not use any stationary spectral information. It used five points in the ambiguity plane, at τ values that correspond to 6.7 and 13.3 Hz. In this case, our time-frequency approach outperformed the the HMM/MFCC based recognizer. The class dependent TFR produced approximately 13% fewer errors than the class dependent TFR. The confidence intervals for this test are greater than the first two, because there are fewer occurrences of specific phone pairs than individual phones in the database.

6.3.4 Hybrid System Phone Recognition

A hybrid system was constructed that combined the outputs of both the MFCC and class dependent time-frequency representation features.

First, the set of recognition likelihoods for each model were normalized across the set of all possible utterances.

$$p_{norm}[k] = \frac{p[k]}{\sum_{k'} p[k']} \tag{6.1}$$

Next, a mixing parameter, α, was used to combine the likelihoods from the two classifiers. The mixing parameter was chosen to minimize the error after resubstitution with the training set.

$$p_{mix}[k] = p^{(1)}_{norm}[k] + p^{(2)}_{norm}[k] \tag{6.2}$$

Table 6.5: Error analysis of the three phone classifiers

Phone Set	Examples	TFR-GMM	MFCC-HMM	Combination
`aa ae ah ao ax ay`	5360	2466	1941	1838
`b d g p t k`	5707	3283	2009	2013

The results of this mixing were quite good, and motivated us in building a larger speech recognition system. On the "Vowels" set, the hybrid system reduced errors by up to six percent. Table 6.5 shows two example cases. For the vowels, the errors are reduced by approximately six percent. For the consonants, the error rate actually went up, but the change was much less than one percent.

6.4 English Alphabet Recognition

The next set of tasks involved recognizing isolated American English letters. This task was chosen for two reasons. First, the recognizer could employ a flat language model, and the recognition accuracy is a function of only the acoustic modeling. Second, smaller number of acoustic classes and amount of data could result in faster turn-around on experiments. Both of these reasons make English alphabet recognition an ideal task for prototyping and demonstrating the power of the CD-TFR.

6.4.1 The Data

The ISOLET corpus[14] is used for the experiments. The task consists of recognizing isolated letters drawn from the 26 letter American English alphabet.

There are a total of 150 speakers, which are conventionally divided into a set of 120 training speakers and 30 testing speakers. Each speaker is associated with 52 utterances, consisting of two utterances of each letter. This makes for a total corpus

size of 7800 utterances, equally divided among the letters of the alphabet and the gender of the speaker.

In 1990, Cole *et. al.* reported a 4.68% speaker-independent error rate (73/1560 errors) on this task[14]. The system consisted of a rule-based segmenter and neural network classifier.

Other authors have also attacked this task, with varying results. The previous best appears to be by Loizou and Spanias, with a speaker independent error rate of 2.63% (41/1560 errors) [32]. This system incorporated context-dependent phone hidden Markov models, as well as new feature representations for improved stop consonant discrimination, and subspace approaches for improved nasal discrimination.

Microsoft's Whisper speech recognizer was used as the baseline system in our experiments. It has the flexibility to take different acoustic and language models. Here we used a set of Hidden Markov Models with continuous-density output distributions consisting of Gaussian mixture densities. A more complete description of the Whisper speech recognition system can be found in [24].

6.4.2 Model Configuration

Three separate model configurations were tested. All three were composed of speaker-independent phone models. The simplest configuration used context and gender independent models. We made the system progressively more complex by first adding context dependent, and then gender dependent, models.

A standard dictionary was used to map letters to corresponding phone models. There were 27 context independent phones in this dictionary, and 53 context dependent phones.

6.4.3 Acoustic Features

Mel-frequency cepstral coefficients (MFCC) were chosen as the feature representation for the baseline system. It has very good performance and is currently widely used

in many state-of-the-art speech recognition systems.

The static MFCC coefficients are generated with a window length of 25ms and a window skip of 10ms. These are augmented by full delta and half of the delta-delta coefficients, for a total of 33 coefficients.

The class dependent time-frequency features were used both as the only features for recognition, and as an alternative feature stream during decoding.

