UNIVERSITY OF CALIFORNIA, SAN DIEGO

High-Rate Optimized Quantization Structures and Speaker-Dependent Wideband Speech Coding

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in
Electrical Engineering
(Signal and Image Processing)

by

Ethan Robert Duni

Committee in charge:

Professor Bhaskar D. Rao, Chair
Professor Robert R. Bitmead
Professor Sanjoy Dasgupta
Professor Kenneth Kreutz-Delgado
Professor Kenneth Zeger

2007

The dissertation of Ethan Robert Duni is approved, and it is acceptable in quality and form for publication on microfilm:

Chair

University of California, San Diego

2007

For Anand Subramaniam.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would first like to thank my advisor, Prof. Bhaskar D. Rao, for his guidance, support and encouragement throughout my graduate career. His insight and participation has greatly enriched my experience in classes, teaching, research, conferences, publications and overall career development. I am also thankful to my committee members, Dr. Kenneth Zeger, Dr. Kenneth Kreutz-Delgado, Dr. Robert R. Bitmead and Dr. Sanjoy Dasgupta for their valuable comments and time. Without the contributions of my advisor and committee members, I would not have been able to complete this work.

I owe a special debt to my former lab-mate Anand Subramaniam. In addition to laying the foundations for my own research, his tutelage was instrumental in my development as a researcher. His sunny disposition and impressive insight will be sorely missed by all who knew him. Next, I owe thanks to my collaborators Chandra R. Murthy and Jun Zheng. Their curiosity and collaborative spirit greatly expanded my horizons, and my publications list owes them a particular debt of gratitude. Also, I would like to thank Dr. William R. Gardner for his input and guidance during the formative part of my research career. This work would surely have suffered had I not been able to avail myself of his experience and acumen. I am also indebted to Koen Vos for many hours of constructive feedback and many helpful references. Finally, I would like to thank other group members: David Wipf, Yogananda Isukapalli, Aditya Jagannatham, June C. Roh and Wenyi Zhang for encouragement and insightful discussions.

This work was supported in part by MICRO Grants 03-073, 04-074, 05-033 and 06-174, sponsored by QUALCOMM Inc.

This dissertation is a collection of papers that were published or submitted for publication. The text of Chap. 2 is in part a reprint of the material which was coauthored with Bhaskar D. Rao and appeared in the March 2007 issue of *IEEE Transactions on Audio, Speech and Language Processing* under the title *"A High-Rate Optimal Transform Coder with Gaussian Mixture Companders"*.

Chap. 3 is a reprint of a paper coauthored with Bhaskar D. Rao which appeared in the March 2007 issue of *IEEE Transactions on Audio, Speech and Language Processing* under the title *"High-Rate Optimized Recursive Vector Quantization Structures Using Hidden Markov Models"*. The material in Chap. 4 is in preparation for a submission, coauthored with Bhaskar D. Rao, for publication in *IEEE Transactions on Audio, Speech and Language Processing* under the title *"Speaker-Dependent Wideband Speech Coding"*. The dissertation author was the primary researcher and author, and the co-authors listed in these publications contributed to or supervised the research which forms the basis for this dissertation.

VITA

2001	B.S. in Electrical Engineering, University of California, San Diego
2001-2003	Teaching Assistant University of California, San Diego
2004	M.S. in Electrical Engineering, University of California, San Diego
2002-2007	Research Assistant University of California, San Diego
2007	Ph.D. in Electrical Engineering, University of California, San Diego

PUBLICATIONS

Ethan R. Duni and Bhaskar D. Rao, "A High-Rate Optimal Transform Coder with Gaussian Mixture Companders", *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 15, No. 3, March 2007, pages 770-783.

Ethan R. Duni and Bhaskar D. Rao, "High-Rate Optimized Recursive Vector Quantization Structures Using Hidden Markov Models", *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 15, No. 3, March 2007, pages 756-769.

Jun Zheng, Ethan Duni and Bhaskar D. Rao, "Analysis of Multiple Antenna Systems With Finite Rate Feedback Using High Resolution Quantization Theory", *To Appear in IEEE Transactions on Signal Processing.*

Ethan R. Duni and Bhaskar D. Rao, "High-Rate Design of Transform Coders with Gaussian Mixture Companders", *ICASSP 2006.*

Chandra R. Murthy, Ethan R. Duni and Bhaskar D. Rao, "High-Rate Analysis of Vector Quantization for Noisy Channels", *ICASSP 2006.*

Ethan R. Duni and Bhaskar D. Rao, "High-Rate Training of Gaussian Mixture Vector Quantizers", *Data Compression Conference, March 2006.*

Jun Zheng, Ethan Duni and Bhaskar D. Rao, "Analysis of Multiple Antenna Systems with Finite-Rate Feedback Using High Resolution Quantization Theory", *Data Compression Conference, March 2006.*

Ethan R. Duni, Anand D. Subramaniam and Bhaskar D. Rao, "Improved Quantization Structures Using Generalized HMM Modelling with Application to Wideband Speech Coding", *ICASSP 2004.*

FIELDS OF STUDY

Major Field: Engineering

Studies in speech processing, quantization theory, digital signal processing, information theory, estimation theory, and their applications in the optimization of speaker-dependent wideband speech coders.

Professor Bhaskar D. Rao, University of California, San Diego

ABSTRACT OF THE DISSERTATION

High-Rate Optimized Quantization Structures and Speaker-Dependent Wideband Speech Coding

by

Ethan Robert Duni

Doctor of Philosophy in Electrical Engineering
(Signal and Image Processing)

University of California, San Diego, 2007

Professor Bhaskar D. Rao, Chair

Modern coding applications, such as wideband speech, are characterized by sources with large dimensions and unknown statistics, complicated distortion measures, and the need for high-quality quantization. However, the complexity of quantization systems must be kept in check as the dimension grows, requiring flexible quantization structures. These structures, in turn, require an automatic training method that can infer statistics from example data and balance the various factors to optimize performance. The development of efficient, flexible quantization structures also opens up new coding applications, such as speaker-dependent coding. This approach promises improved performance but presents a variety of implementational challenges.

The first part of this dissertation presents a variety of structured quantizers which strike different balances between complexity and performance. This includes the scalar transform coder, which is augmented with a flexible companding scalar quantizer based on Gaussian Mixtures. Next, a variety of extensions to the Gaussian Mixture Vector Quantizer (GMVQ) system for recursive coding are examined. Training techniques for these systems are developed based on High-Rate quantization theory, which provides a tractable objective function for use in automatic design. This replaces ad-hoc methods used for design of structured quantizers with a data-driven approach which is able to incorporate various distortion measures and structures. The performance of the systems is demonstrated on the problem of wideband speech spectrum coding.

The second part of this dissertation considers speaker-dependent wideband speech coding. Using the GMVQ system and training approach developed in the first portion, a study of the performance benefits of speaker-dependent coding in the CELP framework is undertaken. The three main types of CELP parameters (spectrum, adaptive codebook and fixed codebook) are all investigated, and the gains quantified. Next, a number of implementational issues related to speaker-dependent coding are addressed. A safety-net approach is utilized to provide robustness, and its implementation in the context of GMVQ is explored. A variety of online training architectures are presented which strike different balances between training complexity, communications overhead and performance. As components of these architectures, techniques for training on quantized data and recursive learning are examined.

1 Introduction

This dissertation considers a variety of quantization structures and examines their performance in the context of wideband speech coding. Increases in available computational power and communications resources have created interest in more demanding coding applications such as wideband speech and video. These applications require high-quality quantization of high-dimensional sources with complicated statistics, under complex distortion measures. Because the complexity of unstructured vector quantization grows exponentially with the source dimension, however, it is necessary to utilize structured quantizers to keep the complexity within reason. Quantizers with structure necessarily exhibit suboptimal performance due to limited ability to exploit source statistics and complex distortion measures, as well as inefficiencies in the spatial arrangement of codepoints. In light of this, a variety of quantization structures are examined, which strike different balances between complexity and performance. In order to ensure the best possible performance of a structured quantizer, design methods are required which balance the effects of source statistics, distortion measure and quantizer structure. To this end, training methods based on high-rate quantization theory are developed. These systems and methods are demonstrated in the context of wideband speech spectrum coding, under the Log Spectral Distortion (LSD) measure. Next, these coding tools are utilized to examine speaker-dependent wideband speech coding. In speaker-dependent coding, a separate coder is designed for each individual speaker, allowing improved performance by exploiting statistical variations between speakers. A number of issues arise in utilizing this potential, however. The

speaker-dependent coders must necessarily be designed in an online fashion, and the resultant designs distributed to other parts of the communications network. Also, robustness against incorrect speakers is required. The performance benefit of speaker-dependent coding in a CELP framework is first experimentally quantified, and a collection of methods for coping with the various implementational challenges is presented. This chapter is organized as follows: Section 1.1 provides relevant background on the topics of speech coding and quantization. Section 1.2 contains a more detailed introduction to our work on structured quantizers and high-rate training, including relevant background on high-rate quantization. Section 1.3 provides a detailed introduction to our work on speaker-dependent wideband speech coding.

1.1 Background

This dissertation considers a variety of problems in speech coding and quantization, and so a brief background on each of these topics is provided here.

1.1.1 Speech Coding and Spectrum Quantization

Most approaches to speech coding are based on an excitation/filter model, also known as Linear Predictive Coding (LPC). That is, the coder operates by first breaking the incoming speech signal into frames, typically with lengths around 20ms, and then modeling the contents of each frame as the response of an all-pole filter to some excitation signal. The parameters of this filter, known as the LPC coefficients, are computed using the well-known Levinson-Durbin algorithm to solve the Normal Equation. Thus, for each frame, there are two types of parameters to be coded: the filter parameters and the excitation parameters. A variety of techniques have been employed in each case, and a good overview of the various approaches and related theory can be found in [57]. Generally, fixed-rate quantization schemes (wherein every codeword has the same length) are used for speech

coding, reflecting the need for constant delay in a telecommunications setting. Traditional speech coding systems limit the input signal bandwidth to around 4 kHz, which is sufficient for intelligibility but results in degraded clarity and presence. In recent years, interest has grown in wideband speech coding, which utilizes a bandwidth of 8 kHz, producing improved audio quality. This extra bandwidth, in turn, requires that the order of the all-pole filter model increase to 16, compared to 10 in the traditional narrowband case. The increased sample rate likewise requires more complex excitation coding schemes. While the issue of excitation coding will be considered in Chapter 4, the majority of this dissertation will focus on the quantization of the filter parameters, also known as spectrum coding.

In most modern speech coders, the LPC coefficients are parameterized as Line Spectral Frequencies (LSFs), which have a variety of desirable properties. For example, it is simple to check the stability of a filter expressed in terms of LSFs, and the process of interpolation is well-behaved. The error between a spectrum and its quantized version is typically measured using Log Spectral Distortion (LSD), given as:

$$\text{LSD} = \sqrt{\frac{1}{2\pi}\int_{-\pi}^{\pi}\left(10\log_{10}\left(\frac{1}{|A(\omega)|^2}\right) - 10\log_{10}\left(\frac{1}{|\widehat{A}(\omega)|^2}\right)\right)^2 d\omega}$$

where $\frac{1}{A(\omega)}$ is the frequency response of the all-pole filter described by the LSF coefficients, and $\frac{1}{\widehat{A}(\omega)}$ is the frequency response of the quantized version. This measure has been found to be a good approximation of perceptual quality, and is widely used. The problem of wideband speech LSF quantization under the LSD measure is used to illustrate the performance of the proposed systems throughout this dissertation. This problem, and the similar narrowband case, have been studied extensively. See [9], [10], [11], [12], [13] or [14] for the narrowband case. An LSF quantization system is said to achieve transparent quality if it produces an average LSD of no more than 1dB, less than 1% of outliers in the 2-4dB range, and negligible larger outliers. It should be noted that some authors (such as [16]) utilize

a less strict definition of transparent quality in the context of wideband speech, typically allowing more outliers and a higher average distortion. While this work utilizes a strict definition in order to ensure that the results are applicable to a wide range of speech coding systems, results will be reported for a range of operating rates to accommodate differing transparency criteria. In [15], Gosset lattices are utilized for wideband speech LSF quantization, attaining transparent quality at around 45 bits per frame. In [16], MSVQ is employed and results in transparent quality at around 50 bits per frame. Both of these works consider only memoryless systems. In contrast, [17] presents a Pyramid VQ-based approach that exploits correlation between successive LSF vectors, attaining transparent quality around 44 bits per frame. It is noteworthy that essentially all authors have found that rates around 40 bits per vector (or higher, in the memoryless case) are required for high quality quantization of wideband LSF vectors. In such a regime, full-search VQ is impractical, both in terms of storage and complexity, and even training. Indeed, the usual requirement for speech coding systems is that they operate in real time, and with minimal storage, which requires the use of a quantizer with some special structure.

1.1.2 Background on Quantization

This section gives a brief introduction to the subject of fixed-rate quantization, a process through which a continuous-valued source is represented in terms of some finite number of codepoints. A general overview of many aspects of quantization can be found in [7] or [8]. Consider the fixed-rate quantization of a source $x \in \mathbb{R}^d$, with a probability density function $f_{\mathrm{x}}(x)$. Generally, a fixed-rate quantizer is defined by a codebook $\{\hat{x}_1, \ldots, \hat{x}_N\}$, where $N = 2^{rd}$ is the number of codepoints and r is the rate, in bits per dimension, and a mapping $Q(x)$, called the *encoder* which maps each input vector to a codepoint. Associated with each codepoint $\hat{x}_i$, then, is a *Voronoi region* $\mathcal{R}_i$ consisting of all input points that Q maps to the i-th codepoint: $\mathcal{R}_i = \{x \in \mathbb{R}^d | Q(x) = \hat{x}_i\}$. Thus, a quantizer al-

lows the input to be represented using a binary index of length rd, at the price of some error. The performance of a quantizer is measured using a distortion measure $d(x, Q(x))$, which is required to be smooth and greater than or equal to zero except when $Q(x) = x$. The most prevalent distortion measure is Mean Squared Error (MSE), given by $d(x, Q(x)) = ||x - Q(x)||^2$. A more flexible class of distortion measures is input-weighted squared error, given as $d(x, Q(x)) = ||x - Q(x)||^2_{S(x)}$, which incorporates a variable *sensitivity matrix* $S(x)$. Here, $S(x)$ is a symmetric, positive-definite matrix, called the *sensitivity matrix*. Analysis of quantization under input-weighted distortion measures can be found in [20] and [21], for the fixed-rate case, and in [22] and [23] for the variable-rate case. Further results relating to its application in speech coding can be found in [26]. A wide variety of distortion measures, and in particular Log Spectral Distortion, can be accurately approximated at high rates in this fashion, which can be seen by appeal to a Taylor series argument. Details of the calculation of the sensitivity matrix for LSD on LSF vectors are given in [20]; notably, $S(x)$ is diagonal in this case. More recently, the sensitivity matrix has been utilized to incorporate advanced auditory models into signal processing and coding applications (see [55]).

The goal in quantizer design, then, is to minimize the expected distortion $\mathrm{E_x}\, d(x, Q(x))$. For an optimal quantizer, two conditions can be derived: the Nearest Neighbor Rule and the Centroid Condition. The Nearest Neighbor Condition states that, given a fixed codebook, the optimal encoder $Q(x)$ should map each input to the codepoint with the smallest distortion. The Centroid Condition states that, given a fixed encoder, each codepoint should be the centroid of its Voronoi region, in the sense of minimizing the average distortion. The well-known Lloyd algorithm alternatingly applies the Centroid Condition and Nearest Neighbor Rule to produce an iterative quantizer design method, resulting in a quantizer that reflects the source statistics and distortion measure. However, this method is suitable only for unstructured quantizers, where one is free to move each codepoint independently, and so has limited application to the design of structured

quantizers.

1.2 High-Rate Training of Structured Quantizers

For quantizers without structure, implementing the Nearest Neighbor Rule requires computing the distortion of every possible codepoint. However, since the number of codepoints grows exponentially with the rate and dimension, this quickly becomes untenable for high-quality quantization of sources with large dimension. For example, wideband speech spectrum coding requires codebooks with trillions of codepoints to quantize spectrum vectors with $d = 16$ at sufficient quality. Such a codebook is too large to even store, let alone search in a reasonable time, and so structured quantizers must be employed to bring down the search and storage complexity. A structured quantizer works by introducing constraints on the allowed arrangements of codepoints, allowing reduced-complexity encoding. Furthermore, a structured quantizer is typically specified by a small set of parameters, easing the storage requirements. That is, the codebook is implicitly defined by the parameters and choice of structure, rather than stored explicitly as in unstructured quantization. The structures considered here are all based on the scalar transform coder, where each coefficient of a transformed input vector source is quantized independently, resulting in a complexity that grows linearly with the dimension of the source vector. Furthermore, if the scalar quantizers are implemented with companders, this type of quantizer can have complexity which does not depend on the rate. However, the constraints imposed by the structure lead to losses in performance. This can be extended to a more flexible structure using the Gaussian Mixture Vector Quantizer (GMVQ) approach, wherein M different scalar transform coders are operated in parallel, and the best output is selected using a vector quantizer. This approach is able to reflect more complex statistics and distortion measures, as well as improve the cell shapes, with a complexity that is linear in d and M. A variety of extensions of the basic transform coder and the

GMVQ are presented which extend their capabilities. In the case of the transform coder, a more flexible scalar quantizer called a Gaussian Mixture Compander is introduced, which allows flexible scalar quantizers with rate-independent complexity. For the GMVQ, methods using Hidden Markov Models are employed to build high-performance recursive quantizers, which are able to exploit dependence on previously-quantized source vectors.

For all of these structured quantizers, a design method is needed that will balance the various factors, including source statistics, distortion measure and cell shapes, to minimize the expected distortion. The Lloyd algorithm cannot be used here, because it does not take into account the constraints imposed by the structures. Moreover, any Lloyd-style method which depends on computing updates for individual codepoints will prove impractical, due to the huge codebooks of interest. What is required, then, is an expression for the expected distortion in terms of the parameters of the quantization structure, which are far less numerous and often independent of the rate. This is provided by the High-Rate theory, which gives a simplified expression for quantizer performance when the rate becomes large. This allows the design of structured quantizers to be performed through minimization of the high-rate distortion. While this is analytically tractable in certain special cases (i.e., a scalar transform coder operating on a Gaussian source under MSE distortion), a data-driven approach is adopted in which statistics are inferred directly from training data, and the various performance factors are automatically balanced to minimize high-rate distortion. The remainder of this section provides background on the high-rate analysis, with an eye towards use in training of structured quantizers.

Denote the parameters of a structured quantizer by $\theta \in \Theta$. That is, any particular setting θ implies a sequence of fixed-rate quantizers, one for each possible rate. One wishes to select the parameters so as to minimize the distortion incurred at any particular rate or, equivalently, to minimize the rate required to transmit with a particular distortion. The high rate analysis of quantizers is based on the

assumption that the rate r is sufficiently high that $f_{\mathrm{x}}(x)$ can be approximated as a constant over any cell in the quantizer. For input-weighted squared error, it is also necessary to assume that $S(x)$ is constant over any cell. Under these conditions, the expected distortion of an rate-r quantizer is given by Bennett's Integral [1]:

$$D_\theta(r) \approx 2^{-2r}\,\mathrm{E}\left(m_\theta(S,X)\lambda_\theta^{-2/d}(X)\right) \tag{1.1}$$

Where $m_\theta(S,x)$, called the *inertial profile*, and $\lambda_\theta(x)$, called the *point density*, describe the fine and coarse structures of the class of quantizers, respectively. Specifically, if $\mathcal{R}(x)$ denotes the cell containing x, and $V(x)$ its volume (Lebesgue integral):

$$\lambda(x) \;\cong\; \frac{2^{-rd}}{V(x)} \tag{1.2}$$

$$m(S,x) \;\cong\; \frac{\int_{\mathcal{R}(x)}(x-y)^{\mathsf{T}}S(x)(x-y)\,\mathrm{d}y}{V(x)^{1+\frac{2}{d}}} \tag{1.3}$$

That is, the inertial profile describes the local shape of the quantizer cells and sensitivity matrix, and the point density describes the inverse of the local cell volume. The denominator in Eq. (1.3) ensures that $m(x)$ is insensitive to scaling of $\mathcal{R}(x)$. Notice that the dependence on r in Eq. (1.1) is entirely in the exponential term, while the dependence on θ is entirely in the coefficient. Thus, the problem of finding the optimal θ becomes, for the high-rate case, that of minimizing the *distortion coefficient*:

$$\min_{\theta\in\Theta}\mathrm{E}\left(m_\theta(S,X)\lambda_\theta^{-2/d}(X)\right) \tag{1.4}$$

The standard approach to quantizer analysis is to first deal with the inertial profile, and then to handle the point density. For example, in [20], it is conjectured that an optimal quantizer will have cells that are well-approximated by ellipsoids aligned with the eigenvectors of $S(x)$, and having elongations proportional to its eigenvalues. This results in $m_{\mathsf{opt}}(x) = \frac{d}{d+2}\kappa_d^{-2/d}|S(x)|^{1/d}$, where κ_d

is the volume of a d-dimensional unit sphere. Using this expression results in an asymptotically tight lower bound on the performance of optimal quantizers. Using this approximation, one can then solve for the optimal point density, which is:

$$\lambda_{\mathsf{opt}}(x) \propto |S(x)|^{1/d} f_{\mathrm{x}}^{\frac{d}{d+2}}(x)$$

Substituting these expressions into Eq. (1.1) results in an approximation of the distortion of an optimal quantizer under source density $f_{\mathrm{x}}(x)$ and distortion sensitivity $S(x)$:

$$D_{\mathsf{opt}}(r) \approx 2^{-2r} \frac{d}{d+2} \kappa_d^{-2/d} \left(\int_{\mathbb{R}^d} \left(|S(x)|^{\frac{1}{d}} f_{\mathrm{x}}(x) \right)^{\frac{d}{d+2}} \, dx \right)^{\frac{d+2}{d}}$$

In the case of quantizers with structure, analysis of the inertial profile is more complex, because the cells may take on irregular shapes and the shapes may vary throughout the space. Moreover, the inertial profile for a structured quantizer typically depends on the parameters θ, as opposed to an optimal quantizer, wherein the inertial profile is determined entirely by the distortion measure and source dimension. The inertial profiles for various scalar quantizers, such as the transform coder, are given in [1]. Moreover, for a quantizer with a suboptimal inertial profile, the best point density may depart from $\lambda_{\mathsf{opt}}(x)$ as the arrangement of codepoints should vary to compensate for the fine structure. Thus, to utilize the high-rate approximation in training of structured quantizers, it is first necessary to derive expressions for the inertial profile and point density of the system in question as functions of the parameters. These expressions are then substituted into Eq. (1.4) to result in the high-rate training problem.

The High-Rate analysis for scalar transform coders under input-weighted squared error is presented in Chap. 2, along with the associated data-driven high-rate training algorithm, which is similar to the approaches used in Independent Components Analysis. Similarly, the high-rate analysis for GMVQ is explored in Chap. 3, and the training problem is considered. In this case, the resulting

algorithm is closely related to the well-known EM algorithm. This approach is then extended to handle the recursive case, using HMMs. Various implementational details of the systems are discussed, and their performance on the wideband speech spectrum problem is explored. In particular, HMM-based recursive quantizers using the GMVQ framework are able to achieve substantial savings compared to other approaches in the literature.

1.3 Speaker-Dependent Wideband Speech Coding

Traditional approaches to coding of speech (and, indeed, most sources) operate in a speaker-independent manner. That is, a single coder is designed and used for every speaker. This has a number of advantages, notably that only a single design process need be performed, which can be carried out ahead of time using a large multi-speaker database. However, because of variations in statistics between speakers, improved performance can be obtained by using speaker-dependent coders. The impact of speaker-dependence is well-known in other realms of speech processing, particularly speech recognition (see [2]) and enhancement (see [44]), and also very-low rate coding see [46], [43]). In the more traditional realm of CELP-based speech coding, however, this potential remains unexplored. One reason for this is that training for speaker-dependent coding must be performed in the field, where the systems have access to example input data from the individual speakers in question, requiring online training architectures. Furthermore, the resulting coder designs must be distributed to the other users to enable communication. Also, there is the requirement of robustness, which is to say that wild fluctuations in quality should be avoided when the speaker-dependent coder does not match the current user. This issue would arise if, for example, the user of a speaker-dependent coder were to hand his phone to a friend in the middle of a call.