The CD-TFR were generated with a window length of 100ms and a window skip of 10ms. This is a shorter window skip than we used in the isolated phone recognition system. The change was necessary to have synchronous MFCC and time frequency features. The time-frequency features were subset of the ambiguity plane representation for a particular task is chosen as described previously, using the class dependent method.

6.4.4 Multiple Feature Decoding

A scheme for multiple feature decoding was incorporated to improve recognition performance in the hybrid system with CD-TFR and MFCC features. During training, one set of models is trained for each feature stream. In the decoder, the search space is augmented with an additional feature stream dimension. The decoder will consider all the features and find a word sequence with maximum likelihood over all the models, states and feature streams. For more details see [27].

We only deal with isolated recognition in the ISOLET task. Given that there is only one letter in each utterance, the framework can be simplified to:

$$L^* = \arg\max_{L} \sum_{i \in \Psi} p\left(L|F_i\right) p\left(F_i|A\right) \tag{6.3}$$

In this equation, Ψ is the set of all feature streams, A is the waveform data for speech signal, and F_i is the feature associated with a given feature stream. The term $p(F_i|A)$ serves as a weighting factor to weight contributions from different feature

streams. In our hybrid system with multiple features, the weights are class-dependent.

6.4.5 Experimental Results

Recognition was performed using MFCC features only, using CD-TFR features only, and both features together in a hybrid system.

For each system, the number of parameters (Gaussian mixtures) was adjusted to maximize recognition accuracy. The goal was not to compare systems with equivalent numbers of parameters, but rather to demonstrate the maximum performance of the system for a given feature set.

6.4.6 Baseline System

The recognition systems based solely on MFCC features performed as well as, or better than, the best published results on this task. The results are summarized in Table 6.6. The recognition systems used either context independent (CI) or context dependent (CD) phones, and either gender independent (GI) or gender dependent (GD) models.

Table 6.6: MFCC performance

System	Mixtures	Error Rate	
CI-GI	12	79/1560	5.06%
CD-GI	8	50/1560	3.21%
CD-GD	8	38/1560	2.44%

The baseline system's errors (Figure 6.5) were consistent with those expected. Out of 38 errors, 32 occurred within three confusable letter classes: the nasals {M, N}, the fricatives {S, F}, and the infamous E-set, {B, C, D, E, G, P, T, V, Z}. If all of

these within-class errors are eliminated, only six errors would remain, a significant improvement over even the best result.

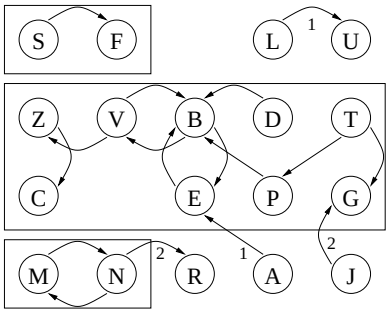

Figure 6.5: MFCC system substitution errors. Each box represents a confusable set.

6.4.7 CD-TFR System

Features for the CD-TFR system were designed to eliminate the within-class confusion typical of the MFCC system. That is, the discriminant was computed to discriminate within classes of phone models {/s/, /f/}, {/m/, /n/}, and {/b/, /s/, /d/, /gh/, /p/, /t/, /v/, /z/}.

The CD-TFR recognition system's accuracy fell between the initial results by Cole[14] and the result by Loizou[32]. The actual error rates for this system are presented in Table 6.7.

Table 6.7: CD-TFR performance

System	Mixtures	Error Rate	
CI-GI	12	73/1560	4.68%
CD-GI	8	64/1560	4.10%
CD-GD	8	58/1560	3.72%

While the CD-TFR system did a better job of not confusing letters within confusable sets, too many between-class errors were introduced.

It is notable that the performance increase in moving to a context dependent and gender dependent system is not nearly as great as it was for the MFCC system. The current hypothesis is that since the CD-TFR feature selection algorithm is both context independent and gender independent, context and gender information is smoothed in the feature space, and not recoverable by modifying the stochastic model.

6.4.8 Hybrid System

As mentioned previously, if all of the within-class errors were eliminated, but the between-class errors remained, the performance of the MFCC system would be phenomenal. The hybrid system we built is a two-stage recognition system that refines the classification accuracy within each confusable class by using multiple features. This two-stage approach has the potential of eliminating within-class errors, but neither increases nor decreases the between-class errors.

The first stage uses only MFCC features for recognition. As shown in section 6.4.6, this stage makes few between-class errors.

The second stage re-labels data within each class using CD-TFR features tuned for discrimination within that class together with MFCC features. It is built from the models generated for the context dependent, gender dependent MFCC and CD-TFR systems. The decoder for this stage used the multiple feature stream approach described in Section 6.4.4.

The performance of the hybrid system is shown in Table 6.8.

Most of the improvement came from eliminating errors in the fricative set and the E-set.

The hybrid system, as expected, performed better than either the MFCC or the CD-TFR systems in isolation. Overall, the error rate was reduced by 34.2%. Our

Table 6.8: Hybrid system performance

System	Error Rate	
CD-TFR	58/1560	3.72%
MFCC	38/1560	2.44%
Hybrid	**25/1560**	**1.60%**
Improvement	**13/38**	**34.2%**

error rate of 1.60% has almost 40% fewer errors than the previous best published result on the ISOLET task[32].

6.5 Conclusion

Using the concepts of operator theory, we've been able to forge a direct connection between a discrete, finite-length input signal, and its discrete-time, discrete-frequency TFR. With our approach, we represent each TFR as the 2-D circular convolution of a kernel with the Rihaczek distribution of the signal.

A study of the properties of this representation has been presented. The time-frequency representations have a covariance structure that can be exploited to produce improved model estimates. These estimates can be used to model the data directly, or in conjunction with the class dependent kernel to reduce the dimensionality of the feature.

All valid time-frequency representations lie on an N dimensional surface embedded in a $N(N+1)/2$ feature space. This structure can be exploited to find distances between representations, and has other applications such as signal morphing.

It is important to note that the kernel we obtain for optimal separation maximizes the time-frequency difference given the original distribution (the Rihaczek). If the two

signal classes have very dissimilar Rihaczek TFRs, then our method will find very little room for improvement.

The class dependent kernel was developed to extract from the time-frequency representation those dimensions useful for classification.

It was shown that all of these techniques can be used together to develop time-frequency features for input to a classification system. These features are not limited to static spectral estimates and their time differences, and are designed not to contain extraneous information.

We have described a CD-TFR feature for speech recognition. It can also be used together with traditional features such as MFCC to improve upon the baseline performance. CD-TFR features can be automatically learned from data to maximize the discriminability. The experiments have demonstrated that CD-TFR features are very effective, especially for discrimination within classes. Together with MFCC, we achieved a 1.60% error rate, a very high performance on ISOLET alphabet recognition.

The outstanding performance of this system can be attributed to both the quality of the newly generated CD-TFR features, and the efficacy of the multiple feature stream decoder.

Chapter 7

FUTURE WORK

Although it has been shown that class dependent time frequency representations can be successfully used in a speech recognition system, several extensions are possible to further this work.

7.1 Signal Morphing

One possible application of this geodesic distance, unrelated to speech recognition, is that of signal morphing. Instead of just calculating the length of the shortest path between two signals, one can keep the entire path. Each point on the path is a valid signal, in the continuum of signals that starts and ends at desired points.

The numerical approximation would be uninteresting for this purpose, because it would always return a linear path through signal space.

The algorithmic approximation, on the other hand, returns an arbitrary number of signals, equidistant in the TFR space. Besides the shortest past, other constraints can be imposed. Possibilities are that the autocorrelation, magnitude spectrum, or energy remain constant.

7.2 Gaussian Mixture Models

It was alluded to in Chapter 3 that multiple mixtures of Gaussians could be used to model inter-speaker or intra-speaker variability in speech recognition.

The parametric covariance estimates developed for this system were designed with a static production model. The waveform production model consisted of white noise

passing through a nonstationary filter, which was identical across all of the data.