Chapter 4 considers issues related to speaker-dependent coding in a comprehensive fashion, first experimentally quantifying the gains available from speaker-

dependent coding in a CELP framework, and then addressing the various system implementation issues. A CELP coder has three types of parameters which must be quantized in each frame: spectrum parameters, adaptive codebook parameters and fixed codebook parameters. Additionally, different distortion measures are appropriate to each type of parameter. In order to test the performance gains available in each type of parameter, then, a sufficiently flexible quantization scheme is required which can accurately reflect the variations in statistics under the appropriate distortion measures. The high-rate optimized GMVQ framework meets this requirement nicely, and so is employed to quantify the benefits of speaker-dependent coding. Gains of around 10% are found for the coding of LSF parameters, while it is shown that there is little benefit to coding of adaptive codebook parameters. The potential benefits are most dramatic in the case of the fixed codebook parameters, where savings amounting to 10-20% of the total bitrate are demonstrated. It should be noted that speaker-dependence is only one aspect that would affect performance in a real-world telecommunications system. One can also consider user-dependence, which would include not only the effects of the speaker, but also differences in background noise, acoustic environment and end-user equipment. However, due to the lack of any accepted models for such variations, and the difficulties inherent in collecting sufficient example data to quantify these effects, this work seeks only to quantify benefits due to variations in the speaker. This can be seen as a first step in quantifying the benefits in the full user-dependent case, providing a baseline estimate. Moreover, the various techniques for implementation that are presented would apply directly to the user-dependent case.

The most pressing challenge for speaker-dependent systems is the requirement for online training and dissemination of the resulting coder designs. In light of this, a variety of training architectures are presented, which strike different balances between training complexity, transmission overhead, and performance. An important problem in this context is learning on quantized data, which allows the training to be carried out in locations remote from the end user. Quantizers us-

ing scalar transform coders, and in particular the GMVQ, can cause problems in the context of learning on quantized data, as they often place codepoints in subspaces. To correct this, a modification to the decoder is presented which avoids this problem. Then, the penalty for using quantized data in learning is quantified as a function of the encoding rate, and found to be small for rates of interest in spectrum coding. Another important technique is recursive learning, wherein the learning takes place on a frame-by-frame basis. This eliminates the storage complexity associated with the learning process, and also enables adaptive operation. Techniques for recursive learning of GMVQ systems are presented, and shown to achieve the same performance as traditional batch learning. In order to provide robustness against incorrect speaker models, a safety-net quantization approach is examined. This method can be naturally incorporated into the GMVQ framework, and a modification to the GMVQ learning methods is presented which allows precise trade-off between robustness and performance.

2 High-Rate Design of Transform Coders with Gaussian Mixture Companders

This chapter considers the problem of designing fixed-rate scalar transform coders. Transform coding is a popular method for quantizing vectors of data using only scalar quantizers. As a result of this structure, transform coders have a number of desirable properties, such as small storage requirements and low coding complexity. The price for these features is that the transform coder suffers from inferior performance as compared to a full-search VQ, or other more flexible quantizers. Figure 2.1 illustrates the structure of a transform coder using companding scalar quantizers.

Many approaches to transform coding depend on fixing the transform in some way, often by assuming the source is Gaussian (see [1], [2], [3]), in which case the Karhunen-Loeve Transform (KLT) is the optimal transform, at least at high rates. In other cases, only certain convenient transforms are considered: an example would be the DCT in image coding [4]. While the problem of designing such a structure (i.e., selecting T, K_i's, g_i's and h_i's) is well-understood for the case of multivariate Gaussian data and Mean Squared Error (MSE) distortion measure, this paper considers a more general problem in which the distribution is unknown, and presumably non-Gaussian. As discussed in [5], the KLT may be

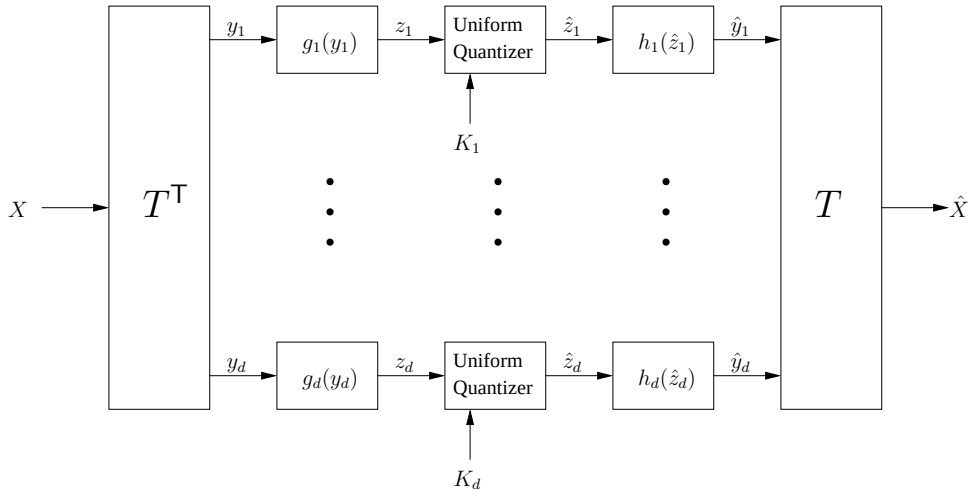

Figure 2.1: Transform coder using companding scalar quantizers. T is an orthogonal matrix, g_i and h_i are (nonlinear) compressor and expander functions, and the parameters K_i specify the allocation of codepoints between the scalar companders.

a very poor choice for non-Gaussian sources. Additionally, this paper considers more general input-weighted distortion measures, which can approximate a wide variety of practical distortion measures such as Log Spectral Distortion (LSD). To accommodate these concerns, an algorithm is developed to set the parameters of the system using a data-driven design technique that automatically balances the source statistics, distortion measure, and structure of the transform coder to minimize high-rate distortion. To allow for unknown source statistics, a flexible compander system based on Gaussian Mixtures is presented. Modifications to the scheme for operation at moderate rates, utilizing unstructured scalar quantizers, are also discussed.

The problem of designing a transform coder for minimum distortion from a database has also been considered by Archer and Leen in [6]. There are several distinctions between that work and the ideas presented here. First, this chapter focuses on fixed-rate systems, whereas [6] covers the variable-rate case. Next, this work considers a more general distortion measure, where Archer and Leen focused only on MSE (although their work could be extended in a straightforward way).

Finally, this system is rate-independent in that it admits operation at arbitrary rates with no additional storage or training requirements, whereas the approach in [6] must be repeated for every distinct operating rate. The problem of learning the optimal transform for a variable-rate transform coder was also considered in [3] for a Gaussian source with unknown covariance. In this work, however, Gaussianity will not be assumed. Another similar previous work is [4], which considers the variable-rate transform coding of images and uses Gaussian mixtures to model the marginal source densities. In that work, however, the transform is considered as fixed and only the problem of learning the component scalar quantizers is addressed.

To illustrate the performance of the proposed system, this chapter first presents a toy problem. This problem is similar to Example V.2 in [5] (which demonstrates the suboptimality of the KLT), and demonstrates that the high-rate design scheme will identify the optimal transform coder settings, including the effects of both the statistics and distortion measure. To examine real-world performance, the example of wideband speech LSF quantization with the LSD measure, is then considered. The transform coder has many desirable implementational properties, as one need only store the system parameters, which are typically far less numerous than the aggregate number of codepoints in the corresponding codebook. As will be seen in Section 2.5, optimized transform coder systems are able to achieve comparable performance to MSVQ. Furthermore, if the scalar quantizers are implemented with companders the encoding complexity becomes rate-independent. Since the parameters of the system are also independent of the rate of operation, the companding transform coder is able to operate at practically arbitrary rates with no additional storage or complexity requirements. Additionally, the performance of transform coders with high-rate optimized transforms and allocations, but Lloyd-optimized codebooks, is examined. Such systems offer superior average distortion performance as compared to companders, but turn out to result in degraded outlier performance. Their application also demonstrates the utility of high-rate design of the transform at moderate rates where companders

suffer from poor performance.

This chapter is organized as follows: Section 2.1 reviews the high-rate analysis of transform coders. Previous results are extended to the case of input-weighted squared error, and the implications for the design problem are discussed. Section 2.2 details the data-driven design algorithm, which designs the system by minimizing estimated high-rate distortion. Section 2.3 describes modifications to the coding and training schemes for operation at moderate rates, using unstructured scalar quantizers. Section 2.4 details the implementation of the proposed systems, with a special emphasis on the GM compander. Section 2.5 presents experimental results utilizing the proposed system and associated design algorithm for a toy problem and for wideband speech LSF quantization. Section 2.6 contains a discussion of the system and its performance, while Section 2.7 gives the details of an integral that is utilized in the high-rate analysis of Section 2.1.

2.1 Scalar Quantizers and Transform Coders

As discussed in Chapter 1, the first step in high-rate analysis is to consider the inertial profile. For many systems of intermediate complexity, no closed-form expression is available for the inertial profile. However, certain strongly structured quantizers are simple enough to allow closed-form analysis. One such case is the transform coder, in which all cells are hyperrectangles. Due to this structure, the transform coder suffers from space-filling loss, oblongitis, and a limited ability to exploit dependence between the elements of X (see [25] for a detailed explanation).

The high-rate approximation for a transform coder under MSE is given in [1]. Here, this analysis is reviewed and extended to the case of input-weighted squared error. First, consider the case of a product quantizer, or the special case of a transform coder with $T = I$. Such a quantizer has cells that are hyperrectangles, aligned with the coordinate axes: $\mathcal{R} = \mathcal{R}_1 \times \ldots \times \mathcal{R}_d$. If $\lambda_{\theta_i}(y_i)$ are the point densities of the scalar quantizers, and K_i are the numbers of levels assigned to

each of them, the high-rate approximation is that a cell centered at a point y, denoted $\mathcal{R}(y)$, has side lengths given by $\mathcal{R}_i(y_i) = (K_i\lambda_{\theta_i}(y_i))^{-1}$. The volume of a such a cell, then, is $V(y) = \prod_{i=1}^{d}(K_i\lambda_{\theta_i}(y_i))^{-1} = (K\lambda_\theta(y))^{-1}$, where K is the total number of codepoints (i.e., 2^{dr}) and $\lambda_\theta(y)$ is the total point density. Using the result in Section 2.7 to evaluate the numerator of Eq. (1.3), the inertial profile for a product quantizer is given by:

$$m_\theta(S, y) = \frac{(K\lambda_\theta(y))^{2/d}}{12} \sum_{i=1}^{d} s_{ii}(y) \left(K_i\lambda_{\theta_i}(y_i)\right)^{-2} \tag{2.1}$$

Where $s_{ii}(y)$ is the i-th diagonal element of $S(y)$. This paper will parameterize the level allocations in terms of new variables β_i as follows:

$$K_i = K^{1/d}\beta_i \left(\prod_{j=1}^{d} \beta_j\right)^{-1/d}$$

Where $\beta_i > 0$, insuring that $K_i > 0$. Notice that the form of this parametrization insures that $\prod_{i=1}^{d} K_i = K$. For training purposes, the constraint that the K_i's must be integers is ignored. This is done because such a constraint is difficult to include in the estimation procedure, and because it would make the training depend on the exact rate of operation. Instead, a pruning algorithm is applied to meet the constraint when implementing the coder, as described in Section 2.4. Combining these results, and substituting into Eq. (1.1) gives:

$$D_{prod} \cong 2^{-2r}\frac{1}{12}\left(\prod_{j=1}^{d} \beta_j\right)^{2/d} \sum_{i=1}^{d} \beta_i^{-2}\, \mathrm{E}\left(s_{ii}(Y)\lambda_{\theta_i}^{-2}(Y_i)\right) \tag{2.2}$$

As described in [1], this analysis can easily be extended to the transform coder by defining $Y = T^{\mathsf{T}}X$ and noticing that, since T is orthogonal, $\lambda_\theta(x) = \lambda_\theta(Ty)$ and $\lambda_{\theta_i}(y_i) = \lambda_{\theta_i}(t_i^{\mathsf{T}}x)$, where t_i is the i-th column of T. As discussed in [26], it is easily seen that $S(y) = T^{\mathsf{T}}S(x)T$, and so $s_{ii}(y) = ||t_i||^2_{S(x)}$. Substituting these relations into Eq. (2.2) gives the approximation for a transform coder:

$$D_\theta \cong 2^{-2r} \frac{1}{12} \left(\prod_{j=1}^{d} \beta_j \right)^{2/d} \sum_{i=1}^{d} \beta_i^{-2} \, \mathrm{E}\left(||t_i||^2_{S(X)} \lambda_{\theta_i}^{-2}(t_i^\mathsf{T} X) \right) \tag{2.3}$$

Notice that this objective function is similar to one used in Independent Components Analysis or Blind Source Separation (see [27]). In those settings, one seeks to minimize the sum of the marginal entropies of the transformed components (and, hence, their mutual information):

$$\min_{\theta, T} \sum_{i=1}^{d} \mathrm{E}\left(\log \lambda_{\theta_i}^{-1}(t_i^\mathsf{T} X) \right)$$

In that case, the density functions λ_{θ_i} are interpreted as the (model) probability densities of the components, rather than as the point densities of scalar quantizers applied to them. Moreover, the transform coder design case has the added elements of level allocation (the β_i terms) and weighting due to the distortion measure. Nevertheless, the basic structure of the objective function is similar in both cases: a sum of expectations of a decreasing, convex function of each transform coefficient's density. The differences between the two objective functions emphasize that in fixed-rate transform coder design, unlike ICA, one does not necessarily seek a transform which results in the components being independent in the usual statistical sense, but rather one which results in the components being amenable to independent quantization. The similarity between the two objective functions, on the other hand, suggests that their respective solutions may be similar, and informs the widespread intuition that independent transform coefficients are desirable in fixed-rate transform coding.

2.1.1 Point Densities

To facilitate a variational design algorithm, this work utilizes a specific parametric form for the point densities implemented by the scalar quantizers (i.e., the compressor and expander functions). It would be ideal for optimization purposes if the point densities were such that they resulted in a positive quadratic

when taken to the power -2, as in Eq. (2.8). However, the function $(C + x^2)^{-1/2}$ does not have a finite integral over $\mathbb{R}$, and so there is no point density that would result in such a quadratic. As will become apparent in the next section, this complicates the estimation problem. In the case of a transform coder operating in the large-d case, it could be argued that the optimal point densities should be Gaussian, since the Central Limit Theorem would imply that $t_i^{\mathsf{T}} X$ would approach a Gaussian for each i. However, it is not clear that $T = I$, or something close to it, is not the optimal setting for some distributions $f_{\mathrm{x}}(x)$ (see [5]), in which case the Central Limit Theorem would not be in effect. Moreover, simulation results have shown that the marginal densities of 16-dimensional speech spectrum vectors fail statistical tests for Gaussianity, both for naive and optimized settings of T. To allow for unknown statistics, this work employs a flexible class of point densities which are mixtures:

$$\lambda_{\theta_i}(y_i) = \sum_{m=1}^{M} \alpha_{im} \lambda_{im}(y_i)$$

This class of point densities can, as M grows large, approximate a wide variety of densities. Even for low values of M, mixtures are able to model features such as multimodality and skew. As will be discussed in Section 2.4, mixture point densities require an iterative decoder in order to operate a compander. This work will also assume that the component point densities are Gaussian: $\lambda_{im}(y_i) = N(y_i | \mu_{im}, \sigma_{im}^2)$ Thus, we have $\theta = \{T, \beta, \alpha, \mu, \sigma^2\}$ where $T^{\mathsf{T}} T = I$, $\beta_i > 0$, $\sum_{m=1}^{M} \alpha_{im} = 1$ and $\sigma_{im}^2 > 0$. An important point to note is that the use of Gaussian Mixtures implies that the tails of the point densities will be Gaussian. In situations where the data has heavier tails, this is inappropriate. However, for sources with finite support, such as images or speech spectrum vectors, this issue is of less concern. While the Gaussian Mixture point densities will always place some finite mass outside the support region, this effect quickly becomes negligible as M increases, as will be seen in Section 2.5.

2.2 Data-Driven Transform Coder Design

The problem at hand is to minimize the high-rate distortion of a transform coder, as given in Eq. (2.8). As described in the preceding subsection, mixture point densities are assumed. Suppose that one has no knowledge of the distribution except for a set of samples $\{x_1, \ldots, x_N\}$, drawn i.i.d. from $f_{\mathrm{x}}(x)$. In this case, the Strong Law of Large Numbers can be invoked to replace the expectations in Eq. (2.8) with averages over the data. After dropping terms that do not depend on the parameters, the data-driven high-rate design problem is given by:

$$\min_{\theta \in \Theta} \left(\prod_{j=1}^{d} \beta_j \right)^{2/d} \sum_{i=1}^{d} \beta_i^{-2} \sum_{n=1}^{N} ||t_i||^2_{S(x_n)} \left(\sum_{m=1}^{M} \alpha_{im} \lambda_{im} (t_i^{\mathsf{T}} x_n) \right)^{-2} \tag{2.4}$$

where the quantity being minimized is now an estimate of the distortion coefficient. In order for this approximation to be valid, it must satisfy a number of requirements. First, the expected distortion must be finite for any θ under consideration. Moreover, one would like the variance of the integrands in Eq. (2.8) be finite, although this is not strictly necessary. Practically speaking, these requirements mean that one will need to use good initializers. While it can be difficult to ensure these requirements are met in the general case, for sources with finite support regions (such as speech spectrum vectors or images), a sufficient condition is that the point densities be greater than 0 over the support. Note that this is satisfied by the Gaussian Mixture point densities considered here. While this property guarantees the analytical validity of the approximation, one still requires good initializers to keep the quantities inside numerical resolution, and to aid in finding a global optimum. Lastly, one should use large enough sets of training data to avoid overtraining. Note that this design procedure is intended to be carried out off-line, and so its complexity does not come to bear on the operation of the resulting transform coder. Also, while successive LSF frames are actually correlated, this dependence is ignored here, as all of the systems under consideration are memoryless; only intra-frame dependencies are exploited.

It is difficult to directly optimize Eq. (2.9) over all parts of θ simultaneously. As such, this chapter proposes an iterative algorithm that alternatingly optimizes over subsets of parameters while holding the others fixed. Specifically, each iteration first optimizes over the transform, then over the point density parameters, and then over the level allocations. For the transform, this work utilizes the steepest descent approach proposed by Manton in [28], which is briefly reviewed in Section 2.2.1. The point density parameters are optimized with an extension of the EM algorithm, which is presented in Section 2.2.2. The level allocation parameters are handled by a standard Lagrange multiplier approach, as discussed in Section 2.2.3. The overall optimization algorithm in summarized in Section 2.2.4.

2.2.1 Transform Optimization

First, consider the problem of optimizing the transform T. While analytically tractable in the case that X is a multivariate Gaussian and MSE is the distortion measure, it is not known generally how T ought to be set [5]. Thus, this work proposes an iterative numerical approach. The problem in this case is:

$$\min_{T^{\mathsf{T}}T=I} \sum_{i=1}^{d} \beta_i^{-2} \sum_{n=1}^{N} ||t_i||^2_{S(x_n)} \left(\sum_{m=1}^{M} \alpha_{im} N(t_i^{\mathsf{T}} x_n | \mu_{im}, \sigma_{im}^2) \right)^{-2} \tag{2.5}$$

To solve this problem, one can turn to the generic algorithms for optimization over unitary matrices presented in [28]. Specifically, this paper utilizes the constrained steepest-descent method, which requires evaluation of the derivative of the objective function. Noting that the objective is a sum of d terms, each of which depends only on a single column t_i, one may write each column of the derivative G as follows:

$$\begin{aligned} G_i &= 2\beta_i^{-2} \sum_{n=1}^{N} \left(\sum_{m=1}^{M} \alpha_{im} N(t_i^{\mathsf{T}} x_n | \mu_{im}, \sigma_{im}^2) \right)^{-2} \\ &\quad \times \left\{ S(x_n) t_i + ||t_i||^2_{S(x_n)} \sum_{m=1}^{M} \left(r_{mn} \frac{t_i^{\mathsf{T}} x_n - \mu_{im}}{\sigma_{im}^2} \right) x_n \right\} \end{aligned}$$

Where r_{mn} is as defined in Eq. (2.7). Having evaluated this derivative at the current estimate T, one then finds Z, the steepest descent direction in the tangent space to the constraint manifold at T, as follows:

$$Z = TG^{\mathsf{T}}T - G$$

An improved estimate $\widehat{T}$, then, is found via a projected linesearch approach along the descent direction Z. That is, one begins with some stepsize γ, which implies a new estimate of $T + \gamma Z$. However, this new matrix may not be orthonormal, so it is projected it onto the Stiefel manifold utilizing an SVD. That is, if $U\Sigma V^{\mathsf{T}} = T + \gamma Z$, the projection is $\widehat{T} = UV^{\mathsf{T}}$. The stepsize is then varied and the process repeated until a suitable stepsize is identified (i.e., small enough to ensure convergence, but large enough that convergence is not too slow). To save computation, the Frobenius norm of Z is checked at each iteration and, if it is found to be small, no update is performed.

2.2.2 Point Density Optimization

Notice that, due to the structure of the transform coder, the overall objective function, Eq. (2.9), is a sum over functions of the different scalar quantizers, and so they may be optimized independently. For the i-th scalar quantizer, the optimization problem is:

$$\min_{\theta_i} \sum_{n=1}^{N} ||t_i||^2_{S(x_n)} \left(\sum_{m=1}^{M} \alpha_{im} \lambda_{im}(t_i^{\mathsf{T}} x_n) \right)^{-2} \tag{2.6}$$

Such a problem can be approached with an extension of the EM algorithm. Where conventional EM applies Jensen's inequality to a logarithm to construct a bound on the objective function (see [29]), this work does the same with the power function $(\cdot)^{-2}$. That is, let r_{mn} be some positive numbers such that $\sum_{m=1}^{M} r_{mn} = 1$, $\forall n$. Then:

$$\begin{aligned}
\sum_{n=1}^{N} ||t_i||^2_{S(x_n)} \left(\sum_{m=1}^{M} \alpha_{im} \lambda_{im}(t_i^{\mathsf{T}} x_n) \right)^{-2} &= \sum_{n=1}^{N} ||t_i||^2_{S(x_n)} \left(\sum_{m=1}^{M} r_{mn} \frac{\alpha_{im} \lambda_{im}(t_i^{\mathsf{T}} x_n)}{r_{mn}} \right)^{-2} \\
&\leq \sum_{n=1}^{N} ||t_i||^2_{S(x_n)} \sum_{m=1}^{M} r_{mn} \left(\frac{\alpha_{im} \lambda_{im}(t_i^{\mathsf{T}} x_n)}{r_{mn}} \right)^{-2} \\
&= \sum_{n=1}^{N} ||t_i||^2_{S(x_n)} \sum_{m=1}^{M} r_{mn}^3 \left(\alpha_{im} \lambda_{im}(t_i^{\mathsf{T}} x_n) \right)^{-2}
\end{aligned}$$

Notice that the bound is in the form of a sum over m, which will ease the optimization problem. The idea is to use this construction iteratively, selecting r_{mn} based on a previous estimate $\widehat{\theta}_i$. One would like to set the bound as tightly as possible; specifically, one would like to attain equality in the second line above when $\theta_i = \widehat{\theta}_i$. It can easily be seen that this criterion is met by the same setting as in conventional EM:

$$r_{mn} = \frac{\alpha_{im} \lambda_{im}(t_i^{\mathsf{T}} x_n)}{\sum_{p=1}^{M} \alpha_{ip} \lambda_{ip}(t_i^{\mathsf{T}} x_n)} \tag{2.7}$$

It can also be shown that using Eq. (2.7) results in the bound having the same derivative as the underlying objective function at $\theta_i = \widehat{\theta}_i$. Thus, one may construct an iterative optimization procedure for θ_i by starting with some initial guess θ_i^0 and then alternating between applying Eq. (2.7) and selecting a new estimate by optimizing the resulting bound. The next two subsubsections discuss optimization of the bound over the mixture weights and density parameters, respectively.