One solution to this problem is to modify the training procedure to train several mixtures at once. In standard Baum-Welch re-estimation, means and variances are updated using a weighted average. The weights $L_j(t)$ are the probability, given the current model and observations o_t, $1 \leq t \leq T$, of the model being in state j at time t. The new mean and covariance formulas are usually,

$$\hat{\mu}_j = \frac{\sum_{t=1}^{T} L_j(t) o_t}{\sum_{t=1}^{T} L_j(t)}, \text{ and} \tag{7.1}$$

$$\hat{\Sigma}_j = \frac{\sum_{t=1}^{T} L_j(t)(o_t - \mu_j)(o_t - \mu_j)^T}{\sum_{t=1}^{T} L_j(t)}. \tag{7.2}$$

Using the same mean update formula, and the new parametric covariance estimate based on the new mean, a multiple Gaussian mixture model could be formed.

7.3 Large Vocabulary Continuous Speech Recognition

For a large vocabulary task, multiple instances of the class conditional time frequency representations could be used to discriminate within confusable classes.

One approach would partition the set of English phones into confusable sets. The first step would be to do a study of the errors made by a well trained HMM/MFCC speech recognition system. Similarly to the alphabet recognition example in the previous chapter, certain phones will naturally group together. The major design tradeoff in this approach is to minimize the number of between-class errors, while keeping the number of class members at a reasonable level. In the extreme, if all phones belonged to the same class, there would be no between-class errors, but we would probably not gain much from a class conditional design.

Given the results presented previously, each feature stream alone should improve recognition within its class members. The problem occurs when there are multiple feature streams competing with each other. At each frame, there is only one correct label, and only one out of the many class conditional features designed to recognize

that label.

What is needed to make this work is a more sophisticated, frame level, integrated training and testing recognition system.

BIBLIOGRAPHY

[1] M.G. Amin, G.T. Venkatesan, and J.F. Carroll. A constrained weighted least squares approach for time-frequency distribution kernel design. *IEEE Transactions on Signal Processing*, 44(5), May 1996.

[2] Les Atlas, James Droppo, and Jack McLaughlin. Optimizing time-frequency distributions for automatic classification. In *Proceedings of the SPIE*, volume 3162, pages 161–171, 1997.

[3] Lalit R. Bahl, Peter F. Brown, Peter V. de Souza, and Robert L. Mercer. Estimating hidden Markov model paramaters so as to maximize speech recognition accuracy. *IEEE Transactions on Speech and Audio Processing*, 1(1):77–83, January 1993.

[4] Lalit R. Bahl, Peter F. Brown, Peter V. de Souza, and Robert L. Mercer. A new algorithm for the estimation of hidden Markov model parameters. In *Proceedings of the 1998 IEEE International Conference on Acoustics Speech and Signal Processing* [26], pages 493–7.

[5] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[6] Hervé Bourlard and Stéphane Dupont. A new ASR approach based on independent processing and recombination of partial frequency bands. In *Proceedings ICSLP 96. Fourth International Conference on Spoken Language Processing*, volume 11, pages 426–9, Philadelphia, PA, USA, October 1996.

[7] Hervé Bourlard and Stéphane Dupont. Subband-based speech recognition. In *Proceedings of the 1997 IEEE International Conference on Acoustics Speech and Signal Processing* [25], pages 1251–4.

[8] C. Cerisara, J.-P. Haton, J.-F. Mari, and D. Fohr. A recombination model for multi-band speech recognition. In *Proceedings of the 1998 IEEE International Conference on Acoustics Speech and Signal Processing* [26], pages 717–20.

[9] C. Chesta, A. Girardi, P. Laface, and M. Nigra. Discriminative training of hidden Markov models using a classification measure criterion. In *Proceedings of the 1998 IEEE International Conference on Acoustics Speech and Signal Processing* [26], pages 449–52.

[10] T.A.C.M. Claasen and W.F.G. Mecklenbrauker. The wigner distribution—a tool for time-frequency signal analysis—part I: Continuous-time signals. *Philips Journal of Research*, 35:217–250, 1980.

[11] Leon Cohen. Generalized phase-space distribution functions. *J. Math. Phys.*, 7:781–786, 1966.

[12] Leon Cohen. *Time-Frequency Analysis*. Prentice Hall Signal Processing Series, New Jersey, 1995.

[13] Leon Cohen. A general approach for obtaining joint representations in signal analysis. i. characteristic function operator method. *IEEE Transactions on Signal Processing*, 44(5):1080–90, May 1996.

[14] Ronald Cole, Mark Fanty, Yeshwant Muthusamy, and Murali Gopalakrishnan. Speaker-independent recognition of spoken English letters. In *Procceedings of the International Joint Conference on Neural Networks*, volume 2, pages 45–51, 1990.

[15] Manuel Davy and Christian Doncarli. Optimal kernels of time-frequency representations for signal classification. In *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pages 581–4, Pittsburg, PA, USA, October 1998.

[16] James Droppo and Les Atlas. Application of classifier-optimal time-frequency distributions to speech analysis. In *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pages 585–588, 1998.

[17] P. Flandrin. A time-frequency formulation of optimum detection. *IEEE Transactions on Signal Processing*, 36(9), September 1988.

[18] Bradford W. Gillespie and Les Atlas. Data driven optimization of time and frequency resolution for radar transmitter identification. In *Proceedings of the SPIE–The International Society for Optical Engineering*, volume 3162, San Diego, CA, USA, 1998.

[19] Bradford W. Gillespie and Les Atlas. Optimizing time-frequency kernels for classification. *IEEE Transactions on Signal Processing (submitted)*, 1998.

[20] P. S. Gopalakrishnan, Dimitri Kanevsky, Arthur Nádas, and David Nahamoo. A generalization of the Baum algorithm to rational objective functions. In *1989 IEEE International Conference on Acoustics Speech and Signal Processing*, volume 1, pages 631–4, Glasgow, UK, May 1989.

[21] P. S. Gopalakrishnan, Dimitri Kanevsky, Arthur Nádas, and David Nahamoo. An inequality for rational functions with applications to some statistical estimation problems. *IEEE Transactions on Information Theory*, 37(1):107–13, January 1991.

[22] Cristoph Heitz. Optimum time-frequency representations for the classification and detection of signals. In *Applied Signal Processing*, volume 3, pages 124–143, 1995.

[23] Hynek Hermansky, Sangita Tibrewala, and Misha Pavel. Towards ASR on partially corrupted speech. In *ICSLP 96. Fourth International Conference on Spoken Language Processing*, volume 1, pages 462–5, Philadelphia, PA, USA, October 1996.

[24] X. Huang, A. Acero, F. Alleva, M. Y. Hwang, L. Jiang, and M. Mahajan. Microsoft Windows highly intelligent speech recognizer: Whisper. In *Proceedings of the 1995 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Detroit, May 1995.

[25] Institute of Electrical and Electronics Engineers Signal Processing Society. *Proceedings of the 1997 IEEE International Conference on Acoustics Speech and Signal Processing*, Munich, Germany, April 21-24 1997. Los Alamitos, Calif.: IEEE Computer Society.

[26] Institute of Electrical and Electronics Engineers Signal Processing Society. *Proceedings of the 1998 IEEE International Conference on Acoustics Speech and Signal Processing*, Seattle, WA, USA, May 12-15 1998. Piscataway, NJ: IEEE Service Center.

[27] Li Jiang and X.D. Huang. Unified decoding and feature representation for improved speech recognition. In *Proceedings of Eurospeech 99*, volume 3, pages 1331–1334, 1999.

[28] Shoji Kajita, Kazuya Takeda, and Fumitada Itakura. Spectral weighting of SB-COR for noise robust speech recognition. In *Proceedings of the 1998 IEEE In-*

ternational Conference on Acoustics Speech and Signal Processing [26], pages 621–4.

[29] S. Kapadia, V. Valtchev, and S. J. Young. MMI training for continuous recognition on the TIMIT database. In *1993 IEEE International Conference on Acoustics Speech and Signal Processing*, volume 2, pages 491–4, Minneapolis, MN, USA, April 1993.

[30] S. Kay and F. Bodreaux-Bartels. On the optimality of the wigner distribution for detection. In *Proceedings of the 1985 IEEE International Conference on Acoustics Speech and Signal Processing*, pages 1017–20, 1985.

[31] Kai-Fu Lee and Hsio-Wuen Hon. Speaker-independent phone recognition using hidden Markov models. *IEEE Transactions on Acoustics Speech and Signal Processing*, 37(11):1641–8, November 1989.

[32] Philipos Loizou and Andreas Spanias. High-performance alphabet recognition. *IEEE Transactions on Speech and Audio Processing*, 4(6):430–445, November 1996.