Mixture Weights

Recalling the constraint $\sum_{m=1}^{M} \alpha_{im} = 1$, form the usual Lagrange multipliers objective:

$$f(\alpha_i) = \sum_{m=1}^{M} \alpha_{im}^{-2} c_{im} + \lambda \sum_{m=1}^{M} \alpha_{im}$$

Taking the derivative and setting it to zero results in the following relations:

$$\alpha_{im} = \sqrt[3]{\frac{2c_{im}}{\lambda}} \ , \forall m$$

Substituting these results into the constraint and solving for λ results in the optimal setting for α_i:

$$\begin{aligned} \alpha_{im} &= \frac{\sqrt[3]{c_{im}}}{\sum_{p=1}^{M} \sqrt[3]{c_{ip}}} \ , \forall m \\ &= \frac{\sqrt[3]{\sigma_{im}^2 \sum_{n=1}^{N} ||t_i||_{S(x_n)}^2 r_{mn}^3 e^{\sigma_{im}^{-2}(x_n-\mu_m)^2}}}{\sum_{p=1}^{M} \sqrt[3]{\sigma_{ip}^2 \sum_{n=1}^{N} ||t_i||_{S(x_n)}^2 r_{pn}^3 e^{\sigma_{ip}^{-2}(x_n-\mu_p)^2}}} \ , \forall m \end{aligned} \tag{2.8}$$

Component Means and Variances

Optimization of the means and variances of each component is not possible in closed form for the Gaussian case. It has been observed in practice that Newton's method works fine on both problems, provided a reasonable initializer is used. Applying Newton's method gives the following update iteration for μ_{im}, based on the previous guess $\hat{\mu}_{im}$:

$$\mu_{im} = \hat{\mu}_{im} + \frac{\sum_{n=1}^{N} ||t_i||_{S(x_n)}^2 r_{mn}^3 e^{\sigma_{im}^{-2}(x_n-\hat{\mu}_{im})^2}(x_n - \hat{\mu}_{im})}{\sum_{n=1}^{N} ||t_i||_{S(x_n)}^2 r_{mn}^3 e^{\sigma_{im}^{-2}(x_n-\hat{\mu}_{im})^2}\left(1 + 2\sigma_{im}^{-2}(x_n - \hat{\mu}_{im})^2\right)} \tag{2.9}$$

Similarly, we obtain a recursion for σ_{im}^2 in terms of $\hat{\sigma}_{im}^2$:

$$\sigma_{im}^2 = \hat{\sigma}_{im}^2 \left(1 - \gamma \frac{\sum_{n=1}^{N} ||t_i||_{S(x_n)}^2 r_{mn}^3 e^{\hat{\sigma}_{im}^{-2}(x_n-\mu_{im})^2}(1 - \hat{\sigma}_{im}^{-2}(x_n - \mu_{im})^2)}{\sum_{n=1}^{N} ||t_i||_{S(x_n)}^2 r_{mn}^3 e^{\hat{\sigma}_{im}^{-2}(x_n-\mu_{im})^2}\hat{\sigma}_{im}^{-2}(x_n - \mu_{im})^2}\right) \tag{2.10}$$

Here, γ is a stepsize parameter, initialized to be 1, that is used to search for positive settings of σ_{im}^2. That is, it is iteratively multiplied by $\frac{1}{2}$ until the resulting value of σ_{im}^2 is positive.

2.2.3 Level Allocation Optimization

This subsection considers the problem of optimizing the level allocation β. With the other parameters fixed, the problem becomes (after taking the logarithm):

$$\min_{\beta>0} \frac{2}{d}\sum_{j=1}^{d} \log \beta_j + \log \sum_{i=1}^{d} c_i \beta_i^{-2}$$

Taking the derivative of this expression and setting it to zero gives the following equations:

$$\frac{\beta_i^{-2} c_i}{\sum_{j=1}^{d} \beta_j^{-2} c_j} = \frac{1}{d} \ , \forall i$$

In other words, one should choose the allocation that balances the distortion of each coefficient in the sense that $\beta_1^{-2}c_1 = \beta_2^{-2}c_2 = \ldots = \beta_d^{-2}c_d$. This can be accomplished by setting $\beta_1 = 1$ and then applying the equation $\beta_i = \sqrt{\frac{c_i}{c_1}}$, $\forall i \geq 2$. That is:

$$\beta_i^* = \begin{cases} 1 & , \quad i = 1 \\ \sqrt{\frac{\sum_{n=1}^{n} ||t_i||^2_{S(x_n)} \left(\sum_{m=1}^{M} \alpha_{im} N(t_i^\mathsf{T} x_n | \mu_{im}, \sigma^2_{im})\right)^{-2}}{\sum_{n=1}^{n} ||t_1||^2_{S(x_n)} \left(\sum_{m=1}^{M} \alpha_{1m} N(t_1^\mathsf{T} x_n | \mu_{1m}, \sigma^2_{1m})\right)^{-2}}} & , \quad i \in \{2, \ldots, d\} \end{cases} \tag{2.11}$$

Thus, an optimal allocation results in each coefficient contributing equally to the total distortion. It is instructive to consider the distribution of β's, which indicates the *energy compaction* of the transform coder. This property, which is typical of the KLT, refers to instances in which most of the energy is confined to a small number of the transform coefficients. While energy compaction is generally considered beneficial in coding problems, it also comes to bear on the rates at which the high-rate assumptions are appropriate. Recall that the derivations in Section 2.1 assumed that all of the scalar quantizers were operating at high rates. With low energy compaction, each quantizer will receive around the same number of levels, so an overall rate of $r > 3$ (or possibly lower) should be sufficient. In

cases where energy compaction is large, a higher rate is required to ensure that most (if not all) quantizers are in high-rate.

2.2.4 Summary of Data-Driven Transform Coder Design Algorithm

The minimum high-rate distortion design algorithm is summarized below:

1. Start with an initial guess of the transform T.
2. Initialize the other parameters as follows:
   - Compute the transformed data $Y = T^{\mathsf{T}}X$
   - For each transform coefficient i, initialize the point density parameters by applying the K-means algorithm to the set of scalar data $\{y_i\}$.
   - Set all β_i's to 1.
3. Optimize the transform by applying the steepest descent method described in Section 2.2.1
4. Optimize the point density and level allocation parameters as follows:
   - Compute the transformed data $Y = T^{\mathsf{T}}X$
   - Perform one or more iterations of the extended EM algorithm of Section 2.2.2 to optimize the point densities:
     - Compute r_{mn} using Eq. (2.7)
     - Update mixture weights using Eq. (2.8)
     - (Optional) Recompute r_{mn} using Eq. (2.7) to reflect updated weights
     - Update component means using one or more iterations of Eq. (2.9)
     - (Optional) Recompute r_{mn} using Eq. (2.7) to reflect updated means
     - Update component variances using one or more iterations of Eq. (2.10)
   - Update level allocation (β_i's) using Eq. (2.11)

5. Return to Step 3 unless convergence has been reached.

To initialize the algorithm, one needs an initial guess of the transform. Two obvious choices are the KLT and identity matrices, which will always be included in this work. One can easily generate more guesses by creating random Givens rotations and applying them to other initializers, to each other, or just using them as initializers directly. It will be especially important to try a wide variety of initializers for this problem, as the flexibility of the mixture point densities will give rise to many local minima. Given an initial transform, the K-means algorithm is applied to each dimension of the transformed data to initialize the point densities. Specifically, the mixture weights are set according to the proportion of points assigned to each cluster, and the means are initialized as the sample means of each cluster. The variances are initialized as 3 times the sample variances of each cluster, as one expects the optimal point density should have larger variance than the probability density.

It should be noted that using very flexible point densities (i.e., large M) leads to the proliferation of local optima. That is, once the point densities have become tightly optimized for a particular choice of T, one will be "stuck" with that that transform. This is because changing T while leaving the point densities fixed will result in mismatch between the transformed data's marginal statistics and the point densities. This is a drawback of the alternating minimization approach, and suggests that one perform only a single iteration of the point density optimization step at each iteration. The hope is that if the procedure takes only incremental steps during each part of the iteration, it will more closely approximate joint optimization of the parameters, which is not as sensitive to these effects. Further, it is often useful to initialize training in a hierarchical manner. That is, first perform the training for the case of $M = 1$, which has much less sensitivity to local minima, then utilize the resulting transform to initialize the $M = 2$, along with the above-described initializers, and so on with the higher orders.

2.3 Modifications for Operation at Moderate Rates

While the high-rate approach described in the previous sections has a number of advantages, it is often the case in practice that the desired operating rate is below the high-rate regime. This is of particular concern in the case of transform coders with high energy compaction, as described in Section 2.2.3. A major source of loss at lower rates is the performance of quantizers based on companders. At moderate rates, better performance can be achieved by unstructured scalar quantizers. Utilizing such a system makes the operational complexity rate-dependent and, crucially, requires a rate-dependent training algorithm. It is important to note that the high-rate training scheme provided in the previous sections does not directly determine the codebooks of the scalar quantizers, but rather their high-rate structures (i.e., point densities). It is only through the additional assumption of a compander implementation that the design algorithm specifies an actual codebook. In the case of unstructured scalar quantizers, the high-rate design algorithm gives us an idea of how the optimal scalar quantizers ought to look, but does not specify the actual codebooks. This begs the question of how such a system ought to be designed.

The major tool for unstructured quantizer design is the well-known Lloyd algorithm. As described below, a weighted Lloyd algorithm can be employed to train the scalar quantizers with the other parameters fixed. However, it is unclear how to simultaneously optimize the codebooks, transform and level allocation in a rate-dependent fashion. One could construct a brute-force search algorithm, which evaluates various combinations of transforms, allocations and Lloyd-optimized codebooks, but such an approach is beyond the scope of this paper. Instead, a hybrid approach is suggested, wherein the estimation is initially carried out using the high-rate approach detailed in the Section 2.2, and then the scalar quantizers are optimized using the weighted Lloyd algorithm. The transform and allocation are left fixed at their high-rate optimized values, and the

compander codebooks are used as initializers in the Lloyd process. The primary advantage to using the high-rate approach in training the transform and allocation is that it provides a simple parametric expression for the expected distortion, enabling gradient-based techniques to be applied to the transform optimization process. Additionally, the process of optimizing the point densities and estimating their associated distortions is much less computationally intensive than the Lloyd optimization procedure.

2.3.1 Input-Weighted Lloyd Algorithm for Scalar Quantizers

Note that, as indicated in Eq. (2.8), the objective function in the scalar quantizer design is affected by an input-weighting term. That is, the objective function for the i-th scalar quantizer is:

$$\sum_{n=1}^{N} ||t_i||^2_{S(x_n)} \left(t_i^{\mathsf{T}} x_n - Q_{K_i}(t_i^{\mathsf{T}} x_n)\right)^2$$

The notation $Q_{K_i}(\cdot)$ denotes the output of a K_i-point scalar quantizer, characterized by K_i codepoints $\widehat{y}_{ik}$. It has been established in [30] that the generalized Lloyd algorithm will converge in this case. This section, then, simply reviews the Nearest-Neighbor and Centroid relations as they apply in the input-weighted squared error case. For a scalar quantizer, there is no change to the classic Nearest-Neighbor result: input points should be quantized to the codepoint nearest to them (in the usual Euclidean sense). To handle input-weighted distortion, a simple weighting term is included in the Centroid step:

$$\widehat{y}_{ik} = \frac{\sum_{n \in \mathcal{R}_{ik}} ||t_i||^2_{S(x_n)} t_i^{\mathsf{T}} x_n}{\sum_{n \in \mathcal{R}_k} ||t_i||^2_{S(x_n)}}$$

where the notation $n \in \mathcal{R}_{ik}$ indicates that the i-th transform coefficient of x_n has been quantized to the k-th codepoint of the i-th codebook.

2.4 Implementation of GM Transform Coder

This section discusses the implementation of a transform coder with GM point densities. In particular, this work uses a compander implementation, which results in a complexity that is independent of the rate r. Note that the overall coding complexity of the transform coder is quite low, consisting of a linear transform, a bank of scalar companders, and final inverse linear transform. To operate such a system, one first must compute the level allocation for the desired rate.

2.4.1 Level Allocation

To compute the level allocation, the pruning scheme presented in [31] is used, and so is reviewed here. One first computes the unconstrained level allocations, and rounds them up:

$$\widehat{K}_i = \left\lceil K^{1/d} \beta_i \left(\prod_{j=1}^{d} \beta_j \right)^{1/d} \right\rceil$$

A pruning algorithm is then applied to make the resulting total number of codepoints as close to K as possible without exceeding it. The pruning algorithm works as follows: suppose that the dimensions have been ordered in increasing β_i. Then, beginning with the first dimension, calculate the total number of codepoints that have been assigned, $\widehat{K} = \prod_{i=1}^{d} \widehat{K}_i$, and an adjustment for the current dimension i: $\widehat{K}_i = \widehat{K}_i - \left\lfloor \widehat{K}_i \left(1 - \frac{K}{\widehat{K}} \right) \right\rfloor$. This process is then repeated in turn for each of the dimensions, and then one level is subtracted from the last dimension, resulting in an allocation that is guaranteed to be less than the total allowed. While iterative algorithms can, in principle, achieve superior allocations, it has been observed that this method, which requires only a single pass, results in good allocations in practice. In particular, the proportion of "wasted" codepoints becomes very small as K becomes large.

2.4.2 Transform Coder

The transform coder operates by first applying the transform to the input vector X: $Y = T^{\mathsf{T}}X$. Next, the coder quantizes each component of Y with a companding scalar quantizer with point density $\lambda_i(y_i)$. A compander works by first applying a nonlinear *compressor function*:

$$z_i = g(y_i) = \int_{-\infty}^{y_i} \lambda_i(\tau)\,\mathrm{d}\tau$$

The resulting z_i is then quantized with a uniform (on $[0,1]$) scalar quantizer resulting in $\hat{z}_i$. Since a uniform scalar quantizer can be implemented without any searches, using rounding operations, the resultant system has rate-independent complexity. The quantized value is then input to the inverse of the compressor function, called the *expander function*:

$$\hat{y}_i = h(\hat{z}_i) = g^{-1}(\hat{z}_i)$$

Finally, then, the quantized vector is given by applying the inverse transform:

$$\widehat{X} = T\widehat{Y}$$

2.4.3 Evaluating the Compander Functions

In the classical case of Gaussian point densities, it is easy enough to compute the compressor and expander functions, as they are the Gaussian cdf and its inverse, respectively, which are very standard in numerical libraries. For the case of a GM density, the compressor function is easy to evaluate:

$$\begin{aligned} g(y_i) &= \int_{-\infty}^{y_i} \sum_{m=1}^{M} \alpha_{im} N(\tau|\mu_{im}, \sigma_{im}^2)\,\mathrm{d}\tau \\ &= \sum_{m=1}^{M} \alpha_{im} \Phi_{im}(y_i) \end{aligned}$$

where $\Phi_{im}(y_i)$ denotes the cdf of the im-th Normal. Also, let $N_{im}(y_i) = N(y_i|\mu_{im}, \sigma_{im}^2)$ denote the pdf of the im-th component. However, the expander function is difficult to evaluate, since one cannot interchange the inverse with the summation.

Iterative Decoder

To get around the difficulty with the expander function, this work proposes an iterative procedure for computing $h(z_i)$. Such a procedure will begin with an initial guess y_i^0 and then perform a number of iterations to improve this estimate. As is well known, Newton's method provides quadratic convergence if one supplies a suitable initializer. Thus, a simple method for selecting a suitable initializer for any possible value of z_i is needed. Such an initializer can be provided by partitioning $g(y_i)$ into concave and convex regions, and assigning a different initializer depending on which region z_i falls into. The method of partitioning is shown in Figure 2.2. Details of this scheme are found below.

The basic problem is, given some $z_i \in [0, 1]$, to find y_i such that:

$$\kappa(y_i) \triangleq \sum_{m=1}^{M} \alpha_{im} \Phi_{im}(y_i) - z_i = 0 \tag{2.12}$$

Note that $\kappa(y_i)$ is a monotonically-increasing function, and so only a single y_i^* will exist that satisfies the above. A classic technique for solving such a root-finding problem is Newton's Method, in which one constructs an iterative algorithm starting with an initial guess y_i^0. The function $\kappa(y_i)$ is then approximated by a line tangent to $\kappa(y_i^0)$. One then constructs a new guess by solving for the root of the line:

$$y_i^{k+1} = y_i^k - \frac{\kappa(y_i^k)}{\kappa'(y_i^k)} \tag{2.13}$$

In this case:

$$\kappa'(y_i) = \sum_{m=1}^{M} \alpha_{im} N_{im}(y_i)$$

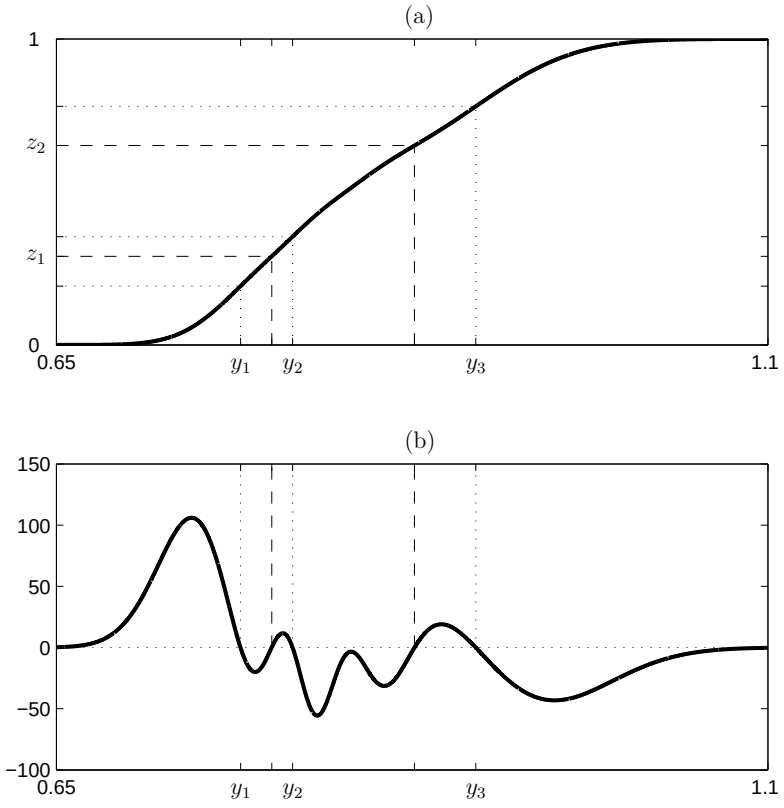

Figure 2.2: Illustration of the expander initialization scheme, showing (a) GM CDF and (b) its Second Derivative. Given some $z \in [0, 1]$ to decode, the initializer is formed by finding the partition that contains z (the dashed lines) and using the inflection point y within that region (the dotted lines).

Furthermore, when κ is a monotonically-increasing function, Newton's method can be shown to converge quadratically (and monotonically) under the following conditions:

$$(y_i^0 - y_i^*)\kappa''(\gamma y_i^0 + (1-\gamma)y_i^*) > 0,\ \forall \gamma \in (0, 1]$$

Which is to say, if $\kappa(y_i)$ is either convex or concave on the interval between the initial guess and the optimal value, and the sign of $(y_i^0 - y_i^*)$ matches that of the second derivative. So, a simple method for determining such an initializer, for any possible value of z_i, is required. This motivates an examination of the second derivative of $\kappa(y_i)$:

$$\kappa''(y_i) = -\sum_{m=1}^{M} \alpha_{im} N_{im}(y_i) \frac{1}{\sigma_{im}^2}(y_i - \mu_{im})$$

Observe that this function is strictly positive for $y_i < \min_m\{\mu_{im}\}$ and strictly negative for $y_i > \max_m\{\mu_{im}\}$, implying that it has an odd number of roots. It is simple to see that the maximum number of roots is $2M - 1$ (well-separated case) and the minimum is 1 (equal-means case). Having fixed the parameters at the end of the training process, it is simple to determine graphically the exact number of such inflection points, L_i, their locations y_{il}^{inf}, and the corresponding values $z_{il} = \sum_m \alpha_{im}\Phi_{im}(y_{il}^{inf})$, where $l \in \{1, \ldots, L_i\}$. Then, given a value of z_i to decode, one can easily determine whether the solution y_i^* lies in a convex or concave region of $\kappa(y_i)$. In the former case, one initializes with the next-largest inflection point, which will make $y_i^0 - y_i^* > 0$ and thus guarantee monotonic, quadratic convergence. In the latter case, one initializes with the next-smallest inflection point, which makes $y_i^0 - y_i^* < 0$ and again guarantees convergence. In other words, the initialization scheme is a simple quantizer for z_i, using the negative-going inflections as codepoints, and the positive-going inflections as cell boundaries. To find the initializer for a given z_i, then, one applies this quantizer to it, and sets y_i^0 to be the inflection point that corresponds to the result. This initialization process is illustrated in Figure 2.2.

Thus, given a stored table of the inflection points, of which there are no more than $(2M-1)$, and another of point density parameters, of which there are $3M$, routines for evaluating scalar Gaussian pdf's and cdf's, and simple arithmetic, one can construct an iterative decoder for the mixture Gaussian expander function. This decoder is guaranteed to converge quadratically, for any value of z_i, which is of great practical value, since one does not wish to perform thousands of iterations in the case of high-rate operation, which will require very tight tolerance in the decoding error. For speech signals, it has been observed that ten iterations of this algorithm are sufficient to result in an average decoding error on the order of 10^{-32}, with a maximum decoding error on the order of 10^{-31}. This is sufficient accuracy for rates up to approximately 30 bits per dimension, which is extremely high. For more moderate rates, five or so iterations should be sufficient.

2.5 Practical Results

2.5.1 A Toy Problem

This toy problem is included to demonstrate the utility of the proposed transform coder design algorithm and compander system. The idea is to consider a case in which the optimal settings for the transform coder are known, both for MSE and input-weighted distortion, and show that the proposed design algorithm can successfully recover the correct parameters in each case. Consider a variable $x \in \mathbb{R}^2$ that is uniformly distributed on a unit square, $[0,1] \times [0,1]$, with the following sensitivity matrix:

$$S(x) = g(x)I$$

Where $g(x)$ is a probability density with all (or almost all) of its mass inside the unit square. In this case, because the sensitivity is proportional to the identity matrix, the problem becomes equivalent to that of an MSE quantizer under a probability density $g(x)$. Suppose that $g(x)$ is a Gaussian Mixture density

with 4 components, as illustrated in Figure 2.3. In this case, each cluster has a covariance of $(24\pi\sqrt{3})^{-1}I$ and equal weighting. This toy problem is inspired by Example V.2 in [5], which demonstrates the suboptimality of the KLT. In this situation, one can specify the optimal transform coder for both MSE and input-weighted squared error. In the MSE case, the optimal transform coder would be as follows: the transform should be the trivial setting $T = I$; the level allocation should be equal for each coefficient, $\beta_i = 1$, $\forall i$; and the point densities should be uniform, $\lambda_i(y) = 1_{[0,1]}(y)$. The high-rate distortion coefficient for these settings is equal to 1/6, for both the MSE and input-weighted distortion measures. For the input-weighted squared error case, the optimal transform should rotate the input by $\pi/4$ radians, as discussed in [5]. The level allocation should again be equal for each coefficient: $\beta_i = 1$. Treating the Gaussian components of $g(x)$ as well-separated, we see that the optimal point densities should be 2-component Gaussian Mixture densities. The high-rate distortion coefficient for these settings is equal to 1/24. Notice that the optimal input-weighted settings result in 1/4 the high-rate distortion of the MSE settings, implying a savings of 1 bit per dimension.

In order to demonstrate the utility of the design algorithm and compander system, we performed a variety of experiments using this example problem. First, 100,000 2-dimensional vectors were generated, distributed uniformly on the unit square. Next, a transform coder was designed from the data to minimize MSE. Although the optimal transform coder for this case calls for uniform point densities, and hence does not require a compander, GM point densities of order 2 were used. As expected, the resulting transform was extremely close to the identity matrix, and the level allocation was extremely close to equal across coefficients. An example point density obtained through this design is seen in Figure 2.4a, along with a histogram of the corresponding transform coefficient, and an estimate of the optimal point density based on the histogram. Note that point densities much closer to uniform can be achieved by using a larger value of M; however, we utilize $M = 2$ here for easy comparison to later parts of this example.