[33] Paul McCourt, Saeed Vaseghi, and Naomi Harte. Multi-resolution cepstral features for phoneme recognition across speech sub-bands. In *Proceedings of the 1998 IEEE International Conference on Acoustics Speech and Signal Processing* [26], pages 557–60.

[34] Jack McLaughlin, James Droppo, and Les Atlas. Class-dependent, discrete time-frequency distributions via operator theory. In *Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 2045–2048, 1997.

[35] Bernard Merialdo. Phonetic recognition using hiden Markov models and maximum mutual information training. In *Proceedings of the 1998 IEEE International Conference on Acoustics Speech and Signal Processing* [26], pages 111–14.

[36] J. E. Moyal. Quantum mechanics as a statistical theory. In *Proc. Camb. Phil. Soc.*, volume 45, pages 99–124, 1949.

[37] A. Nádas. A decision theoretic formulation of a training problem in speech recognition and a comparison of training by unconditional versus conditional maximum likelihood. *IEEE Transactions on Acoustics, Speech and Signal Processing*, pages 814–17, August 1983.

[38] Siva Bala Narayanan, Jack McLaughlin, Les Atlas, and James Droppo. An operator theory approach to discrete time-frequency distributions. In *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pages 521–524, 1997.

[39] Yves Normandin, Régis Cardin, and Renato De Mori. High-performance connected digit recognition using maximum mutual information estimation. *IEEE Transactions on Speech and Audio Processing*, 2(2):299–311, April 1994.

[40] Yves Normandin and Salvatore D. Morgera. An improved MMIE training algorithm for speaker-independent, small vocabulary, continuous speech recognition. In *1991 IEEE International Conference on Acoustics Speech and Signal Processing*, volume 1, pages 537–40, Toronto, Ontario, CA, May 1991.

[41] A. H. Nutall. Alias-free Wigner distribution functions and complex ambiguity functions for discrete-time samples. Technical Report 8533, Naval Underwater Systems Center, April 14 1989.

[42] Shigeki Okawa, Enrico Bocchieri, and Alexandros Potamianos. Multi-band speech recognition in noisy environments. In *Proceedings of the 1998 IEEE International Conference on Acoustics Speech and Signal Processing* [26], pages 641–4.

[43] Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–86, February 1989.

[44] M. Richman, T. Parks, and R. Shenoy. Features of a discrete Wigner distribution. In *1996 IEEE Digital Signal Processing Workshop Proceedings*, pages 427–430, 1996.

[45] A. W. Rihaczek. Signal energy distribution in time and frequency. *IEEE Transactions on Information Theory*, 14:369–374, 1968.

[46] B. Tacer and P. Loughlin. Time-frequency based classification. In *Proceedings of the SPIE–The International Society for Optical Engineering*, volume 2846, pages 186–92, Denver, CO, USA, August 1996.

[47] Sangita Tibrewala and Hynek Hermansky. Sub-band based recognition of noisy speech. In *Proceedings of the 1997 IEEE International Conference on Acoustics Speech and Signal Processing* [25], pages 1255–8.

[48] M. J. Tomlinson, M. J. Russell, R. K. Moore, A. P. Buckland, and M. A. Fawley. Modeling asynchrony in speech using elementary signle-signal decomposition. In *Proceedings of the 1997 IEEE International Conference on Acoustics Speech and Signal Processing* [25], pages 1247–50.

[49] J. Ville. Theorie et applications de la notion de signal analytique. *Cables et Transmission*, 2 A:61–74, 1948.

[50] I. Vincent, C. Noncarli, and E. Le Carpentier. Non stationary signals classification using time-frequency distributions. In *Proceedings of the International Symposium on Time-Frequency and Time-Scale Analysis*, pages 233–6, June 1996.

[51] Kuansan Wang, Chin-Hui Lee, and Biing-Hwang Juang. Selective feature extraction via signal decomposition. *IEEE Signal Processing Letters*, 4(1):8–11, January 1997.

[52] Hideyuki Watanabe, Alain Biem, and Shigeru Katagiri. Toward a unified design of pattern recognizers. In *Neural Networks for Signal Processing VI. Proceedings of the 1996 IEEE Workshop*, pages 283–92, Kyoto, Japan, September 1996.