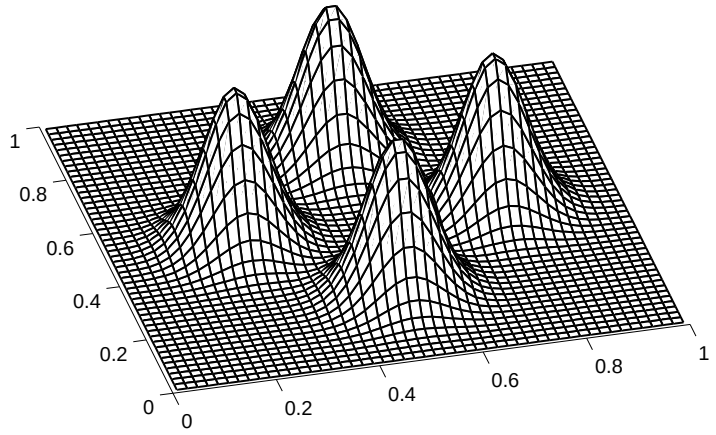

Figure 2.3: Illustration of Distortion Sensitivity

Next, the sensitivity matrix for each point in the database was computed and the design process was repeated under input-weighted squared error. In this case, the resulting parameters were very close to the expected values: the transform was approximately a $\pi/4$ rotation, the level allocation was close to equal, and the point densities were as shown in Figure 2.4b. In this case, each histogram bin has been weighted by the sum of the sensitivities for elements in that bin, and then the entire histogram is normalized. Notice that the point density designed by our algorithm is very close to the estimated optimal point density, as expected. This outcome shows that the design algorithm is able to incorporate the data statistics and the sensitivity matrix to arrive at the proper transform coder parameters.

Finally, both of the systems designed above were operated on the data, under the input-weighted distortion measure. The results are shown in Figure 2.5. Notice that for both the MSE-assumption and optimized systems, the actual performance and high-rate estimates converge around 3 bits per dimension. The theoretical curves are slightly inaccurate in this case because, in the MSE case,

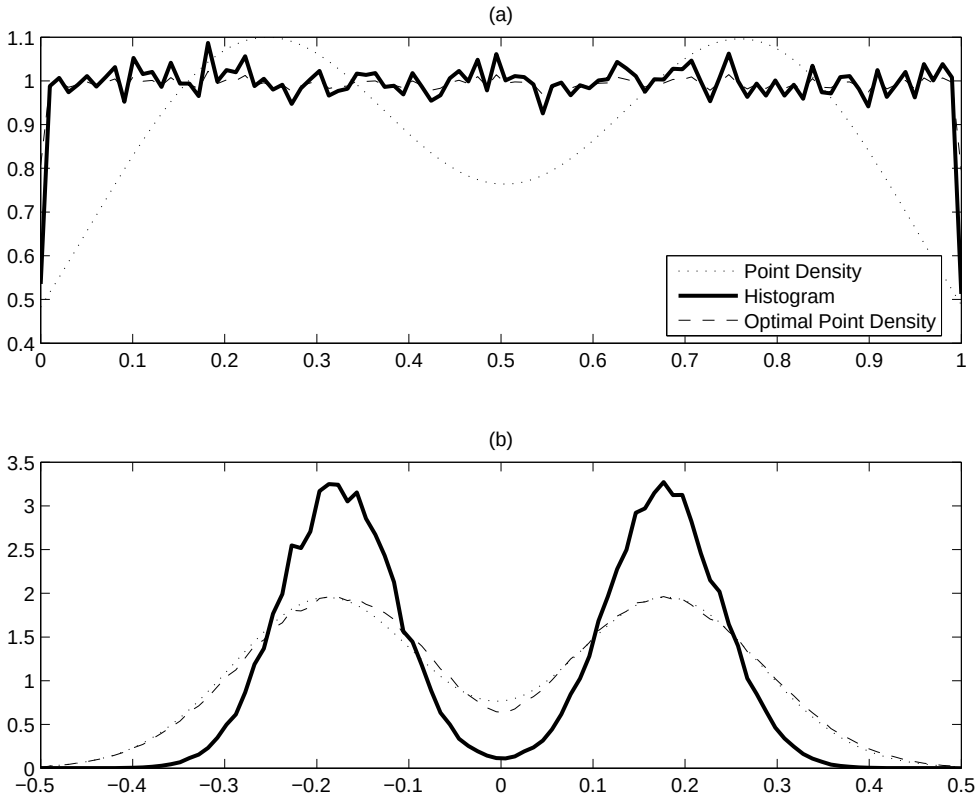

Figure 2.4: Point Densities and Weighted Histograms for a) MSE and b) Input-Weighted Distortion Measures. The first transform coefficient is pictured in both cases.

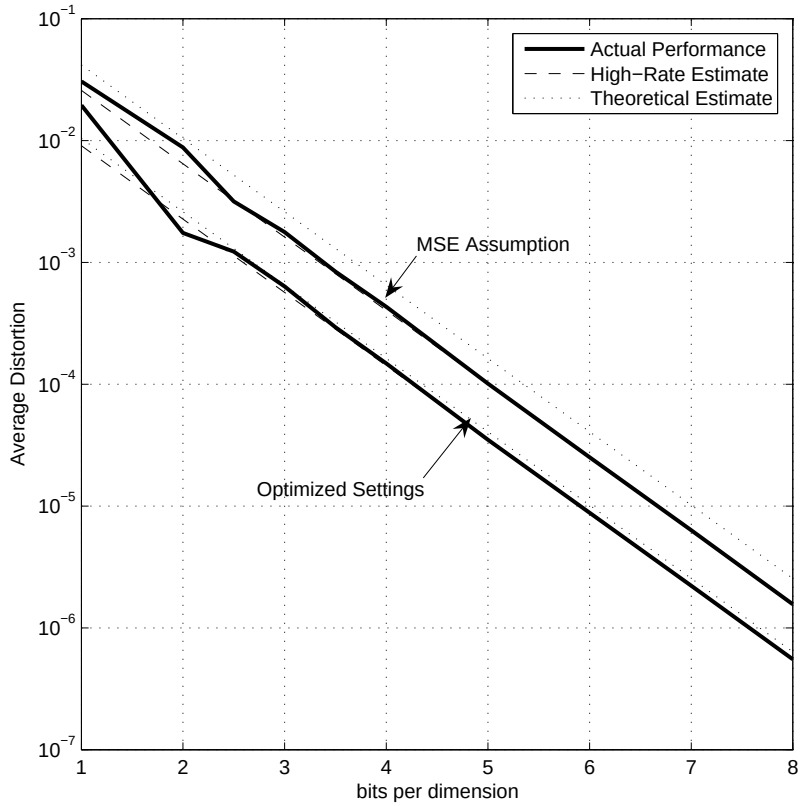

Figure 2.5: Theoretical, Estimated and Actual Performance on the Toy Problem for MSE and Input-Weighted Designs.

they assume uniform point densities and, in the input-weighted case, they assume perfect separation of the clusters and infinite support. The result of this is that the optimized system saves slightly less than the predicted 1 bit per dimension over the MSE-assumption system. Nevertheless, it is clear that the high-rate estimate is a good predictor of true high-rate performance, and that the algorithm is successful at exploiting both statistics and distortion sensitivity in designing the system.

2.5.2 Speech Spectrum Quantization

Next, the problem of wideband speech spectrum coding, under the Log Spectral Distortion measure, is considered. To facilitate this experiment, a training set of 300,000 wideband speech LSF vectors of order 16 was gathered, and the LSD-sensitivity matrix was evaluated for each vector using the method described in [20]. It should be noted that, for the high-rate analysis, LSD is measured in dB^2 in order to correspond to input-weighted squared error, and so high-rate approximations use this metric (Figure 2.6, specifically). However, when calculating operating point and outlier statistics (as in Tables 2.1-3), the more conventional approach of measuring in dB is used. For testing purposes, an independent database of 65,000 LSF vectors was employed. First, a transform coder was designed using single Gaussian scalar quantizers. To initialize the parameters, the data was assumed to be Gaussian and so the sample mean and covariance were used to set the parameters to be optimal under the MSE assumption. This initial setting is intended to represent a "naive" design, which ignores the details of the statistics and distortion measure. The data-driven transform coder design algorithm described in Section 2.2 was then applied to optimize over the actual statistics and distortion measure. After trying a variety of other initializers ($T = I$, and many random transforms), it was determined that this result was indeed the best possible.

The performance before and after optimization, both actual and predicted, is seen in Figure 2.6 and Table 2.1. Notice that, at high rates, optimization resulted in a large savings of around 8 bits per dimension (128 bits per frame in this case). However, this is only applicable at very high rates, above 20 bits per dimension. At lower rates, the non-optimized coder does better than the high-rate predictions, diminishing the advantage of the optimized system. Notice that the performance of the optimized system matches the high-rate predictions over a wide range of rates. It is generally expected that an optimal system will fulfill the high-rate assumptions at a lower rate than a suboptimal system, since it is better matched to the source. The magnitude and direction of the discrepancy

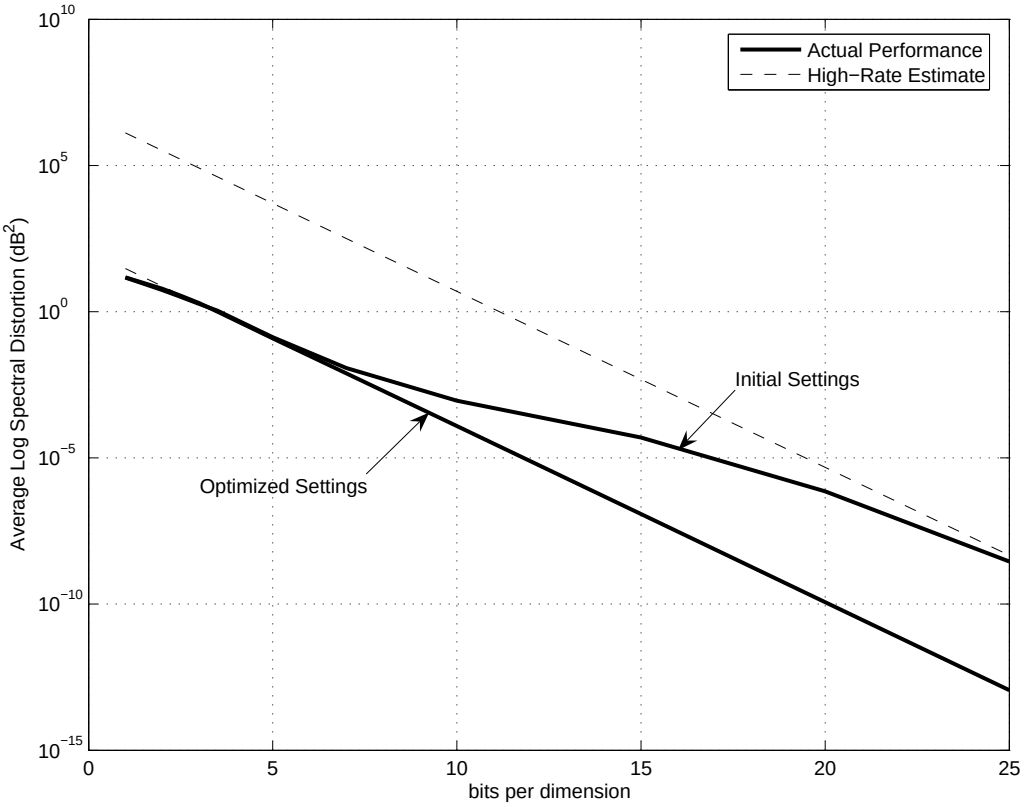

Figure 2.6: Wideband Speech Spectrum Performance for Gaussian Transform Coder

Table 2.1: Spectral Distortion performance of Gaussian Transform coder ($M = 1$)

	LSD-Optimized			MSE-Optimized			Non-Optimized		
bits/frame	Avg. LSD	Outliers (in %)		Avg. LSD	Outliers (in %)		Avg. LSD	Outliers (in %)	
	(in dB)	2-4dB	> 4dB	(in dB)	2-4dB	> 4dB	(in dB)	2-4dB	> 4dB
51	1.218	2.708	0.008	1.301	4.835	0.015	1.204	2.677	0.008
52	1.175	2.117	0.003	1.153	2.312	0.005	1.165	2.355	0.009
53	1.135	1.687	0.006	1.113	1.870	0.006	1.059	1.351	0.003
54	1.040	1.072	0.003	1.094	1.589	0.006	1.020	1.105	0.005
55	0.984	0.759	0.000	1.035	1.374	0.003	0.981	0.843	0.005
56	0.964	0.656	0.002	0.949	0.764	0.000	0.965	0.797	0.003

between actual and predicted performance are difficult to anticipate, and depend heavily on the source and the nature of the suboptimality (compare with results in Section 2.5.1, for example). At very low rates, the non-optimized system performs slightly better. This makes sense in that, for a rate of 0 bits, the coder should be centered at the data mean, which it is in the non-optimized case. Since the optimization has moved the center of the transform code to improve high-rate performance (by adjusting for skew in the marginal distributions of the transform coefficients), it necessarily suffers a small loss at very low rates. As it turns out, the desired operating point of 1dB LSD lies in the intermediate range, where neither system has a clear advantage. Notice that, while the average performance of the systems are very similar, the optimized system produces fewer outliers. It is also noteworthy that, while the transform departs from the exact KLT during the optimization process, the result is still a "KLT-like" transform, in that most of the energy is compacted into a small number of coefficients, and the covariance of the transformed data is close to diagonal.

Next, since the transform coefficients are clearly non-Gaussian (see Figure 2.7), the number of Gaussians used in each scalar quantizer was increased. At each stage, M was doubled, using the result from the previous stage to initialize via a simple splitting scheme. As above, a variety of other initializers were tried, but the former method was found to work best. No significant gains in the high rate distortion were observed for $M > 2$, and so a mixture of 2 Gaussians are

Table 2.2: Spectral Distortion Performance of Optimized GMM Transform Coder ($M = 2$)

bits/frame	LSD-Optimized			MSE-Optimized		
	Avg. LSD (in dB)	Outliers (in %)		Avg. LSD (in dB)	Outliers (in %)	
		2-4 dB	> 4 dB		2-4 dB	> 4 dB
50	1.232	2.520	0.008	1.218	2.973	0.006
51	1.189	2.029	0.005	1.175	2.418	0.005
52	1.148	1.656	0.008	1.096	1.571	0.008
53	1.109	1.342	0.005	1.094	1.621	0.006
54	1.018	0.813	0.003	1.059	1.360	0.005
55	0.980	0.650	0.005	0.975	0.860	0.002
56	0.942	0.538	0.005	0.937	0.727	0.000

used for each companding scalar quantizer. This resulted in a reduction in high-rate distortion of 10% over the $M = 1$ case. The performance of this system, including outlier statistics, is listed in Table 2.2 (these curves are not pictured in Figure 2.6 because, at this scale, it is difficult to distinguish from the $M = 1$ curves). Comparing to the other tables, there is an improvement in average distortion, corresponding to a savings of roughly one-half a bit per frame compared to the single-Gaussian case. Note that the outlier statistics have improved by an even greater margin. While high-rate optimization of the companding transform coder may produce only modest improvements in average distortion at rates of interest, it can produce significant reductions in outliers. The point densities of two of the optimized component scalar quantizers, along with the estimated pdf's of the corresponding transform coefficients, are seen in Figure 2.5. Notice that the GM compander is able to account for multimodality and skew, and so closely approximates the estimated optimal point densities.

Next, the importance of including the LSD sensitivity in the training

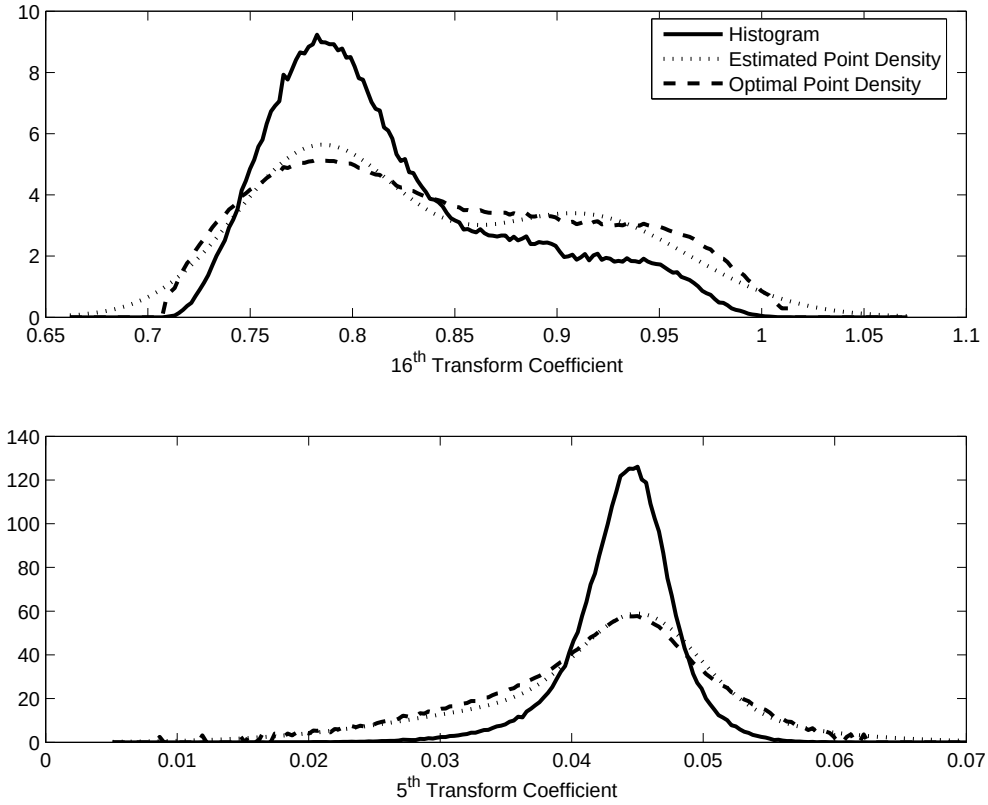

Figure 2.7: Histograms and Point Densities for Two Example Transform Coefficients

Table 2.3: Spectral Distortion Performance of Unstructured Transform Coder

bits/frame	Optimized Transform			KLT		
	Avg. LSD (in dB)	Outliers (in %)		Avg. LSD (in dB)	Outliers (in %)	
		2-4 dB	> 4 dB		2-4 dB	> 4 dB
50	1.081	2.167	0.025	1.122	2.637	0.032
51	1.046	1.987	0.019	1.076	2.240	0.026
52	1.009	1.715	0.017	1.048	2.104	0.035
53	0.977	1.536	0.019	1.009	1.768	0.025
54	0.918	0.975	0.015	0.945	1.141	0.011
55	0.887	0.890	0.017	0.925	1.068	0.019
56	0.844	0.681	0.008	0.878	0.898	0.014

process was investigated. As seen in the toy problem of section 2.5.1, it is possible for the distortion measure to play a strong role in the training process. To check this, the previous training process was repeated assuming MSE (i.e., setting the sensitivity as $S(x) = I$). The performance was still measured in LSD. The resulting estimated high-rate distortions were essentially equivalent to those in the LSD-optimized case, and so are not repeated here. The results for $M = 1$ and $M = 2$, at rates of interest, are included in Tables 2.1 and 2.2, respectively. Notice that the MSE-optimized systems display essentially the same average distortion as the LSD-optimized systems, but produce significantly more outliers. This implies that while the transform coder is not capable of exploiting sensitivity to LSD in the average sense, including LSD in training can nevertheless result in significantly improved outlier performance.

Finally, it is noted that the desired operating point is not terribly high: a little more than 3 bits per dimension. Also, all of the transform coders designed in this subsection exhibit significant energy compaction, with the ratio of the largest β to the smallest being on the order of ten. Together, these facts suggest that bet-

ter performance could be achieved by forgoing the use of companders and utilizing unstructured scalar quantizers instead. To that end, the hybrid training approach of Section 2.3 was applied to design such an unstructured scalar transform coder. That is, the high-rate optimized transform and allocation (for $M = 2$ and LSD) were utilized, and a bank of scalar quantizers were then trained using the weighted Lloyd algorithm of Section 2.3.1. The performance of this system is shown in Table 2.3. Note that the unstructured system achieves substantially better average distortion (around 3 bits per frame), but does so at the cost of degraded outlier performance. From this we infer that the compander is producing a more "uniform" distribution of codepoints as compared to the unstructured scalar quantizers, resulting in fewer outliers. As a final test of the hybrid training strategy, another transform coder was designed using the KLT. That is, the transform was left fixed throughout the training phase, and the high-rate scheme was used only to estimate the optimal allocation and initialize the scalar quantizers. The scalar quantizers were again designed using the weighted Lloyd algorithm. The performance of this system is illustrated in Table 2.3. Notice that its performance lags behind the system with high-rate optimized transform by about 1 bit per frame, in both the average and outlier senses, demonstrating the utility of high-rate selection of the transform, even at moderate rates.

2.6 Discussion

This chapter has presented a flexible companding scalar quantizer based on Gaussian Mixtures, and a data-driven method for training transform coders based on these companders that minimizes high-rate distortion. The design algorithm is able to incorporate a wide variety of distortion measures by utilizing the input-weighted squared error formulation, which is characterized by a variable sensitivity matrix. Gaussian Mixture companders provide a flexible, extensible approach to quantizing sources with arbitrary distributions. However, it should be

noted that some sources, such as the uniform source in the toy problem, are not well-suited to this type of compander. Additionally, companders suffer from well-known losses at low-to-moderate rates, and so modifications to the design scheme were provided to incorporate unstructured scalar quantizers, which are suitable for operation at moderate rates. The ability of the system and design algorithm to account for both data statistics and the distortion measure was demonstrated in a toy problem. The transform coder has remained popular due to its low complexity; however, the very structure that provides this low complexity means that the transform coder necessarily suffers from performance limitations. This work seeks to reduce these, both through introducing more flexible scalar companders and through the use of a minimum-distortion design algorithm.

This system and design algorithm were then applied to the problem of wideband speech spectrum quantization, using Line Spectral Frequencies under the Log Spectral Distortion measure. While very large high-rate gains (as much as 8 bits per dimension) are possible, these results are only applicable at very high rates. At rates of interest, the compander-based system showed only modest gains, on the order of 1 bit per frame. Interestingly, application of the design algorithm resulted in significant improvements in outlier performance. It was also discovered that including the LSD measure in training made no difference in the average-distortion performance, but did result in improved outlier performance. Next, systems employing unstructured scalar quantizers and a hybrid design algorithm were employed. These systems exhibited significantly better average distortion performance than their compander-based counterparts (around 3 bits per frame), but at the cost of degraded outlier performance. It was also demonstrated that high-rate design of the transform is desirable, even at moderate rates for which companders perform poorly. In the end, transform coding systems were shown to achieve transparent quality at rates comparable to MSVQ. While other schemes have been seen to provide superior rate-distortion performance for wideband speech LSF quantization, the transform coder may still be attractive for certain applica-

tions by virtue of its extremely small storage and coding complexities.

2.7 Inertial Moment Integral for Input-Weighted Squared Error on a Hyperrectangle

Consider a vector $x \in \mathbb{R}^d$ and a matrix $S \in \mathbb{R}^{d \times d}$. Define the origin-centered rectangle $\mathcal{R}^d = \prod_{i=1}^{d} [\frac{-1}{2} c_i, \frac{1}{2} c_i]$. It is desired to evaluate the following integral:

$$\int_{\mathcal{R}^d} x^\mathsf{T} S x \, \mathrm{d}x$$

Since the region of integration is defined by a product, this integral can always be evaluated by iterated integration. One can easily perform the integral for low dimensions. Denoting the ij-th element of S as s_{ij}, the results of the integral for d from 1 to 3 are:

$$\begin{aligned}
&\underline{\mathrm{d} = 1:} \quad && \frac{s_{11}}{12} c_1^3 \\
&\underline{\mathrm{d} = 2:} \quad && \frac{s_{11}}{12} c_1^3 c_2 + \frac{s_{22}}{12} c_1 c_2^3 \\
&\underline{\mathrm{d} = 3:} \quad && \frac{s_{11}}{12} c_1^3 c_2 c_3 + \frac{s_{22}}{12} c_1 c_2^3 c_3 + \frac{s_{33}}{12} c_1 c_2 c_3^3
\end{aligned}$$

Examining the results, one can hypothesize the solution for arbitrary d:

$$\int_{\mathcal{R}^d} x^\mathsf{T} S x \, \mathrm{d}x = \frac{1}{12} \left(\prod_{i=1}^{d} c_i \right) \sum_{i=1}^{d} s_{ii} c_i^2 \tag{2.14}$$

In fact, one can show that this must be the solution by induction. Since it has already been shown that the solution is of the form of Eq. (2.14) for $d = 1, 2, 3$, one need only show the implication from $d-1$ to d. Partition x and S as follows:

$$x = \begin{bmatrix} \hat{x}_{d-1} \\ x_d \end{bmatrix}, \; S = \begin{bmatrix} \widehat{S}_{d-1} & \hat{s}_1 \\ \hat{s}_2^\mathsf{T} & s_{dd} \end{bmatrix}$$

Then, the integral can be written as:

$$\int_{\mathcal{R}^{d-1}} \left[\int_{\frac{-1}{2}c_d}^{\frac{1}{2}c_d} \hat{x}_{d-1}^{\mathsf{T}} \widehat{S}_{d-1} \hat{x}_{d-1} + x_d \left(\hat{x}_{d-1}^{\mathsf{T}} \hat{s}_1 + \hat{s}_2^{\mathsf{T}} \hat{x}_{d-1} \right) + s_{dd} x_d^2 \right] \partial x_d \, \partial \hat{x}_{d-1} \qquad (2.15)$$

Evaluating the inner integral results in:

$$\int_{\mathcal{R}^{d-1}} \left[c_d \hat{x}_{d-1}^{\mathsf{T}} \widehat{S}_{d-1} \hat{x}_{d-1} + \frac{s_{dd}}{12} c_d^3 \right] \mathrm{d}\hat{x}_{d-1} \qquad (2.16)$$

Using the assumption that the $d-1$ integral is in the desired form, and the fact that $\int_{\mathcal{R}^{d-1}} \mathrm{d}\hat{x}_{d-1} = \prod_{i=1}^{d-1} c_i$:

$$\begin{aligned} \int_{\mathcal{R}^d} x^{\mathsf{T}} S x \, \mathrm{d}x &= c_d \frac{1}{12} \left(\prod_{i=1}^{d-1} c_i \right) \sum_{i=1}^{d-1} s_{ii} c_i^2 + \frac{1}{12} s_{dd} c_d^3 \left(\prod_{i=1}^{d-1} c_i \right) \\ &= \frac{1}{12} \left(\prod_{i=1}^{d} c_i \right) \sum_{i=1}^{d} s_{ii} c_i^2 \end{aligned} \qquad (2.17)$$

Which is the desired result. Notice that no assumptions were made on the matrix S, and that only its diagonal elements enter into the result. Also, note that no terms from S appear in the product term, which implies that the usual cancelations will still occur when this result is used in high-rate analysis.

The text of this chapter is in part a reprint of the material which was coauthored with Bhaskar D. Rao and appeared in the March 2007 issue of *IEEE Transactions on Audio, Speech and Language Processing* under the title *"A High-Rate Optimal Transform Coder with Gaussian Mixture Companders"*. The dissertation author was the primary researcher and author, and the co-author contributed to or supervised the research which forms the basis for this chapter.

3 High-Rate Optimized Recursive Vector Quantizers using Hidden Markov Models

This chapter develops a variety of fixed-rate recursive vector quantization systems using Hidden Markov Models offering a wide range of performance to complexity tradeoffs. These systems build on the idea of Gaussian Mixture Vector Quantizers (GMVQ), first presented in [12], which exhibit low complexity, high quality and scalability (see Figure 3.1). GMVQ systems operate M Gaussian quantizers in parallel (typically with low-complexity structures) and then choose amongst their outputs with a vector quantizer. In this context, a Gaussian quantizer can be any system that produces a Gaussian point density. Since the number of component Gaussian quantizers, M, is typically much smaller than the effective codebook size, 2^r, a substantial savings in complexity can be realized. Such an approach is attractive in applications where large codebooks are required, particularly in high-dimensional problems such as wideband speech spectrum coding and image coding. A GMVQ is specified by three sets of parameters: a set of mixture weights, $\widehat{\alpha}_m$, that determine what percentage of the available codepoints are assigned to the m-th quantizer; a set of mean vectors $\widehat{\mu}_m$ that specify the locations of each Gaussian quantizer; and a set of covariances $\widehat{\Sigma}_m$ that specify their dispersions. This compact parametric form allows the system to be updated on the fly. Thus, flexible recursive quantizers can be implemented by altering the pa-

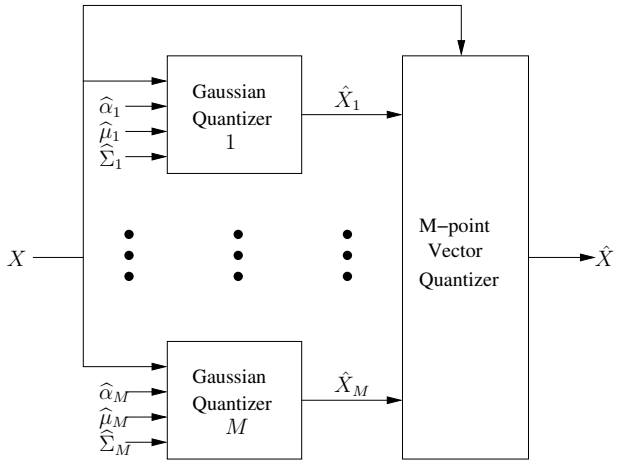

Figure 3.1: The Gaussian Mixture Vector Quantizer system. Each component quantizer produces a Gaussian point density $N(x|\mu_m, \Sigma_m)$. The parameters α_m specify the proportion of codepoints allocated to each component quantizer, resulting in an overall point density of $\sum_{m=1}^{M} \alpha_m N(x|\mu_m, \Sigma_m)$.

rameters at each time step based on previous quantized outputs as in Figure 3.2. This chapter examines a variety of techniques, based on Hidden Markov Models, for accomplishing this. The proposed systems are capable of exploiting both linear and nonlinear dependencies between vectors. The contributions of this chapter are twofold: first, the high-rate theory for GMVQ systems is developed, leading to novel system training approaches that minimize distortion. Second, a variety of practical recursive extensions are presented, and their implementation and complexity are explored. Performance of the proposed systems is demonstrated for the problem of wideband speech Line Spectral Frequency (LSF) quantization under the Log Spectral Distortion (LSD) measure.

This chapter first revisits the problem of training the system parameters based on example data. The classic approach to this problem is known as *model-based training* [12], wherein a statistical model of the source is first constructed using Maximum Likelihood techniques, and then the quantizer parameters are set based on the model using a closed-form, heuristic approach. Specifically, for an order-M GMVQ system, an order-M Gaussian Mixture Model (GMM) is used as

the statistical model. What is generally desired is to set the system parameters so as to minimize the high-rate distortion incurred by the quantizer. A related approach called HRO was employed in [13], where the GMM is estimated so as to minimize the high rate distortion incurred by a codebook that is optimal for the model density (that work did not employ GMVQ systems). This chapter discusses the high-rate analysis of GMVQ systems, using random coding. It is shown that the standard model-based training approach results in minimal high-rate distortion for the special case of well-separated GMM sources with MSE distortion. Next, an alternate design technique called *Weighted Maximum Likelihood* (WML) is proposed that works by equating the model and quantizer parameters; it is shown that this technique results in minimal high-rate distortion in the case of large dimensions. WML is suitable for sources with arbitrary statistics and distortion measures, and can be extended to handle mismatched distortion measures. The relative merits of these design schemes are compared for a variety of simple examples, and for the practical problem of wideband speech LSF quantization.

Next, this chapter considers the practical problem of recursive coding. The GMVQ system can be made recursive by changing the parameters at each time step based on previous quantized outputs. The goal, then, is to match the codebook at any given time to the conditional probability density of the source, given the previous data. Thus, any recursive coding scheme corresponds to a dynamic model of the source, and more flexible models will require more complex recursive updates. The ideal recursive structure would provide a very flexible mechanism for exploiting dependence on previous data, while incurring only a modest increase in complexity. A recursive GMVQ system based on a joint-GMM model was proposed in [31]. This work seeks to extend these results to more flexible recursive structures, while incurring a minimal increase in complexity. This chapter first proposes using Hidden Markov Models, which provide a simple mechanism for recursion by varying the level allocations, $\widehat{\alpha}_m$, based on previous data. The remaining coder parameters (i.e., the means $\widehat{\mu}_m$ and covariances $\widehat{\Sigma}_m$) remain fixed.

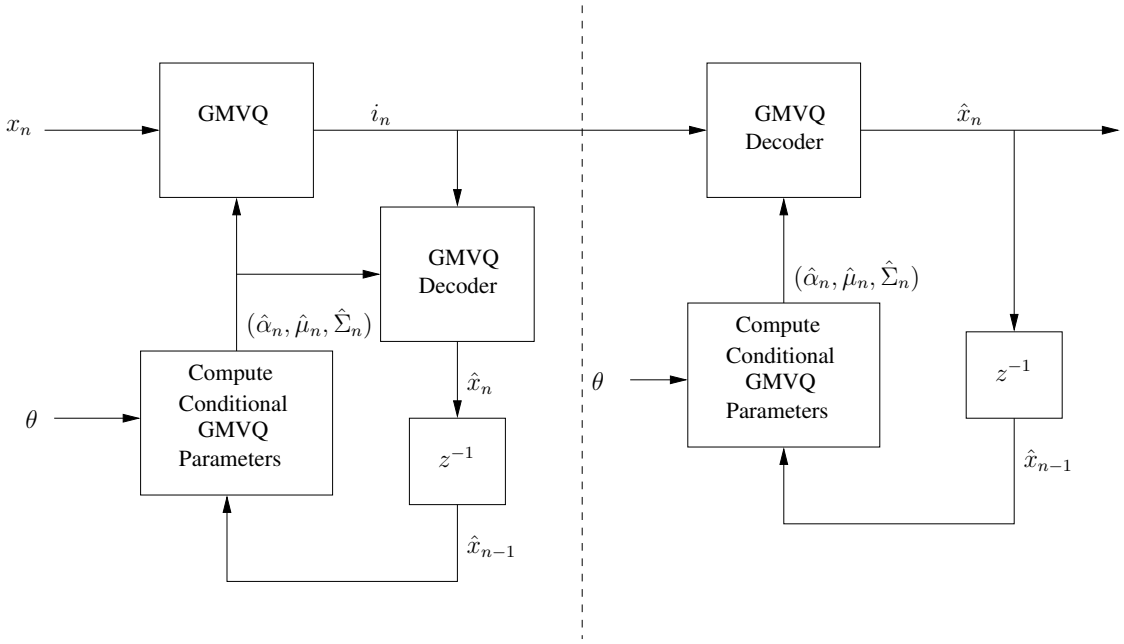

Figure 3.2: Proposed recursive coding architecture based on GMVQ. The systems considered here store only one previous sample, and the covariances $\widehat{\Sigma}_m$ remain constant.

Inspired by the joint-GMM recursive model in [31], a generalized HMM approach is proposed which is also capable of updating the codebook means $\widehat{\mu}_m$. All of the systems considered in this chapter leave the covariances $\widehat{\Sigma}_m$ fixed, in order to keep the complexity low. Note that essentially all structured Gaussian quantizers utilize the eigendecomposition of $\widehat{\Sigma}_m$, both to obtain the Karhunen-Loeve Transform and in bit allocation; thus, recursive schemes that leave $\widehat{\Sigma}_m$ fixed can avoid the complexity associated with computing the eigendecomposition. This system provides for inclusion of both strong short-time dependencies and weaker long-time dependencies. The additional encoding complexity of these schemes is minimal, and only the most recent quantized vector needs to be stored. HMM-based recursive VQ has also been examined in [32] and [33]. This work is differentiated from [32] in that it can handle continuous-valued vector sources, and is for fixed-rate operation. It is differentiated from [33] in that new codebooks are created on-the-fly, rather than relying on a fixed set of pre-designed codebooks.

A wide variety of performance/complexity tradeoffs can be implemented by the GMVQ system through selection of the component Gaussian quantizers. For example, one could use full-search VQs, optimized by the Lloyd algorithm for synthetic Gaussian sources; this would presumably offer very good performance, at the cost of large complexity. The most popular structure for implementing Gaussian coders in this context is the scalar transform coder, which represents the other end of the spectrum: very low complexity at the cost of performance. While the high-rate analysis employed in this paper is based on random coders, the classic transform coder-based GMVQ is not really a random coder. However, the gap between the two can be bridged by a related class of semi-random Gaussian quantizers that has recently been introduced in [19] and [34], called CURTZ systems. These systems feature a parameter L that allows them to scale between a scalar transform coder on one extreme, to what is effectively a Gaussian random coder on the other. In light of this, GMVQ systems will be considered as random coders for purposes of analysis and training. Performance arbitrarily close to the

high rate estimates can then be achieved by appropriate selection of L, at the cost of increasing complexity.

This chapter is organized as follows: Section 3.1 discusses the theoretical issues pertaining to the analysis and training of GMVQ systems, using random coding and Maximum Likelihood techniques. Section 3.2 covers relevant background on Hidden Markov Models, with an eye towards their application in recursive coding. Section 3.3 discusses the implementation of the proposed systems. Section 3.4 examines the performance of the proposed recursive quantization systems on the problem of wideband speech LSF quantization. Section 3.5 contains a discussion of the results.

3.1 System Training - High Rate Theory and Maximum Likelihood

This section discusses the problem of setting GMVQ parameters based on example data. The usual practice, called *model-based design* and described in [12], is to first estimate a statistical model of the source, using Maximum Likelihood. This model is typically a GMM of order M, the same order as the GMVQ system being designed. Then, a closed-form procedure is applied to give the quantization system parameters in terms of the model parameters. However, it would be preferable to design the system so as to minimize the distortion directly. In light of this, the high-rate theory for GMVQ systems is developed, and its relationship to Maximum Likelihood is discussed. Specifically, it is shown that the model-based design approach produces minimal high-rate distortion in the case of well-separated GMM sources and MSE distortion. Next, it is shown that Weighted Maximum Likelihood (WML) can be used to find the quantization system parameters directly from the data in high dimensional cases, for arbitrary sources and input-weighted squared error measures. That is, where model-based design splits the problem into a statistical modeling step and a model-based design step, WML

fuses the two steps to infer the optimal system parameters directly from example data. The relationship between model-based design and WML design is explored for a variety of examples. Finally, it is shown that the two design techniques produce equivalent high-rate distortion for wideband speech LSFs, and so either may be used.

3.1.1 High-Rate Analysis of Gaussian Mixture Vector Quantizers

To consider the high-rate analysis of a structured quantizer, one must examine both the point density and inertial profile, as discussed in Chapter 1. Derivation of the point density for a GMVQ is straightforward. The total codebook in a GMVQ system is the union of the component Gaussian codebooks. Thus, as discussed in [37], a GMVQ system, regardless of its exact component structures, produces a point density that is also a Gaussian Mixture:

$$\lambda_\theta(x) = \sum_{m=1}^{M} \widehat{\alpha}_m N(x|\widehat{\mu}_m, \widehat{\Sigma}_m) \tag{3.1}$$

Here, θ denotes the set of system parameters $\left\{\widehat{\alpha}_m, \widehat{\mu}_m, \widehat{\Sigma}_m\right\}_{m=1}^{M}$. In general, it is difficult to analyze the inertial profile of a GMVQ system, for the reason that the process of selecting amongst the component quantizers has a very complicated effect on the cell shapes of the resultant quantizer. That is, even when the component Gaussian systems can be analyzed in closed form (such as in the case of scalar transform coders), no expression is available for the overall inertial profile. While the encoding in a GMVQ system is optimal provided that the encodings of the component systems are optimal, the aggregate codebook takes on a randomized character in regions where the component codebooks overlap. Thus, the codepoints are typically not the centroids of their cells, and any structure in the component codebooks is disturbed. The exception is the case of random coders: if random Gaussian coders are utilized, it is intuitively clear that the overall system will also be a random coder. These effects are illustrated in Figure 3.3. Shown is a

two-dimensional, two-component GMVQ using a variety of CURTZ systems as the component quantizers. On one extreme is the scalar transform coder, and on the other is a CURTZ system with $L = 2^d = 4$, which gives performance equivalent to random coding. Note that the Voronoi regions of the overall GMVQ system are very complex in regions where the coders overlap, taking on a randomized character. However, the disparity between the overall and component cell shapes disappears as the component codebooks become more randomized.

Thus, the inertial profile of a GMVQ system tends to depart from the profiles of the underlying coders and become more like that of a random coder. The severity of this effect depends on the degree of overlap between the component codebooks. In the case of inefficient component systems, such as scalar transform coders, this effect tends to result in an improved inertial profile. Conversely, this implies that the performance of near-optimal Gaussian quantizers will be somewhat degraded by the mixing process. Thus, random coders can be thought of as a fixed-point of the mixing process, and so this analysis proceeds under the assumption of random coders. The classic high-rate results on random coders can be found in [35] or [36]. Note that these works consider only MSE distortion, and do not utilize the idea of an inertial profile. It is straightforward to reinterpret their results in terms of an *expected inertial profile*, where the expectation is over the codebook realization, and we further conjecture an extension to input-weighted squared error along the lines of [20]. That is, the expected inertial profile of a random coder is given as $\kappa_d^{2/d}\Gamma(1+2/d)|S(x)|^{1/d}$. Note that in this context, the expected distortion is an expectation over both the source and codebook realization, although the variance due to codebook randomness diminishes as the rate grows large. Also note that the random coder approaches the performance of an optimal quantizer as d grows large.

The high-rate design problem for GMVQ systems employing random Gaussian coders is then:

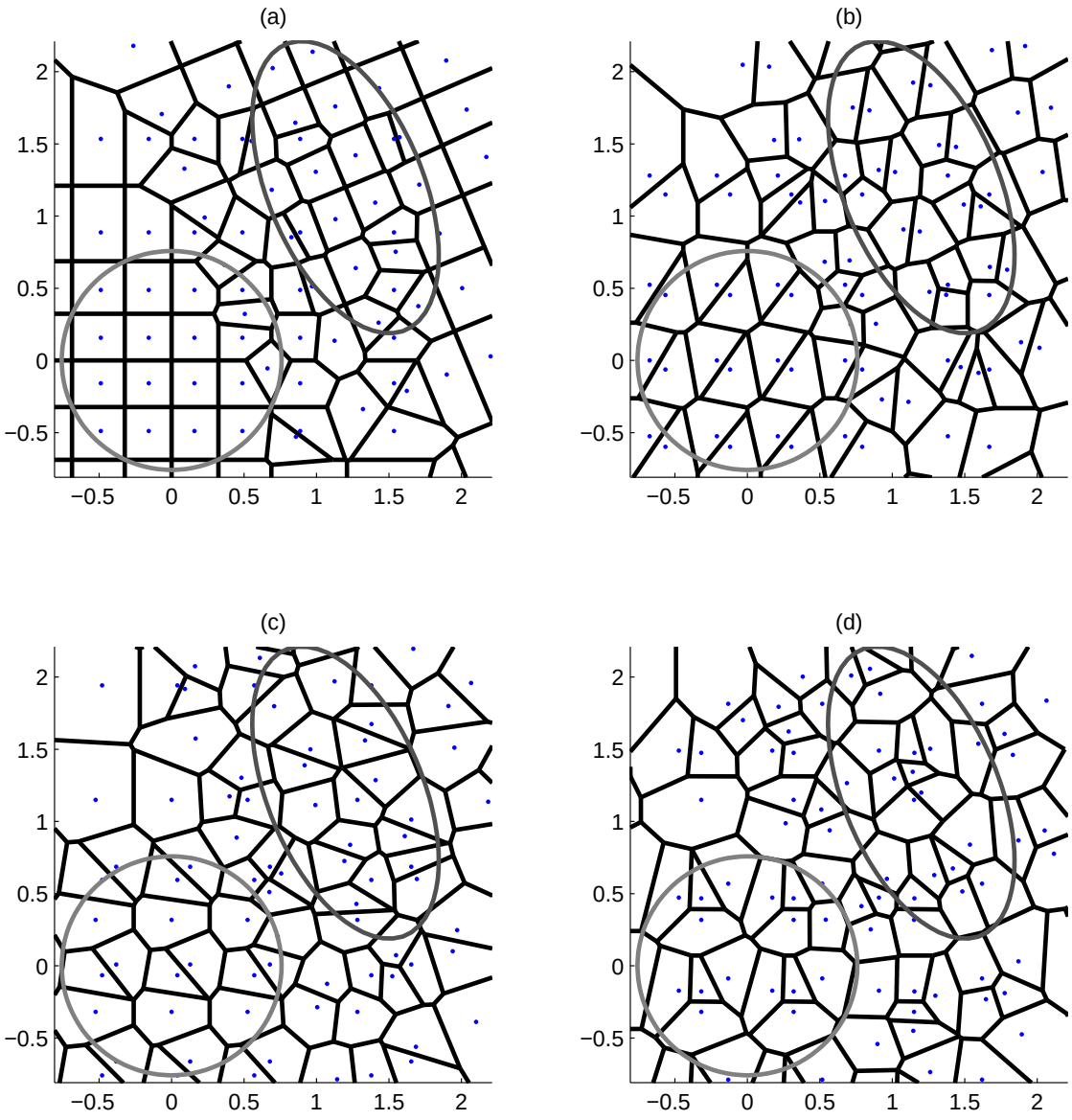

Figure 3.3: Illustration of GMVQ codebooks for a variety of component Gaussian systems. Shown are (a) scalar Gaussian transform coders, (b) CURTZ coders with $L = 2$, (c) CURTZ with $L = 3$ and (d) CURTZ with $L = 4$, which gives performance equivalent to random coding. Ellipses correspond to equi-density contours of the two component quantizers, each of which contain equal numbers of codepoints.

$$\min_{\theta} \mathrm{E}_{\mathrm{x}} \left(|S(x)|^{1/d} \lambda_{\theta}^{-2/d}(x) \right) \tag{3.2}$$

Notice that this objective function is independent of r, which appeared only in the exponential term in Eq. (1.1), implying that a single, rate-independent setting of θ will suffice for any high rate. As discussed in [20], the optimal solution to this problem is given by:

$$\lambda_{\mathrm{opt}}(x) \propto |S(x)|^{1/d} f_{\mathrm{x}}^{\frac{d}{d+2}}(x). \tag{3.3}$$

The extension of high rate theory to the case of recursive systems was covered in [38], where the *conditional point density* $\lambda(x|Y)$ was introduced. Here, Y represents all previously observed vectors. The conditional point density is, for any fixed Y, a density over x, with the same interpretation in terms of the local cell size. What the conditional point density provides, then, is a description of the dependence of the quantizer on past data; in the case of GMVQ systems, this amounts to a mapping from Y to a set of GM parameters. The above results on high rate theory can be generalized to the recursive case by substituting $\lambda(x|Y)$ for $\lambda(x)$ and $f_{\mathrm{x}}(x|Y)$ for $f_{\mathrm{x}}(x)$. In the recursive case, the expectation in Eq. (1.1) is taken to be an expectation over x and Y.

3.1.2 Relationship Between High-Rate Theory and Maximum Likelihood

Reflecting the dependence of the optimal point density on the probability density in Eq. (3.3), quantizer design is often built around probability estimation. It is often desirable to model the source pdf as a GMM, since it is a flexible model and there exist well-studied techniques for estimating its parameters. One notable approach is the HRO algorithm developed in [13], which estimates the GMM parameters so as to minimize the high-rate distortion of a quantizer with the optimal point density implied by the model. One difficulty in using GMMs as probability models in the context of GMVQ design is that the optimal point

density corresponding to the model is not itself a GMM, due to the exponent in Eq. (3.3), and so the GMVQ system cannot implement it. Nevertheless, it has been standard practice to design GMVQs by employing a GMM (with the same number of Gaussians as the quantizer) as a statistical model of the source. This practice can be theoretically justified in two different cases: well-separated mixtures and high dimensions, each resulting in a different training scheme. The following subsections discuss these cases and the relationship between the training schemes that result.

Well-Separated Mixtures and Model-Based Training

Consider the case that the source density is a well-separated GMM:

$$\begin{aligned} f_{\mathrm{x}}(x) &= \sum_{m=1}^{M} \alpha_m N(x|\mu_m, \Sigma_m) \\ ||\mu_i - \mu_j||_{\Sigma_i^{-1}} &\gg 1 \ , \forall i, j \in \{1, \ldots, M\} \end{aligned}$$

The "well-separated" condition means that the means μ_m are far apart relative to the covariances Σ_m, so that only a single Gaussian is "active" in any particular region. Specifically, this property means that the optimal point density is, for the MSE case ($S(x) = I$), again a GMM:

$$\begin{aligned} \lambda_{\mathrm{opt}}(x) &\propto \left(\sum_{m=1}^{M} \alpha_m N(x|\mu_m, \Sigma_m) \right)^{\frac{d}{d+2}} \\ &\approx \sum_{m=1}^{M} \left(\alpha_m N(x|\mu_m, \Sigma_m) \right)^{\frac{d}{d+2}} \\ &\propto \sum_{m=1}^{M} \tilde{\alpha}_m N(x|\mu_m, \frac{d+2}{d}\Sigma_m) \qquad (3.4) \\ \tilde{\alpha}_m &= \frac{\left(\alpha_m |\Sigma_m|^{1/d}\right)^{\frac{d}{d+2}}}{\sum_{p=1}^{M} \left(\alpha_p |\Sigma_p|^{1/d}\right)^{\frac{d}{d+2}}} \qquad (3.5) \end{aligned}$$

where the second line follows from the well-separated assumption. Another implication of this assumption is that the EM algorithm has a very easy

time of estimating the true parameters. Employing these assumptions results in the model-based design technique given in [12]. That is, one begins by training a GMM from a set of example data using the EM algorithm. This results in the model parameters α_m, μ_m and Σ_m. One then sets the system parameters in terms of the model parameters according to Eqs. (3.4) and (3.5). It should be mentioned that under the well-separated assumption, the issues discussed in Section 3.1.1 about the inertial profile of a GMVQ do not apply. Since the codebooks will have negligible overlap, the system simply inherits the inertial profile of whichever component Gaussian quantizer is active in a given region. Thus, the design procedure can be modified to account for this, for example by changing the covariance scaling in Eq. (3.4) from $\frac{d+2}{d}$ to 3 in the case of scalar transform coders. In practice, of course, the source is not actually a well-separated GMM, and so this technique has only heuristic support as a general training procedure. Also note that this technique assumes MSE. Nevertheless, it has been seen to work quite well in practice, and is the standard approach.