[53] Hideyuki Watanabe and Shigeru Katagiri. HMM speech recognizer based on discriminative metric design. In *Proceedings of the 1997 IEEE International Conference on Acoustics Speech and Signal Processing* [25], pages 3237–40.

[54] Hideyuki Watanabe, Tsuyoshi Yamaguchi, and Shigeru Katagiri. A novel approach to pattern recognition based on discriminitive metric design. In *Neural Networks for Signal Processing V. Proceedings of the 1995 IEEE Workshop*, pages 48–37, Cambridge, MA, USA, September 1995.

[55] E. J. Zalubas, J. C. Oneill, W. J. Williams, and A. O Hero III. Shift and scale invariant detection. In *Proceedings of the 1997 IEEE International Conference on Acoustics Speech and Signal Processing* [25], pages 3637–40.

[56] Y. Zhao, L. E. Atlas, and R. J. Marks. The use of cone-shaped kernels for generalized time-frequency representations of nonstationary signals. *IEEE Transactions on Acoustics Speech and Signal Processing*, 38:1084–91, 1990.

Appendix A

PARAMETRIC NONSTATIONARY AUTOCORRELATION COVARIANCE ESTIMATE

In Chapter 3, autocorrelation features were used as an illustration of the parametric covariance estimation. In this appendix, the more general case of nonstationary autocorrelation features.

Recall from Chapter 2 that the discrete time frequency representation in the auto-ambiguity plane can be interpreted as a nonstationary autocorrelation estimate. That is, the representation $A[\eta, \tau]$ contains the average, stationary, autocorrelation where $\eta = 0$. For other values of η, this representation consists of that portion of the autocorrelation of the signal that is being modulated at a rate of *eta* cycles per window.

A.1 Useful Identities

In the sections to come, three formulas will prove useful. First, the definition of the time discrete frequency representation in the auto-ambiguity plane is

$$A[\eta, \tau] = \sum_m x[m]x^*[m+\tau]W_N^{m\eta}. \tag{A.1}$$

This can be related to the instantaneous autocorrelation through an N-point discrete Fourier transform.

$$\frac{1}{N}\sum_\eta A[\eta, \tau]W_N^{-\eta m} = x[m]x^*[m+\tau] \tag{A.2}$$

Finally, shifting the instantaneous autocorrelation by b rows and a columns yields,

$$\frac{1}{N}\sum_\eta A[\eta, \tau + a - b]W_N^{-\eta(m+b)} = x[m+b]x^*[m+\tau+a]. \tag{A.3}$$

A.2 The Solution

Using Equation A.1, the covariance is expanded as the expectation of the product of four random variables. According to the signal model, each $x[n]$ is Gaussian, so that the expectation of this product is the sum of three pairwise expectations. As a result, the solution has three terms, each of which is dealt with separately below.

$$\begin{aligned}
\Sigma[\eta_1,\tau_1,\eta_2,\tau_2] &= E\left[\left(\sum_n x[n]x^*[n+\tau_1]W_N^{\eta_1 n}\right)^*\left(\sum_m x[m]x^*[m+\tau_2]W_N^{\eta_2 m}\right)\right] \\
&= \sum_{m,n} E\left[x^*[n]x[n+\tau_1]x[m]x^*[m+\tau_2]\right]W_N^{\eta_2 m-\eta_1 n} \\
&= A^*[\eta_1,\tau_1]A[\eta_2,\tau_2] \\
&\quad +\frac{1}{N}\sum_{a,n} A[a+\eta_2,n+\tau_2]A^*[a+\eta_1,n+\tau_1]W_N^{a\tau_2-\eta_1 n} \\
&\quad +\frac{1}{N}\sum_{a,n} A[a+\eta_2,n-\tau_2]A^*[a+\eta_1,n+\tau_1]W_N^{-a\tau_2-\eta_1 n}W_N^{-\eta_2\tau_2}
\end{aligned}$$

A.2.1 Term One

This term can be attributed to the variance induced by the mean of the feature. This is usually subtracted out after the estimation process, so in practice it never needs to be computed.