To utilize model-based training in the recursive case, one would apply Maximum Likelihood to estimate the parameters of a dynamic model, such as an HMM (see Section 3.3). At each time step, then, this model would supply a conditional density in the form of a GMM of order M, and the coder parameters would be set in terms of the conditional density parameters using Eqs. (3.4) and (3.5). To justify the use of model-based training in the recursive case, it is necessary to assume that every conditional GMM supplied by the dynamic model is well separated. This assumption is straightforward in the case that the means $\widehat{x}_m$ and covariances $\widehat{\Sigma}_m$ are fixed, but not for more general dynamic models.

High Dimensions and Weighted ML Training

Notice in Eq. (3.3) that, for MSE distortion, the optimal point density approaches the probability density as d grows. This fact suggests that, in high dimensions, the problem of finding the best point density becomes equivalent to

that of estimating the true probability density, which is conventionally solved via Maximum Likelihood approaches. This subsection explores the asymptotic connection between Problem (3.2) and the classic minimum cross-entropy problem, which in turn gives rise to a *Weighted Maximum Likelihood* (WML) approach. A thorough discussion of entropy optimization problems, and their relationships to statistical estimation, can be found in [39]. The results of this section stem from the following theorem, which applies to any function $0 \leq f(x) < \infty$ and positive measure $p(x)$ with $\int p(x)\,dx < \infty$:

$$\left(\frac{\int p(x) f^q(x)\,dx}{\int p(x)\,dx}\right)^{1/q} \underset{q\to 0}{\searrow} \exp\left(\frac{\int p(x)\log f(x)\,dx}{\int p(x)\,dx}\right) \tag{3.6}$$

where $0 < q < 1$ and the downward arrow indicates that the left-hand side is monotonically decreasing. Details of this theorem can be found in Sections 6.6-8 of [40]. To apply Eq. (3.6) to Bennett's Integral, one would use $Y = \lambda_\theta^{-1}(x)$, $p(x) = |S(x)|^{1/d} f_{\mathrm{x}}(x)$ and $q = 2/d$. However, this introduces the complication that $p(x)$ depends on d via the sensitivity matrix. Rather than invoke further assumptions on the source parametrization and distortion measure such that $|S(x)|^{1/d}$ would tend to some fixed function of x, as would be required to obtain a rigorous limit in Eq. (3.6), we simply state the result as an approximation valid for large d. After some algebra, the above substitutions result in:

$$\mathrm{E}_{\mathrm{x}}\left(|S(x)|^{1/d}\lambda_\theta^{-2/d}(x)\right) \underset{d}{\gtrapprox} \mathrm{E}_{\mathrm{x}}\left(|S(x)|^{1/d}\right)\exp\left(\frac{2}{d}\frac{\mathrm{E}_{\mathrm{x}}\left(|S(x)|^{1/d}\log\lambda_\theta^{-1}(x)\right)}{\mathrm{E}_{\mathrm{x}}\left(|S(x)|^{1/d}\right)}\right) \tag{3.7}$$

where the symbol $\underset{d}{\gtrapprox}$ indicates that the right-hand side is a lower bound that becomes tight as d becomes large. Note that all point densities considered in this paper are Gaussian Mixtures: this ensures that $0 < \lambda_\theta^{-1}(x) < \infty$. Provided that $\mathrm{E}_{\mathrm{x}}\left(|S(x)|^{1/d}\right) < \infty$, then, for sufficiently large d, problem (3.2) can be approximated as:

$$\min_\theta \mathrm{E}_{\mathrm{x}}\left(|S(X)|^{1/d}\log\lambda_\theta^{-1}(x)\right) \tag{3.8}$$

This correspondence can also be established directly via Jensen's Inequality, as in [24]. Observe that (3.8) is the classic (weighted) Minimum Cross-Entropy problem: as is well known, the optimal solution to this problem is $\lambda_{\mathrm{WML}}(x) \propto |S(x)|^{1/d} f_{\mathrm{x}}(x)$. In the case where the distribution is unknown, and only N samples x_n drawn from it are available, the Strong Law of Large Numbers can be used to arrive at the Weighted Maximum Likelihood parameter estimation problem:

$$\max_{\theta} \sum_{n=1}^{N} |S(x_n)|^{1/d} \log \lambda_\theta(x_n) \tag{3.9}$$

Thus, this approximation is very convenient, in that it justifies using the well-studied tools of ML estimation directly to estimate the system parameters. Note that for the case of MSE distortion ($S(x) = I$), the result is exactly the classic ML problem. In the recursive case, one replaces $\lambda_\theta(x_n)$ with $\lambda_\theta(x_n|x_1, \ldots, x_{n-1})$, again justifying the use of Maximum Likelihood techniques. The WML approach is differentiated from the model-based design approach in that it considers the output of the training process as a model for the point density, not the source probability. Thus, the conditional density parameters supplied by the dynamic model are used directly in the WML case, bypassing Eqs. (3.4) and (3.5). The details of Weighted ML estimation for HMMs are discussed in Section 3.2.4. A drawback to this approach is that the approximation (the right-hand side of Eq. (3.7)) is a lower bound on the high-rate distortion, and so the issue of the tightness of the approximation becomes important. The next subsections explore this issue for the memoryless case.

3.1.3 Examples: Uniform, Gaussian and Well-Separated GMM

To illustrate the dependence of the large-d approximation on the source distribution, consider three examples: Uniform, Gaussian and Well-Separated GMM memoryless sources. In each case, the loss incurred by utilizing the large-d approximation (i.e., maximum likelihood) compared to the exact minimum distortion approach is examined. Said another way, this subsection considers the loss

incurred by using $\lambda_{\mathrm{WML}}(x) = f_{\mathrm{x}}(x)$ instead of the optimal $\lambda_{\mathrm{opt}}(x) \propto f_{\mathrm{x}}^{\frac{d+2}{d}}(x)$. For these examples, only MSE distortion ($S(x) = I$) is considered.

Uniform Sources

Notice that for the source distributed uniformly on some support $\mathcal{R}$, there is no penalty for using the large-d approximation. That is, $f_{\mathrm{x}}(x) \propto 1_{\mathcal{R}}(x)$ implies that $f_{\mathrm{x}}^{\frac{d}{d+2}}(x) \propto 1_{\mathcal{R}}(x)$, which in turn implies that $\lambda_{\mathrm{WML}}(x) = \lambda_{\mathrm{opt}}(x) = f_{\mathrm{x}}(x)$. Thus, there is no penalty for using the large-d approximation for uniform sources, even when $d = 1$.

Gaussian Sources

For the Gaussian source, the penalty for using the large-d approximation quickly becomes very small. Suppose that $f_{\mathrm{x}}(x) = N(x|\mu, \Sigma)$. The optimal point density is given as $\lambda_{\mathrm{opt}}(x) = N(x|\mu, \frac{d+2}{d}\Sigma)$, and the WML-optimal point density is $\lambda_{\mathrm{WML}}(x) = N(x|\mu, \Sigma)$. Compare the performance of these two point densities by examining the logarithm of the ratio of their high-rate distortions:

$$L_{\mathrm{WML}} = \frac{1}{2}\log_2 \frac{\mathrm{E}_{\mathrm{x}}\left(N(x|\mu, \Sigma)^{-2/d}\right)}{\mathrm{E}_{\mathrm{x}}\left(N(x|\mu, \frac{d+2}{d}\Sigma)^{-2/d}\right)}$$

where the scaling and logarithm base have been chosen so that the resulting loss is expressed in bits per dimension. For the cases $d = 1$ and $d = 2$, the upper expectation diverges, and the loss is infinite. This is expected, as the results underpinning the WML approach are not generally valid for $d = 1$ or $d = 2$, as indicated in Eq. (3.7). For $d \geq 3$, however, the following result holds:

$$L_{\mathrm{WML}} = \frac{d+1}{2}\log_2(d) - \frac{d+2}{4}\log_2(d+2) - \frac{d}{4}\log_2(d-2)$$

Notice that this result is independent of both μ and Σ. This loss is plotted as a function of d in Figure 3.4. Examining the plot, it is clear that the loss is fairly small for dimensions 3 and higher. In particular, for $d = 3$, the total loss is

less than 1 bit per vector. For dimensions above $d = 15$, the total loss is less than 0.1 bits per vector, so the penalty for using maximum likelihood quickly becomes very small.

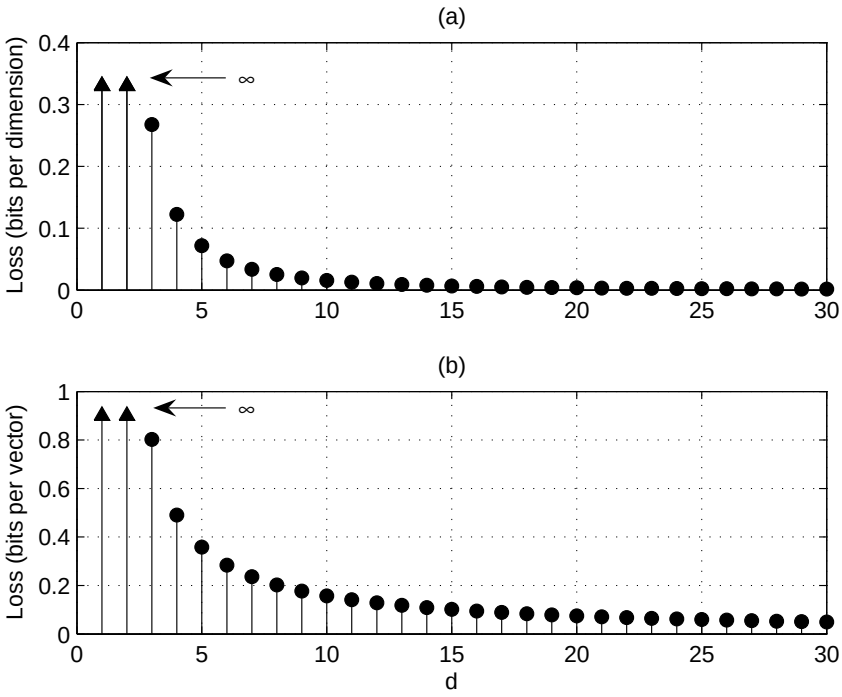

Figure 3.4: Point Density Loss when using WML approximation on a Multivariate Gaussian source. Plot (a) shows the loss in bits per dimension, and (b) shows loss in bits per vector. Notice that both approach zero as d increases.

Well-Separated GMM Sources

Finally, consider the case of a well-separated GMM source. As discussed in Section 3.1.2, the optimal point density in this case is a GM with the covariance of each Gaussian scaled as in the previous example, and the level allocation given by Eq. (3.5) in terms of the mixture weights α_m and the determinants of the covariances $|\Sigma_m|^{1/d}$. It has been shown in the previous example that the loss due to ignoring covariance scaling for a multivariate Gaussian source diminishes to zero as d grows large. This was expected, as the scaling factor applied to

Σ approaches 1 as d grows. However, Eq. (3.8) results in the level allocation $\widehat{\alpha}_m \propto \alpha_m |\Sigma_m|^{1/d}$ as d grows, rather than the simple $\widehat{\alpha}_m = \alpha_m$ formula obtained by using the pdf directly as the point density. It remains to evaluate the penalty for using the mixture weights directly as level allocations. To investigate this problem, suppose that $M = 2$, $\alpha_1 = \alpha_2 = \frac{1}{2}$, $\Sigma_1 = I$ and $\Sigma_2 = \sigma^2 I$. In order to isolate the effect of the bit allocation from that of covariance scaling, suppose that the WML system implements optimal covariance scaling. That is, $\lambda_{\mathrm{WML}}(x) = \frac{1}{2}(N(x|\mu_1, \frac{d+2}{d} I) + N(x|\mu_2, \frac{d+2}{d}\sigma^2 I))$. In this case, the loss of using the WML bit allocation is given, in bits per dimension, as:

$$\begin{aligned} L_\alpha &= \frac{1}{2}\log_2 \frac{\mathrm{E_x}\left(\lambda_{\mathrm{WML}}^{-2/d}(x)\right)}{\mathrm{E_x}\left(\lambda_{\mathrm{opt}}^{-2/d}(x)\right)} \\ &\approx \frac{1}{2}\log_2 \frac{\int_{\mathbb{R}^d} N(x|\mu_1, I)\left(\frac{1}{2}N(x|\mu_1, \frac{d+2}{d} I)\right)^{\frac{-2}{d}} dx + \int_{\mathbb{R}^d} N(x|\mu_2, \sigma^2 I)\left(\frac{1}{2}N(x|\mu_2, \frac{d+2}{d}\sigma^2 I)\right)^{\frac{-2}{d}} dx}{\int_{\mathbb{R}^d} N(x|\mu_1, I)\left(\frac{1}{1+\sigma^{\frac{2d}{d+2}}} N(x|\mu_1, \frac{d+2}{d} I)\right)^{\frac{-2}{d}} dx + \int_{\mathbb{R}^d} N(x|\mu_2, \sigma^2 I)\left(\frac{\sigma^{\frac{2d}{d+2}}}{1+\sigma^{\frac{2d}{d+2}}} N(x|\mu_2, \frac{d+2}{d}\sigma^2 I)\right)^{\frac{-2}{d}} dx} \\ &= \frac{1}{d} + \frac{1}{2}\log_2(1+\sigma^2) - \frac{d+2}{2d}\log_2\left(1+\sigma^{\frac{2d}{d+2}}\right) \end{aligned}$$

where the second line makes use of the well-separated assumption ($||\mu_1 - \mu_2|| \gg 1$). This loss function is plotted in Figure 3.5. An interesting effect arises here, which is that the loss goes to zero when measured in bits per dimension, but not when measured in terms of bits per vector. This is a potential drawback to using the large-d approximation, since practical systems are often operated with intermediate values of d wherein this loss may still be significant. Also notice that the loss disappears as σ^2 nears 1, as in that case the contribution of the covariance terms to the optimal bit allocation disappears. In scenarios where there is little variation in the determinants of the covariances, then, there is little penalty for using the high-d approximation. This example is easy to generalize to larger values of M, in which case the maximum loss (in bits per vector) is $\log_2 M$, corresponding to the case that a single covariance dominates.

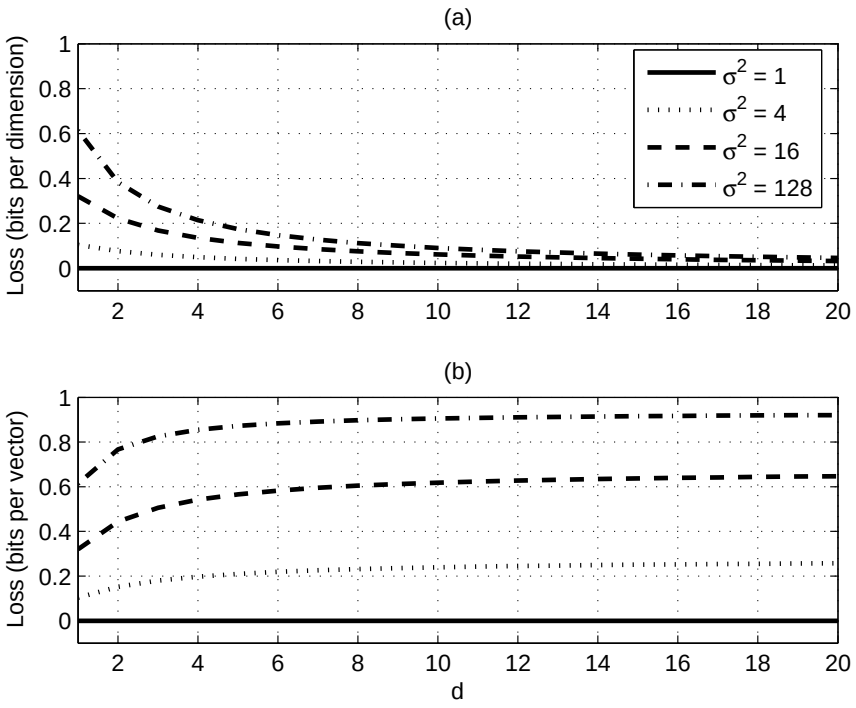

Figure 3.5: Loss for Ignoring Covariances in Bit Allocation among 2 Well-Separated Gaussians

3.1.4 Tightness for Wideband Speech

In light of the results of the previous subsections, the tightness of the large-d approximation needs to be investigated for real-world sources, such as wideband speech LSF's. To that end, this subsection discusses using GMMs trained from wideband speech data to estimate the loss incurred by using the probability density as the point density, compared to the optimal distortion for a source with the model pdf. This subsection will consider only MSE distortion. In this case, the loss can be expressed as:

$$L_{\mathrm{ML}} = \frac{1}{2}\left\{\log_2\left(\mathrm{E}\left(f_{\mathrm{M}}^{-\frac{2}{d}}(x)\right)\right) - \frac{d+2}{d}\log_2\left(\mathrm{E}\left(f_{\mathrm{M}}^{-\frac{2}{d+2}}(x)\right)\right)\right\} \tag{3.10}$$

where $f_{\mathrm{M}}(x)$ is an order-M GMM that approximates the source density. Here, the expectations are approximated by averages over data distributed accord-

ing to $f_{\mathrm{x}}(x)$. To this end, an order-16 GMM was trained on wideband speech LSF vectors of dimension 16 using the EM algorithm; this is f_{M}. Evaluating Eq. (3.10) on a disjoint test set of 60,000 vectors resulted in a loss of 0.0141 bits per dimension, or 0.225 bits per vector. In order to confirm this result, the experiment was repeated on a database of synthetic data drawn from $f_{\mathrm{M}}(x)$, and the results were identical to within double precision.

Another test was performed to check whether the model-based design technique would work better. In this test, the point density parameters were set using the model-based approach (Eqs. (3.4) and (3.5)). Then, the previous experiment was repeated to measure the loss of this point density with respect to the optimal; this resulted in exactly the same results as for the previous case. This is not surprising, since the variation in the determinants of the covariances was found to be small in this case. Taking all the results of this section together, it is argued that the problem of designing a GMVQ using random coders for 16-dimensional wideband speech LSF quantization is equivalent to the WML problem described by Eqs. (3.7-9). The conventional model-based approach can also be used, and should result in equivalent performance, as evidenced by the results of this subsection.

3.2 Models for Recursive Coding

This section reviews Hidden Markov Models with an eye towards use in recursive GMVQ quantizer systems. HMMs are well known, and details can be found in [41]. In the context of recursive coding with GMVQ systems, the HMM is used to produce a conditional density in the form of a GMM, which can then be used by the GMVQ system. For increased flexibility, a generalization of the conventional HMM is discussed. Finally, it is shown how to extend the Baum-Welch algorithm to the WML case.

3.2.1 Conventional HMM

Let s_n be a Markov chain taking values in $\{1, ..., M\}$ and denote the state transition matrix for s_n by $A \in R^{M \times M}$, with individual elements denoted by $a_{mj} = P(s_n = j | s_{n-1} = m)$. It is assumed that a sample x_n is conditionally independent of all other samples, given s_n: $f_{\mathrm{x}}(x_n | s_n = m, x_{n-1}, \ldots, x_0) = f_{\mathrm{x}}(x_n | s_n = m) = f_m(x_n)$. Let the density associated with state m be $f_m(x) = N(x | \mu_m, \Sigma_m)$, where $x \in \mathbb{R}^d$. It should be noted that the choice of a Gaussian Mixture Model for each state's density would also work, with minor modifications, but this work uses the single Gaussian case for notational simplicity. Denote by θ the set of parameters (A, μ, Σ, π), where π is the initial state distribution. This paper will assume that the Markov chain s_n is irreducible and so a stationary distribution p_s exists which can be obtained from the eigendecomposition of A. Further, it is assumed that $\pi = p_s$, so that the model defines a stationary process. Notice that an HMM reduces to a memoryless GMM model in the case that all rows of A are equal.

Denote by α_n the *a priori* state distribution at time n:

$$\alpha_n(m) \equiv P(s_n = m | x_{n-1}, x_{n-2}, ..., x_0) \tag{3.11}$$

Let β_n denote the *a posteriori* state distribution:

$$\beta_n(m) \equiv P(s_n = m | x_n, x_{n-1}, ..., x_0) \tag{3.12}$$

Given β_{n-1}, α_n may be obtained as follows:

$$\alpha_n(m) = \sum_{j=1}^{M} a_{jm} \beta_{n-1}(j) \tag{3.13}$$

Similarly, given α_n and x_n, β_n is obtained as follows:

$$\beta_n(m) = \frac{\alpha_n(m) N(x_n | \mu_m, \Sigma_m)}{\sum_{j=1}^{M} \alpha_n(j) N(x_n | \mu_j, \Sigma_j)} \tag{3.14}$$

To initialize the recursion, α_0 is set to the initial state distribution. Fi-

nally, the density of x_n conditioned on all of the previous data is given by:

$$\begin{aligned} f_{\mathrm{x}}(x_n|x_{n-1},...,x_0) &= \sum_{m=1}^{M} p(x_n, s_n = m|x_{n-1},...,x_n) \\ &= \sum_{m=1}^{M} \alpha_n(m) N(x_n|\mu_m, \Sigma_m) \end{aligned} \tag{3.15}$$

Thus, the density of the current data x_n, conditioned on all of the previous data, is an order-M Gaussian Mixture with mixture weights given by the state priors. Note that only the weights of the mixture components change with time, while the component means and covariances are fixed. In this sense, the HMM generalizes the GMM from a sequence of i.i.d. observations to a model with memory. It is proposed to construct a recursive quantizer based on an HMM by using a GMVQ with mixture weights updated at each time step. As discussed in [37], the complexity of the GMVQ system is low enough to permit updating the parameters in this way. In order to maintain synchronization between the encoder and decoder without sending side information, it is required that the update to the mixture weights depend only on past data. Examining the recursions for α_n and β_{n-1} (Eqs. (3.13) and (3.14)), it is clear that this is indeed the case. Note that only one previous quantized output needs to be stored in order to implement these recursions.

3.2.2 Generalized HMM

In [31], a recursive GMVQ system was presented based on a jointly-Gaussian Mixture Model of the source. That is, it is assumed that each datum x_n is conditionally independent of all previous data except x_{n-1}: $f_{\mathrm{x}}(x_n|x_{n-1},\ldots,x_0) = f_{\mathrm{x}}(x_n|x_{n-1})$. It is also assumed that the joint density of x_n and x_{n-1} is an order-M GMM. Thus, the conditional density is again an order-M GMM, with the cluster means and weights depending on x_{n-1}. The conditional covariances are constant. The ability to move the cluster means based on past data provides a very flexible mechanism for exploiting information from the previous sample. However, this

approach has the disadvantage that it ignores any dependence on samples farther back in time. Conversely, the conventional HMM provides a simple model of the dependence on all previous data, but is not as flexible in describing the dependence on the previous sample, which may be significant. It is proposed to generalize the joint-GMM model by adding on a Hidden Markov structure, much in the same way that the conventional HMM generalizes the memoryless GMM. The idea is that the joint-GMM structure can exploit the strong dependence on the previous sample, while the Markov structure will model longer-term dependency.