$$\sum_{m,n} E\left[x^*[n]x[n+\tau_1]\right]E\left[x[m]x^*[m+\tau_2]\right]W_N^{\eta_2 m-\eta_1 n} = A^*[\eta_1,\tau_1]A[\eta_2,\tau_2] \tag{A.4}$$

A.2.2 Term Two

The second term resolves into a weighted inner product between two shifted auto-ambiguity functions.

$$\sum_{m,n} E\left[x[m]x^*[n]\right]E\left[x^*[m+\tau_2]x[n+\tau_1]\right]W_N^{\eta_2 m-\eta_1 n}$$

$$= \sum_{a,n} A[a+\eta_2, n+\tau_2] A^*[a+\eta_1, n+\tau_1] W_N^{a\tau_2 - \eta_1 n} \tag{A.5}$$

The detailed derivation re-writes each pairwise expectation of instantaneous autocorrelation as the DFT of an equivalent auto-ambiguity function. The simplification then follows from the properties of sums of the complex exponential function.

$$\sum_{m,n} E\left[x[m]x^*[n]\right] E\left[x^*[m+\tau_2]x[n+\tau_1]\right] W_N^{\eta_2 m - \eta_1 n}$$

$$= \sum_{m,n} E\left[x[m]x^*[m+n]\right] E\left[x^*[m+\tau_2]x[m+n+\tau_1]\right] W_N^{\eta_2 m - \eta_1 (m+n)}$$

$$= \sum_{m,n} \left(\frac{1}{N} \sum_a A[a,n] W_N^{-am} \right) \left(\frac{1}{N} \sum_b A[b, n+\tau_1-\tau_2] W_N^{-b(m+\tau_2)} \right)^* W_N^{\eta_2 m - \eta_1 (m+n)}$$

$$= \frac{1}{N^2} \sum_{a,b,n} A[a,n] A^*[b, n+\tau_1-\tau_2] W_N^{b\tau_2} W_N^{-\eta_1 n} \sum_m W_N^{m(b-a+\eta_2-\eta_1)}$$

$$= \frac{1}{N} \sum_{a,n} A[a+\eta_2, n+\tau_2] A^*[a+\eta_1, n+\tau_1] W_N^{a\tau_2 - \eta_1 n}$$

A.2.3 Term Three

The third also term resolves into a weighted inner product between two shifted auto-ambiguity functions, under the assumption that the signal $x[n]$ is real.

$$\sum_{m,n} E\left[x^*[m+\tau_2]x^*[n]\right] E\left[x[m]x[n+\tau_1]\right] W_N^{\eta_2 m - \eta_1 n}$$

$$= \frac{1}{N} \sum_{a,n} A[a+\eta_2, n-\tau_2] A^*[a+\eta_1, n+\tau_1] W_N^{-a\tau_2 - \eta_1 n} W_N^{-\eta_2 \tau_2} \tag{A.6}$$

$$\sum_{m,n} E\left[x^*[m+\tau_2]x^*[n]\right] E\left[x[m]x[n+\tau_1]\right] W_N^{\eta_2 m - \eta_1 n}$$

$$= \sum_{m,n} E\left[x^*[m]x^*[n]\right] E\left[x[m-\tau_2]x[n+\tau_1]\right] W_N^{\eta_2 m - \eta_1 n}$$

$$= \frac{1}{N} \sum_{a,n} A[a+\eta_2, n-\tau_2] A^*[a+\eta_1, n+\tau_1] W_N^{-a\tau_2 - \eta_1 n} W_N^{-\eta_2 \tau_2}$$

VITA

James "Jasha" Garnet Droppo III was born in Albany, New York, on March 2, 1972. Later that year, his family moved to Pasco, Washington without his consent. He received the Bachelor of Science degree in Electrical Engineering, *cum laude*, honors, from Gonzaga University in 1994; the Master of Science degree in Electrical Engineering from the University of Washington in 1996; and the Doctor of Philosophy degree in Electrical Engineering from the same institution in 2000. At the University of Washington, he helped to develop and promote a discrete theory for time-frequency representations of audio signals, with a focus on speech recognition. His other projects have included acoustic pyrometry, subliminal audio message encoding, and speaker verification.