This generalized model weakens the usual HMM assumption that a given sample, x_n, is conditionally independent of all other data given the current state s_n. Instead, x_n will still depend on x_{n-1}:

$$f_{\mathrm{x}}(x_n|s_n = m, x_{n-1}, \ldots, x_0) = N(x_n|\tilde{\mu}_m(x_{n-1}), \Sigma_m) \tag{3.16}$$

$$\tilde{\mu}_m(x_{n-1}) = \mu_m^{\mathrm{x}} + \Omega_m(x_{n-1} - \mu_m^{\mathrm{y}}) \tag{3.17}$$

The parameters Ω_m, μ_m^{x} and μ_m^{y} are discussed in detail in Section 3.2.4. Adopting this model changes the derivation of the conditional density only slightly. In particular, the definitions of α_n and β_n remain the same, and the update formula for α_n (Eq. (3.13)) is unchanged. The conditional covariance matrix, Σ_m is a constant. The update formula for β_n then becomes:

$$\begin{aligned} \beta_n(m) &= P(s_n = m|x_n, x_{n-1}, ..., x_0) \\ &= \frac{p(s_n = m, x_n|x_{n-1}, ..., x_0)}{f_{\mathrm{x}}(x_n|x_{n-1}, ..., x_0)} \\ &= \frac{\alpha_n(m)N(x_n|\tilde{\mu}_m(x_{n-1}), \Sigma_m)}{\sum_{j=1}^{M} \alpha_n(j)N(x_n|\tilde{\mu}_j(x_{n-1}), \Sigma_j)} \end{aligned} \tag{3.18}$$

This gives the the conditional density of the current data:

$$\begin{aligned} f_{\mathrm{x}}(x_n|x_{n-1}, ..., x_0) &= \sum_{m=1}^{M} P(s_n = m|x_{n-1}, \ldots, x_0) f_{\mathrm{x}}(x_n|s_n = m, x_{n-1}, \ldots, x_0) \\ &= \sum_{m=1}^{M} \alpha_n(m)N(x_n|\tilde{\mu}_m(x_{n-1}), \Sigma_m) \end{aligned} \tag{3.19}$$

Thus, the conditional density is again an order-M Gaussian Mixture density. Note that in this case, the mixture weights are adjusted by the HMM structure, as before, and that the component means are adjusted at each step by the jointly-Gaussian structure. Thus, use of this model requires updating the means as in Eq. (3.17), as well as the mixture weights. Note that, again, only the previous quantized vector needs to be stored.

3.2.3 Summary of Recursive Procedures

The recursive update and coding procedure is summarized below, for both the conventional HMM and generalized HMM. Certain steps apply only to the generalized HMM and are indicated as such. As illustrated in Figure 3.2, the goal of the recursion is to produce, for each sample x_n, an order-M Gaussian Mixture. That is, an updated set of parameters $\alpha_n(m)$ and $\tilde{\mu}_m$ representing the conditional density of x_n given all previous samples. In all cases, the covariances Σ_m remain fixed, and in the conventional HMM the means also remain fixed with $\tilde{\mu}_m = \mu_m$. In the case that the WML training approach was used, these parameters represent the conditional point density, and so are used directly as the GMVQ parameters. In the model-based training framework, the parameters are considered as describing the conditional probability density of the source, and so Eqs. (3.4) and (3.5) are then applied to obtain the GMVQ parameters.

- Initialize the parameters and quantize the first sample (i.e., for $n = 0$).
    1. Initialize $\alpha_0(m)$ to the stationary distribution p_s.
    2. (generalized HMM) Initialize $\tilde{\mu}_m$ to μ_m.
    3. Quantize x_0 using the initial settings.
- Recursively update the parameters and quantize successive samples (i.e., for $n \in \{1, 2, \ldots\}$).
    1. Use $\widehat{x}_{n-1}$ to compute β_{n-1} using Eq. (3.14) for conventional HMM or Eq. (3.18) for generalized HMM.

2. Use β_{n-1} and A to compute α_n using Eq. (3.13).

3. (generalized HMM) Compute $\tilde{\mu}_m$ using Eq. (3.17).

4. Quantize x_n using the updated parameters.

3.2.4 Weighted ML Training for HMMs

In light of the results of Section 3.1, a Weighted Maximum Likelihood technique (i.e., Eq. (3.9)) is used to set the model parameters. WML is used instead of regular Maximum Likelihood because it is able to incorporate non-mean-square distortion measures into the training process. That is, to train a system based on the regular HMM, we apply the Baum-Welch algorithm to a database of example vectors. The only modification from the classic Baum-Welch algorithm is that the contribution of each piece of data x_n is weighted by $|S(x_n)|^{1/d}$. This change has no effect on the E-step of the algorithm, and the forward and backward variables are computed in the usual way:

$$\alpha_{n+1}(j) = \left[\sum_{i=1}^{M} \alpha_n(m) a_{mj}\right] f_j(x_{n+1}) \tag{3.20}$$

$$\psi_n(m) = \sum_{j=1}^{M} a_{mj}\psi_{n+1}(j) f_j(x_{n+1}) \tag{3.21}$$

For convenience, define the following two variables:

$$\xi_n(m,j) = \frac{\alpha_n(m) a_{mj} f_j(x_{n+1})\psi_{n+1}(j)}{\sum_{m=1}^{M}\sum_{j=1}^{M} \alpha_n(m) a_{mj} f_j(x_{n+1})\psi_{n+1}(j)} \tag{3.22}$$

$$\gamma_n(m) = \sum_{j=1}^{M} \xi_n(m,j) \tag{3.23}$$

To incorporate the weightings, the M-step is slightly modified as follows:

$$a_{mj} = \frac{\sum_{n=1}^{N} |S(x_n)|^{1/d} \xi_n(m,j)}{\sum_{n=1}^{N} |S(x_n)|^{1/d} \gamma_n(m)} \tag{3.24}$$

$$\mu_m = \frac{\sum_{n=1}^{N} |S(x_n)|^{1/d} \gamma_n(m) x_n}{\sum_{n=1}^{N} |S(x_n)|^{1/d} \gamma_n(m)} \tag{3.25}$$

$$\Sigma_m = \frac{\sum_{n=1}^{N} |S(x_n)|^{1/d} \gamma_n(m)(x_n - \mu_m)(x_n - \mu_m)^T}{\sum_{n=1}^{N} |S(x_n)|^{1/d} \gamma_n(m)} \tag{3.26}$$

Note that weightings other than $|S(x_n)!^{1/d}$ could be used to handle suboptimal cases. For example, if the GMVQ system were to encode under a mismatched distortion measure $d_2(x_1, x_2) = (x_1 - x_2)^{\mathsf{T}} Q(x_n)(x_1 - x_2)$, the appropriate weighting would be $\mathrm{tr}\,(Q^{-1}(x_n)S(x_n))$, as discussed in [20]. Such a weighting would redistribute the point density in order to compensate for suboptimal encoding. Thus, WML can be extended to the case of mismatched distortion measures by appropriate choice of weighting terms. Also note that this weighted Baum-Welch algorithm can easily be specialized to a weighted EM algorithm to handle the memoryless case.

To handle the generalized HMM, successive samples are "stacked" into vectors of dimension $2d$, and then the weighted Baum-Welch algorithm is applied to this new database. That is, the n-th sample in the stacked database is a concatenation of x_n and x_{n-1}. The weighting term for each sample in the stacked database is the same as in the previous case, $|S(x_n)|^{1/d}$, as the desired result is a weighted conditional density for x_n. This results in a model of the joint-density of pairs of successive vectors, which is then converted into the conditional densities, as described in [31]. That is, the result of the training process is a transition matrix A, a set of mean vectors $\nu_m \in \mathbb{R}^{2d}$ and a set of covariances $\Phi_m \in \mathbb{R}^{2d \times 2d}$. These parameters can be partitioned as:

$$\nu_m = \begin{bmatrix} \mu_m^{\mathrm{x}} \\ \mu_m^{\mathrm{y}} \end{bmatrix}$$
$$\Phi_m = \begin{bmatrix} \Sigma_m^{\mathrm{xx}} & \Sigma_m^{\mathrm{xy}} \\ \Sigma_m^{\mathrm{yx}} & \Sigma_m^{\mathrm{yy}} \end{bmatrix}$$

where $\mu_m^{\mathrm{x}}, \mu_m^{\mathrm{y}} \in \mathbb{R}^d$ and $\Sigma_m^{\mathrm{xx}}, \Sigma_m^{\mathrm{yy}} \in \mathbb{R}^{d \times d}$. The desired conditional density parameters can then be extracted as:

$$\Sigma_m = \Sigma_m^{\mathrm{xx}} - \Sigma_m^{\mathrm{xy}} (\Sigma_m^{\mathrm{yy}})^{-1} \Sigma_m^{\mathrm{yx}}$$
$$\Omega_m = \Sigma_m^{\mathrm{xy}} (\Sigma_m^{\mathrm{yy}})^{-1}$$

Note that Σ_m is interpreted as a conditional covariance in this context. Moreover, it will have a smaller determinant than in the fixed-mean case, owing to the second term above. Thus the conditional densities will be "tighter" in this case, reflecting the improved flexibility of the model. Notice that this "stacked" approach can be specialized to the training of joint-GMM models in the same way the Baum-Welch algorithm can be specialized to memoryless GMM models.

3.3 Implementation of Recursive Coders

This section discusses the implementation of GMM-based recursive coders using HMMs. First, issues related to the basic GMVQ system are discussed; these issues apply to all of the systems under consideration. Next, specifics of the recursive update process are provided for each of the recursive systems under consideration.

3.3.1 Basic GMVQ Issues

A number of issues arise in implementing the GMVQ system. This paper utilizes the CURTZ system proposed by Shabestary in [19] and [34] to implement

the component Gaussian quantizers. This system is an extension of the classic Gaussian transform coder, and obtains improved performance by incorporating random coding techniques. It operates by applying a KLT to the input vector, and then using a bank of scalar Gaussian compressor functions. The compressed components are then quantized using L rectangular lattices, which are randomly offset from one another. The outputs of each quantizer are then fed into Gaussian expander functions (the inverses of the compressor functions), and the inverse KLT is applied to each of them. Finally, the best of the L candidates is chosen using a VQ, under input-weighted squared error. For $L = 1$, this system reduces to the classic scalar transform coder, and as L approaches 2^d, its performance becomes very close to random coding. Note that the encoding complexity is independent of the rate of operation, and linear in L. That said, the complexity can become prohibitively large for high values of L and d. As such, this paper considers only two cases: $L = 2^{d/2}$, which represents an intermediate complexity/performance point, and $L = 1$ (the regular scalar transform coder), which represents a minimal complexity. Further details of the implementation of CURTZ systems can be found in [19] and [34]. In particular, it is demonstrated in [19] that the CURTZ system attains performance very close to random coding when $L = 2^d$.

In addition to the usual costs of operating these systems in a memoryless setting, the bit allocation must be recomputed every time the system parameters are updated. Since the covariance matrices are left constant in all systems under consideration in this paper, no eigendecompositions or matrix inverses need by computed during operation. The bit allocation process used in this paper differs somewhat from that proposed in [19], and so is described here. The first step is to compute the number of codepoints to be assigned to each component coder, which is given by $N_m = \alpha_m 2^r$. Note that this number is likely not an integer; this issue will be dealt with later in the bit allocation process. The next step is to allocate levels amongst the d dimensions of each component quantizer. This is accomplished by dividing N_m by L and applying regular level allocation techniques

for a transform coder (since the CURTZ system will produce L codepoints for every one in the underlying transform coder it generalizes). That is, levels are assigned in proportion to the standard deviations of each dimension:

$$N_{mi} = \left(\frac{N_m}{L}\right)^{1/d} \frac{\sigma_{mi}}{(\prod_{i=1}^{d} \sigma_{mi})^{1/d}} \tag{3.27}$$

where σ_{mi}^2 is the i-th eigenvalue of the Σ_m. Again, N_{mi} is not guaranteed to be an integer. To resolve this issue, a pruning algorithm is applied. The first step in pruning is to round up each N_{mi} to the nearest integer. Assuming that the dimensions are ordered in increasing σ_{mi}, a correction is applied to each dimension in turn:

$$N_{mi} = N_{mi} - \left\lfloor N_{mi} \left(1 - \frac{N_m}{L \prod_{i=1}^{d} N_{mi}}\right) \right\rfloor$$

Finally, N_{md} is decremented by 1, ensuring that the total allocation is less than the target rate r. The total complexity of the bit allocation process is as follows: $Md(d+2) + M + 1$ multiplies, $M(2d+1)$ additions, $2Md$ rounding operations and M power computations.

3.3.2 Recursive Updates

This subsection discusses the complexity of implementing recursive updates. The total complexity is that of the basic GMVQ system, as described above, plus the additional complexity required to implement the parameter update, described in detail below. The complexities of recursive updates are summarized in Table 3.1.

Conventional HMM

In the case of the conventional HMM, the recursive update procedure consists of updating α_n. To carry out this update, one must store the previously quantized vector, $\hat{x}_{n-1}$, and apply Eq. (3.14) to find β_{n-1}. Then, α_n is found

by multiplying β_{n-1} by A, as in Eq. (3.13). Thus, the total complexity of the update process consists of M multivariate Gaussian density evaluations, $M(M+2)$ multiplications and M^2-1 additions.

Generalized HMM

The generalized HMM inherits all of the structure of the conventional HMM, and also requires the conditional means to be updated according to Eq. (3.17). Notice that the means must be updated after the density evaluations, as in Eq. (3.19). The total complexity of the update for this system is then: M multivariate Gaussian density evaluations, $M(M+2)+d^2$ multiplications and $M^2+d^2+d(2M-1)-1$ additions.

Table 3.1: Additional Complexity for Recursive Systems

	Multiplies	Additions	Density Evaluations
Conventional HMM	$M(M+2)$	M^2-1	M
Generalized HMM	$M(M+2)+d^2$	$M^2+d^2+d(2M-1)-1$	M

3.4 Practical Results

This section demonstrates the performance of the proposed recursive systems on the problem of wideband speech spectrum coding, under the Log Spectral Distortion measure. A training set of 300,000 wideband speech LSF vectors of order 16 was utilized, and the LSD-sensitivity matrix for each vector was evaluated using the method described in [20]. It should be noted that, for the high-rate analysis, LSD is measured in dB^2, and so predictions and training use this metric. This is done in order to correspond to a squared-error type of distortion measure, as is used in the high-rate analysis. However, when calculating operating point and outlier statistics, the more conventional approach of measuring in dB is used. For testing purposes, an independent database of 65,000 LSF vectors was utilized.

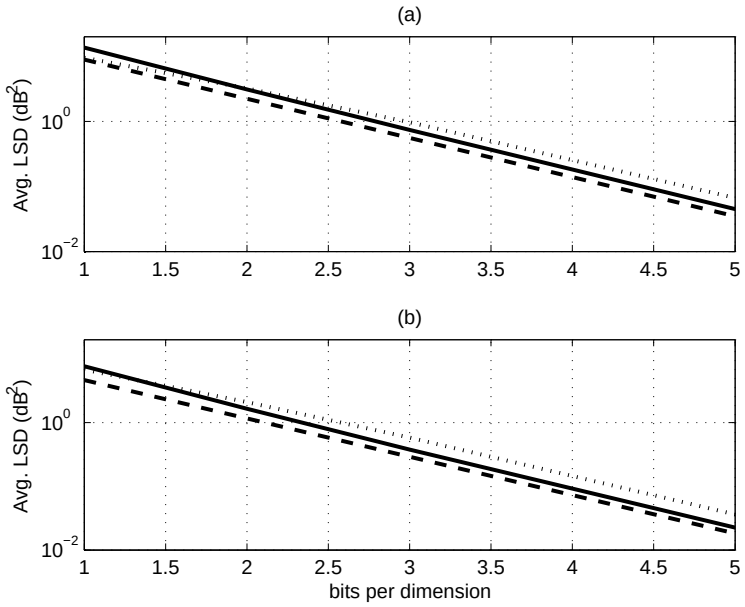

Figure 3.6: Performance of (a) Memoryless GMM and (b) joint-GMM Systems. The dashed lines indicate the high-rate estimates for random coders. The solid lines indicate the performance using CURTZ systems with $L = 2^{d/2}$, and the dotted lines indicate the performance using scalar transform coders.

Memoryless GMVQ

To begin with, a memoryless GMVQ system was constructed. To train this system, a Weighted EM algorithm was utilized, as described in Section 3.2. It was found that for model orders above $M = 16$, there was no significant improvement in performance, and so this order is used throughout the tests. To evaluate the performance, two systems were used: scalar transform coders (as in [12]) and CURTZ systems (see [19]) with $L = 2^{d/2} = 256$. The estimated high-rate performance of a random coder with the specified point density is also plotted for comparison. The performance of this system over a wide range of rates is seen in Figure 3.6a.

Notice that, for high rates (above 3 bits per dimension) the CURTZ system achieves performance midway between that of the transform coder-based

Table 3.2: Spectral Distortion Performance of Memoryless Systems Around Operating Point

bits/frame	CURTZ ($L = 2^{d/2}$)			Transform Coder		
	Avg. LSD (in dB)	Outliers (in %)		Avg. LSD (in dB)	Outliers (in %)	
		2-4 dB	> 4 dB		2-4 dB	> 4 dB
42	1.108	0.191	0	1.209	1.912	0
43	1.060	0.091	0	1.162	1.208	0
44	1.014	0.062	0	1.117	0.954	0
45	0.970	0.029	0	1.078	0.629	0
46	0.926	0.028	0	1.034	0.410	0.002
47	0.882	0.020	0	0.993	0.316	0

system and a true random coder. Specifically, the CURTZ system achieved around 0.3 bits per dimension of improvement (around 5 bits per vector) on the transform coder, and is around 0.2 bits per dimension (around 3 bits per vector) behind a true random coder. This is expected, since an intermediate value of L is utilized. Also notice that the transform coder system performs better than the CURTZ system at low rates; it is not known why this effect arises, but it implies that the savings at rates of interest (around 1dB LSD; a bit less than 3 bits per dimension) may not be as large as at high rates. The performance of the systems at rates of interest is shown in Table 3.2. Note that the CURTZ-based system is able to achieve transparent quality (less than 1dB average distortion, less than 1% small outliers, and negligible large outliers) between 44 and 45 bits per vector, while the transform-coder based system requires 46-47 bits per vector, a savings of 1-2 bits per vector. This is indeed smaller than the 5 bits observed at higher rates, but is significant nonetheless.

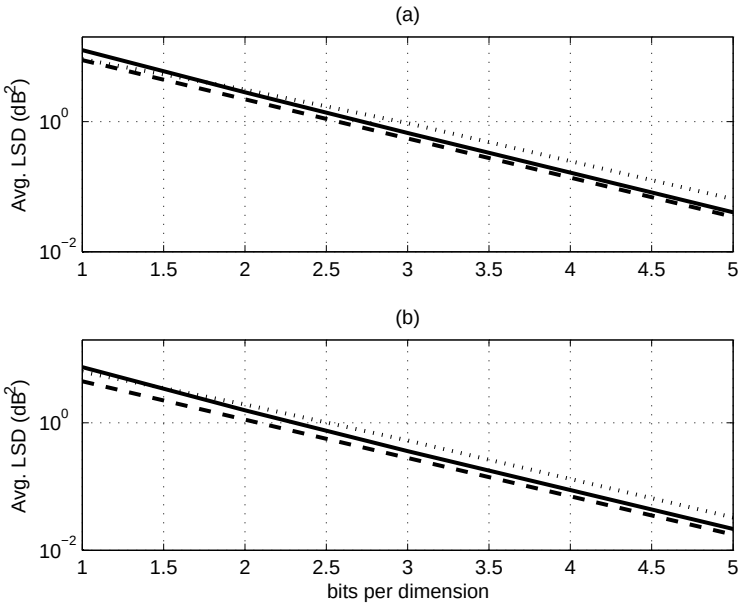

Figure 3.7: Performance of (a) HMM and (b) Generalized HMM Systems. The dashed lines indicate the high-rate estimates for random coders. The solid lines indicate the performance using CURTZ systems with $L = 2^{d/2}$, and the dotted lines indicate the performance using conventional transform coders.

Table 3.3: Spectral Distortion Performance of HMM Systems Around Operating Point

bits/frame	CURTZ ($L = 2^{d/2}$)			Transform Coder		
	Avg. LSD (in dB)	Outliers (in %)		Avg. LSD (in dB)	Outliers (in %)	
		2-4 dB	> 4 dB		2-4 dB	> 4 dB
41	1.096	0.238	0	1.230	2.355	0
42	1.044	0.143	0	1.184	1.821	0.003
43	1.001	0.104	0	1.141	1.294	0
44	0.954	0.056	0	1.100	0.912	0
45	0.911	0.036	0	1.056	0.744	0
46	0.873	0.023	0	1.015	0.579	0
47	0.836	0.014	0	0.978	0.405	0

Conventional HMM

Next, an order-16 HMM was trained, again using the Weighted ML technique of Section 3.2.4. The performance of this system over a wide range of rates is seen in Figure 3.7a, and the performance around the desired operating point is tabulated in Table 3.3. Notice that the HMM-based system requires 1-2 bits per frame fewer than the memoryless GMVQ system to achieve transparent quality when employing CURTZ coders. The transform coder system did not show as much improvement, lagging about 3 bits behind the CURTZ system. Note that in both the GMM and HMM cases, use of the CURTZ system resulted in greatly improved outlier performance; in particular, the CURTZ-based systems display roughly one fifth as many outliers around the operating point of 1dB average LSD.

Generalized HMM and Joint-GMM

Finally, two more systems were tested, using the joint-GMM and generalized HMM models. The performance of these systems over a wide range of rates

Table 3.4: Spectral Distortion Performance of Joint GMM Systems Around Operating Point

bits/frame	CURTZ ($L = 2^{d/2}$)			Transform Coder		
	Avg. LSD (in dB)	Outliers (in %)		Avg. LSD (in dB)	Outliers (in %)	
		2-4 dB	> 4 dB		2-4 dB	> 4 dB
34	1.121	1.564	0.002	1.281	4.936	0.023
35	1.070	1.092	0	1.232	3.854	0.019
36	1.022	0.724	0	1.182	2.999	0.020
37	0.975	0.459	0	1.136	2.385	0.011
38	0.931	0.331	0	1.092	1.833	0.015
39	0.888	0.206	0	1.047	1.451	0.011
40	0.847	0.169	0	1.007	1.101	0.014
41	0.810	0.096	0	0.967	0.846	0.011

are shown in Figures 3.6b and 3.7b, respectively. Notice that both of these systems show a high-rate advantage of 0.5 bits per dimension (8 bits per frame) relative to their fixed-mean counterparts. The performance of the systems around the desired operating point is shown in Tables 3.4 and 3.5, respectively. The CURTZ-based systems achieve transparent quality at 36-37 bits per frame and 36 bits per frame, respectively, resulting in a 0.5-1 bit gain due to using the HMM structure. The transform coder-based systems required 40-41 and 39-40 bits per frame, respectively, again consistent with previous results. An interesting effect is that the difference in outliers between the CURTZ and transform coder based systems is smaller in this case, suggesting that allowing mean updates has a strong impact on outlier performance.

Table 3.5: Spectral Distortion Performance of Generalized HMM Systems Around Operating Point

bits/frame	CURTZ ($L = 2^{d/2}$)			Transform Coder		
	Avg. LSD (in dB)	Outliers (in %)		Avg. LSD (in dB)	Outliers (in %)	
		2-4 dB	> 4 dB		2-4 dB	> 4 dB
33	1.153	1.902	0.002	1.285	4.284	0.017
34	1.099	1.297	0	1.235	3.454	0.028
35	1.048	0.820	0	1.187	2.636	0.017
36	1.000	0.553	0	1.139	1.970	0.017
37	0.955	0.324	0	1.095	1.524	0.011
38	0.912	0.219	0	1.051	1.197	0.009
39	0.871	0.137	0	1.011	0.914	0.006
40	0.830	0.085	0	0.970	0.706	0.008
41	0.784	0.046	0	0.925	0.554	0.006

3.5 Discussion

This chapter presented a framework for extending GMVQ systems to the recursive case by utilizing HMMs and generalizations. The additional complexity required by these approaches is minimal, and large increases in performance can be achieved in the wideband speech LSF quantization problem. The problem of training the parameters of the systems to minimize estimated high-rate distortion was investigated, and it was concluded that Maximum Likelihood approaches can be used to solve this problem in the case of random coders and large dimensions. A weighted extension to Maximum Likelihood was proposed for dealing with the case of input-weighted squared error measures, which allows the inclusion of Log Spectral Distortion.

Motivated by the random coding argument used in justifying the training approach, CURTZ systems were employed to implement GMVQs with performance close to that predicted by the high rate estimates. These systems are convenient in that they allow the user to approach random-coding performance while retaining rate-independent complexity, which is important in applications that require large codebooks, such as wideband speech LSF quantization. Since these systems can scale smoothly between the conventional transform coder and random coding, they provide a bridge between the estimation results, which assume random coding, and the popular practice of using transform coders in GMVQ systems.

The proposed systems were applied to the problem of wideband speech LSF quantization under the Log Spectral Distortion measure. It was found that employing CURTZ systems results in an improvement in average distortion equivalent to 3 bits per frame relative to transform coders, in both the memoryless and recursive cases. The outlier performance of the CURTZ systems were seen to be far superior in the memoryless and HMM case, and were somewhat superior in the joint-GMM and generalized HMM cases. This implies that allowing mean updates in the recursive structure can result in strong improvements in outliers, in addition

to the roughly 8 bits per vector savings apparent in the average distortion. Employing a Markov structure for updating the mixture weights was found to provide 1-2 bits per frame of improvement, both in the case of fixed means (conventional HMM) and variable means (generalized HMM). When employing CURTZ coders and generalized HMM recursion, transparent quality was achieved at a rates of 36 bits per frame.

The text of this chapter is a reprint of a paper coauthored with Bhaskar D. Rao which has been published in the March 2007 issue of *IEEE Transactions on Audio, Speech and Language Processing* under the title *"High-Rate Optimized Recursive Vector Quantization Structures Using Hidden Markov Models"*. The dissertation author was the primary researcher and author, and the co-author contributed to or supervised the research which forms the basis for this chapter.

4 Speaker-Dependent Wideband Speech Coding

This chapter examines the problem of speaker-dependent wideband speech coding. Speaker-dependent systems have been used in a variety of speech processing applications. Most prominent is the field of speech recognition, where their performance advantages relative to speaker-independent systems are well known [42]. They have also been employed in speech enhancement settings (c.f. [44]). In the realm of coding, they have been applied in the area of low-rate compression, particularly in the context of phonetic vocoding (see [46], [43]). More recently, speaker-dependent systems have been applied in the context of concatenative text-to-speech synthesizers in [45], attaining toll quality output. In this work, we examine the potential of speaker-dependent systems in the context of CELP (Code Excited Linear Prediction) coding of wideband speech for telecommunications.

Conventional approaches to speech coding are speaker-independent, employing a single coder designed to work for any speaker. This conventional approach has the advantage of simplicity: only one coder needs to be designed, and the design can be carried out ahead of time using a single large, multispeaker database. However, since the statistics of various coder parameters vary widely from speaker to speaker, speaker-dependent coding offers the promise of improved performance. The recent development of the GMVQ system, discussed in the previous chapter, provides a flexible coding framework that is able to incorporate arbitrary source statistics and distortion measures. This framework, then, enables

the study and implementation of speaker-dependent coding. A number of issues arise in exploiting this potential. Foremost is the profusion of coders required: a separate design process must be performed for each speaker, and the resulting coders must be distributed to the appropriate locations to enable communication. Since it is impractical to collect an example database of every possible speaker, the training process cannot be carried out ahead of time and must instead be implemented in an on-line fashion. The costs of distributing the speaker-dependent designs to the required locations must also be carefully considered, lest this overhead wipe out the gains that speaker-dependent coding provides. Another issue is robustness with respect to "incorrect" speakers. This chapter seeks to experimentally quantify the performance gains provided by speaker-dependent coding and to address these various implementational issues in a comprehensive fashion.

Note that, in a telecommunications setting, the variation in user inputs will depend not only on differences between the speakers themselves, but also on differences in background noise, acoustic environment and telephony equipment. These factors distinguish what can be termed "user-dependent" gains from purely speaker-dependent gains, which depend only on variations in speech patterns and anatomy. One can imagine scenarios in which the user-dependent gains would be smaller than the speaker-dependent gains. For example, if all users suffered from background noise of an identical character and level, the statistical variation between them would decrease, and along with it the user-dependent gains. Conversely, there are scenarios in which user-dependent gains exceed speaker-dependent gains. That is, if users are exposed to widely different background noise or microphone acoustics, the statistical variation between them would increase, and so the user-dependent gains would exceed the speaker-dependent gains. Unfortunately for our purposes, there do not exist any agreed-upon models for how these various environmental factors vary with different users, and so we restrict attention to quantifying the gains due solely to speaker-dependence. While this is only the first step in understanding the potential gains available in a fully user-dependent

setting, it should be noted that that various techniques for online learning and robustness presented here are directly applicable to the broader case.

In order to investigate the performance of speaker-dependent wideband speech coding, a simplified (and somewhat idealized) CELP framework is utilized, as described in Section 4.1. This approach allows us to examine the speaker-dependence of various coder parameters common to a wide variety of speech coders; namely, the LPC coefficients, adaptive codebook parameters and fixed codebook parameters. In order to quantify the performance of speaker-independent coding, we require a sufficiently general class of quantizers to represent the statistical variations between speakers. To this end, we utilize the Gaussian Mixture Vector Quantizer (GMVQ) system presented in [12] (see Figure 4.1). In addition to the ability to represent reasonably arbitrary source statistics, this system has a number of properties that make it attractive in the speaker-dependent context. Chief among these is the parametric form of the coder design: only a small number of rate-independent parameters are required to describe the coder. For a GMVQ of order M, operating in dimension d, this consists of $M(1+d+d(d+1)/2)-1$ scalar parameters. Thus, the number of parameters that must be transmitted and stored for each speaker is small, and independent on the rate of operation. Additionally, when the component Gaussian quantizers are implemented using scalar transform coders (as is common practice), the encoding complexity is linear in d, allowing operation in large dimensions. This is particularly important for the case of the fixed excitation, in which case d equals the subframe size. Also, the values of M typically used in coding are in the range of 10-20, making it feasible to train entire speaker-dependent systems. Contrast this with the case of speech recognition, in which much higher model orders are required in, for example, triphone modeling. The high model order, and the lack of supervised data on which to perform speaker adaptation, gives rise to suboptimal approaches such as Maximum Likelihood Linear Transform (MLLT). In such a suboptimal approach, the speaker-dependent system is produced by applying a linear transformation to a speaker-independent

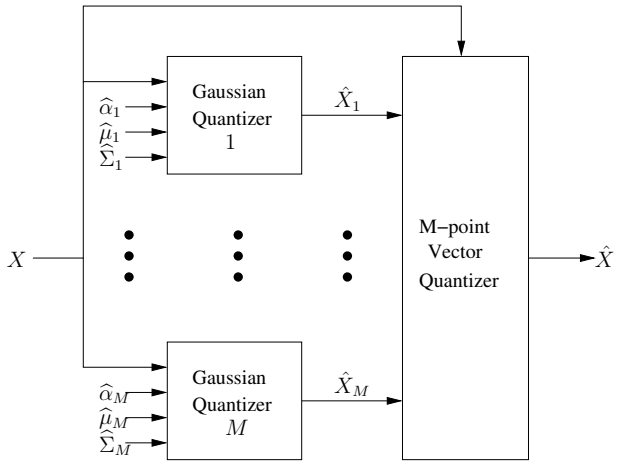

Figure 4.1: The Gaussian Mixture Vector Quantizer system. Each component quantizer produces a Gaussian point density $N(x|\mu_m, \Sigma_m)$. The parameters α_m specify the proportion of codepoints allocated to each component quantizer, resulting in an overall point density of $\sum_{m=1}^{M} \alpha_m N(x|\mu_m, \Sigma_m)$.

system. This reduces the number of speaker-dependent parameters which must be learned and, in the case of cepstrum features, restricts the adaptation to acoustic features, leaving the underlying language structure (which was learned from labeled data) intact. In the coding context, however, the model order is sufficiently low that the reduction in parameters obtained from MLLT is less dramatic. Furthermore, coder design is typically an unsupervised process, driven entirely by the input data, and so there is no reason to restrict the scope of speaker-dependent learning. Moreover, even if a suboptimal approach was required, the benchmark for performance would be that of the unrestricted approach. For these reasons, we focus on unrestricted learning of entire coders for each speaker.

In Section 4.1, the GMVQ system is applied to coding of the parameters of a simplified CELP system in order to quantify the speaker-dependence of each type of parameter. It is found that the LSF parameters exhibit roughly a 10% gain in rate-distortion performance (i.e., 4 bits per frame). In voiced frames, similar gains are achievable for the pitch lags, however, in unvoiced frames, the

pitch-lag statistics exhibit a nearly-uniform distribution. Furthermore, the pitch gains exhibit negligible speaker-dependence, and so there may be little benefit to speaker-dependent coding of the adaptive codebook parameters. Lastly, it is shown that much larger gains, on the order of 40-50 bits per frame, are achievable with speaker-dependent coding of the fixed excitation. This makes sense, given that such a large portion of the bit budget is typically spent on the fixed codebook, and that in frames where the LPC model is less salient (i.e., unvoiced frames), most of the burden of representing the input speech falls on the fixed codebook. Next, Section 4.2 discusses the different ways in which speaker-dependent gains can be exploited. Besides improvements in the rate-distortion sense, the gains can be used to reduce coding complexity and storage while leaving the rate-distortion performance unchanged. Additionally, if speaker-dependent rates are permitted, uniform quality can be achieved across all speakers. Lastly, the use of a safety-net approach is considered. Such a system operates a speaker-independent coder in parallel with the speaker-dependent coder, thereby providing robustness against "incorrect" speakers. In particular, we show how the safety-net approach can be naturally incorporated into the GMVQ system and present a modified training algorithm that allows a precise trade-off of robustness and performance.

Section 4.3 discusses issues pertinent to the training and distribution of speaker-dependent coders. Three different online training architectures are considered, each of which strikes a different balance between training complexity, overhead and performance. As a component of these architectures, methods for learning on quantized data are presented, which enable the design to be carried out at remote locations. This allows synchronized learning, obviating the need to transmit the coder designs explicitly. The benefits and disadvantages of synchronized learning are compared to single-point learning and explicit transmission of coders. Next, methods for recursive learning are examined. These approaches perform the learning process in a frame-by-frame manner, removing the need to store large training databases and enabling adaptive operation. Section 4.4 contains a

discussion of the results.

4.1 Performance of Speaker-Dependent Systems

This section demonstrates the performance of the proposed systems on the problem of wideband speech coding using a CELP framework (see Figure 4.2). Here, $A(z)$ denotes the analysis filter for a given subframe, and $W(z)$ the corresponding perceptual weighting filter given by $W(z) = \frac{A(z/0.92)}{1-0.68z^{-1}}$. A subscript of 0 denotes a zero-state filter, while a subscript of s denotes a filter with the state taken from the LPC synthesis filter at the previous subframe. All other filters are assumed to retain the memory of their input signals as pictured. The CELP framework has three main types of parameters: LPC parameters (here represented as LSFs), adaptive codebook parameters, and fixed excitation parameters. We consider an idealized CELP framework, in which the adaptive codebook consists of the true residual. This assumption is only accurate in cases were the bit-rate is large, enabling the decoded speech to be very accurate. The reason for employing this idealization is to separate speaker-dependent effects in the pitch period from the fixed excitation. That is, in cases where the adaptive codebook is very different from the true excitation, the adaptive codebook will not be able to account for all of the pitch effects, resulting in residual pitch information left over for the fixed codebook to handle. Since the goal of this chapter is to estimate the speaker-dependence of the various coder parameters, an idealized adaptive codebook is utilized to ensure that all pitch effects are accounted for by the adaptive codebook.

To carry out the experiments, a database of 45 speakers (23 male, 22 female) was gathered from the Wall Street Journal corpus. The recordings were made at the standard wideband speech sampling rate of 16 KHz, and consists of each of the speakers reading a common set of 140 sentences. The recordings were preprocessed using a typical wideband speech chain, split into frames of length 20 ms (320 samples), and 16-th order LSF coefficients for each frame were computed,

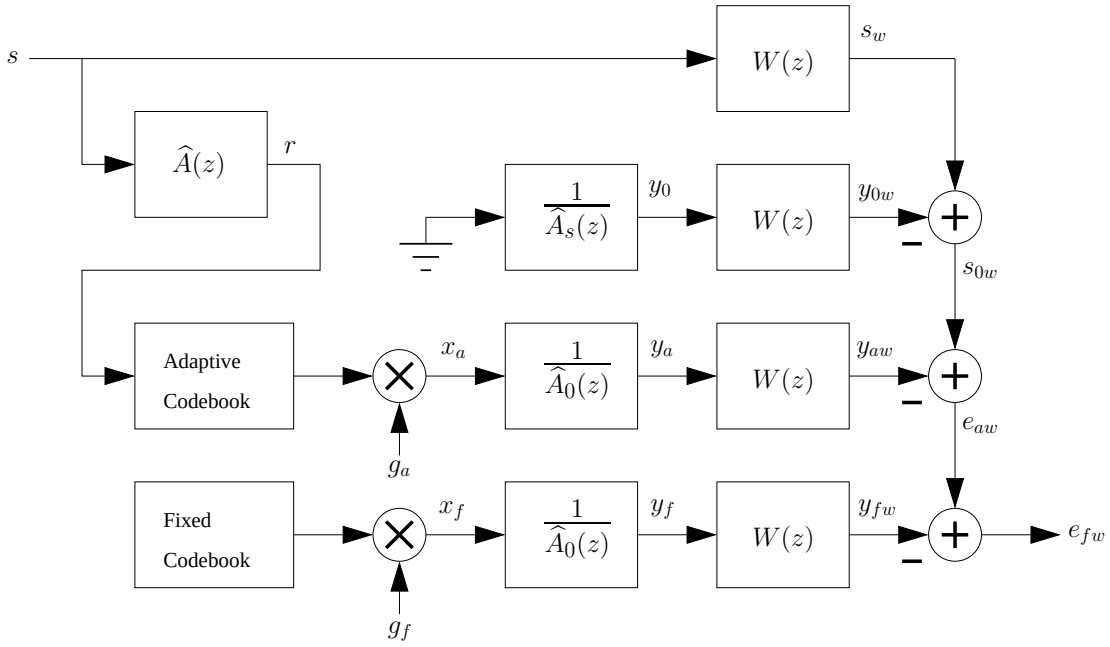

Figure 4.2: Idealized CELP coder. Note that the adaptive codebook contains the true residual signal. The subscript A_0 designates a zero-state filter, while the notation $\widehat{A}$ denotes a suitably interpolated synthesis filter.

using 10 ms of symmetric overlap with the adjoining frames. A modified Hamming window was utilized, emphasizing the last 5 ms of each frame, along with white noise regularization and 60 Hz bandwidth expansion. For each frame, the open-loop pitch lags were then computed using the method described in [47]. Each frame was then split into 4 subframes of 5 ms (80 samples), and interpolated LSFs were computed for each subframe. These interpolated LSFs specify the perceptual weighting filter as $W(z) = \frac{\widehat{A}(z/0.92)}{1-0.68z^{-1}}$, and are utilized in the closed-loop adaptive codebook search. Then, the closed-loop pitch lags and gains were computed for each subframe to minimize $||e_{aw}||$, with the constraint that only lags within 40 samples of the open-loop value are searched. The target signal for the fixed codebook, then, is e_{aw}, which represents the weighted error in a subframe after the zero-input response and adaptive codebook contributions have been removed. From each speaker's database, the last 5000 frames were set aside for testing purposes and the remainder were designated as training sets (sizes ranging from 25000 to 50000 frames in each training set, reflecting the different rates of speaking). Additionally, a speaker-independent training set was constructed using the first 20000 frames from each speaker's training set.

4.1.1 Gains from Speaker-Dependent Coding

The following three subsections discuss the gains available in speaker-dependent coding of the LSF parameters, adaptive codebook parameters, and fixed excitation, respectively. In the case of LSF quantization, a savings of 4 bits per frame results, or around 10% of the typical operating rate. During voiced frames, a savings of 1 bit per subframe is possible on pitch lag parameters, and much less during unvoiced frames. Moreover, the gains due to speaker-dependent coding of adaptive gains are found to be negligible, indicating that the adaptive codebook does not benefit significantly from speaker-dependent coding. Finally, gains of 10-15 bits per subframe result from speaker-dependent coding of the fixed excitation, amounting to a significant portion of the overall bit budget in a typical wideband

speech coder.

LSF quantization

The problem of wideband speech LSF quantization, with the Log Spectral Distortion measure, is considered first, and utilized later in the chapter to demonstrate other techniques relevant to speaker-dependent coding. This problem, and the similar narrowband case, have been studied extensively, and rates of 35-50 bits per frame are typical (c.f., [15], [12], [16], [17]). To quantify the impact of speaker dependence on LSF quantization, a GMVQ was designed for each speaker using the Weighted EM algorithm (see [12] and [18]), as well as a speaker-independent GMVQ, using the databases described above. The Weighted EM algorithm is able to incorporate the LSD measure through the use of a sensitivity matrix (see [20]) which represents the local second-order behavior of the distortion measure. The order of the GMVQs was $M = 16$, which is a common value in the literature.

Three trials were then performed: first, each speaker-dependent coder was applied to the corresponding speaker's test database. The results for each rate were then averaged over the speakers, giving an average performance curve for speaker-dependent coding. Next, the speaker-independent coder was applied to each speaker's test database, and the results again averaged as above, to indicate the average performance of a speaker-independent system. Finally, for each speaker's test database, 10 randomly selected speaker-dependent coders were applied, and the results were averaged over both the randomly selected coder and the speaker databases, indicating the average performance of a speaker-dependent system which is employing an incorrect speaker model. These results are summarized in Table 4.1. Notice that the speaker-dependent systems exhibit an improvement of 4 bits per frame in the sense of average distortion, and 5 bits in the sense of small outliers, relative to speaker-independent coding. Also note that the speaker error case shows a comparable disadvantage relative to speaker-independent coding, highlighting the need for speaker-identification and robustness against speaker

Table 4.1: Spectral Distortion Performance Around Operating Point

bits/frame	Speaker Dependent			Speaker Independent			Speaker Error		
	Avg. LSD (in dB)	Outliers (in %)		Avg. LSD (in dB)	Outliers (in %)		Avg. LSD (in dB)	Outliers (in %)	
		2-4dB	> 4dB		2-4dB	> 4dB		2-4dB	> 4dB
36	1.073	0.636	0	1.248	2.420	0.001	1.427	13.407	0.249
37	1.034	0.480	0	1.205	1.880	0.001	1.380	11.819	0.237
38	0.996	0.337	0	1.160	1.436	0.000	1.331	10.372	0.213
39	0.960	0.248	0	1.119	1.095	0	1.284	9.125	0.193
40	0.925	0.190	0	1.080	0.841	0	1.241	8.054	0.189
41	0.891	0.136	0	1.039	0.603	0.000	1.200	7.191	0.174
42	0.858	0.109	0	1.002	0.478	0	1.159	6.295	0.163
43	0.826	0.082	0	0.964	0.351	0.000	1.117	5.422	0.153
44	0.796	0.062	0	0.929	0.257	0.000	1.079	4.700	0.146
45	0.765	0.043	0	0.894	0.120	0	1.041	4.127	0.140

error.

Adaptive Codebook

Next, the issue of speaker-dependent coding of adaptive codebook parameters was considered. A typical wideband speech coder computes 4 pitch lags per frame (1 in each subframe), which in turn requires about 30 bits per frame for transmission. These parameters are usually not "coded" as such, but rather simply assigned a fixed-length binary index. Typical schemes alternate between an 8 bit index for odd-numbered subframes and a 7-bit index for even-numbered subframes, resulting in an average rate of 7.5 bits per pitch lag. In order to estimate the improvements offered by speaker-dependent coding, we will consider the entropy of the pitch-lag distribution. This gives the average length of an optimal variable-rate code applied to the pitch lag parameters. To this end, histograms showing the relative frequency of each lag value were produced for each speaker, as well as for the speaker-independent case. These results are illustrated in Figure 4.3. Note that, in voiced frames, the individual speakers exhibit an entropy roughly 0.85 bits less than the speaker-independent case. This amounts to a savings of around 3.5 bits per frame, roughly equal to the savings due to speaker-dependent coding of

the LSF parameters. In the unvoiced case, the difference is much smaller, with both speaker-dependent and -independent categories showing entropies of around 7 bits. Furthermore, applying the GMVQ framework to the quantization of the pitch gains under MSE (i.e., using the same methods as for the LSF quantization experiment), it was found that the gains for speaker-dependent coding of the pitch gains are negligible (only a fraction of a bit per frame). Overall, then, there may be little benefit to speaker-dependent coding of the adaptive codebook parameters, despite the evident variations in the pitch lag statistics in voiced frames.

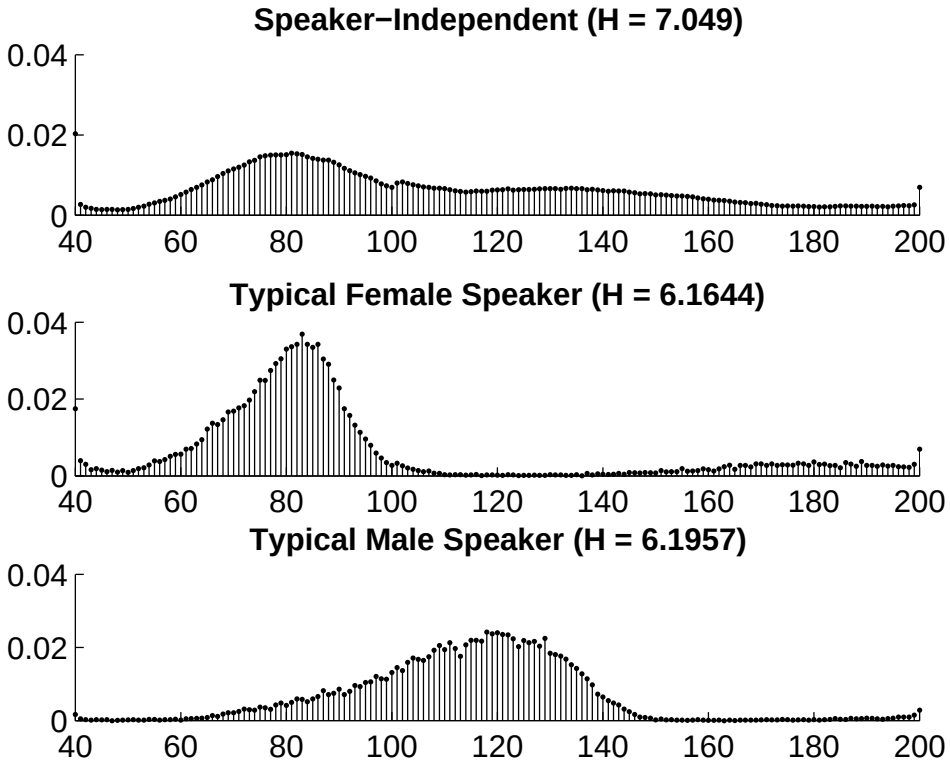

Figure 4.3: Statistics and Entropy of Pitch Lags in Voiced Frames

Fixed Codebook

Finally, we examine speaker-dependence in the fixed excitation. Although the fixed codebook is typically implemented in the residual domain (as pictured in Figure 4.2), the codebook search is normally conducted in the weighted signal

domain. That is, the goal of the fixed excitation search is to match the target signal e_{aw} in the sense of minimizing $||e_{fw}||$. For this reason, we consider the coding of the fixed excitation directly in the weighted signal domain. While this would be impractical for a real coder, because the resulting codevector would need to be run through the inverse of the weighting filter at the decoder (and also through the analysis filter in order to update the adaptive codebook), the fact that the search is carried out in the weighted signal domain and that the synthesis and perceptual weighting filters are invertible implies that the speaker-dependent performance gains should be the same. Thus, we dispense with the separation into a fixed residual codebook and gains, and instead consider the direct quantization of e_{aw} using the GMVQ framework. For this experiment, the appropriate quality measure is the Weighted Segmental Signal-to-Noise Ratio (WSSNR), $\mathrm{E}\left[\frac{||s_w||^2}{||e_{fw}||^2}\right]$, which reflects the masking effects of loud subframes. Note that the signal power of interest here corresponds to the (weighted) signal power before the adaptive codebook contribution has been removed. However, utilizing a segmental signal-to-noise ratio is problematic in the context of quantizer design, which is developed in terms of expected distortion measures. In the case of the usual (non-segmental) signal-to-noise ratio, this disparity is easily erased by the use of an inverse transformation

$$\arg\max_Q \frac{\mathrm{E}[||Y||^2]}{\mathrm{E}[||X-Q(X)||^2]} = \arg\min_Q \frac{\mathrm{E}[||X-Q(X)||^2]}{\mathrm{E}[||Y||^2]} = \arg\min_Q \mathrm{E}[||X-Q(X)||^2],$$

that is, by minimizing the noise power, conventional MSE-based quantization design methods also maximize the signal-to-noise ratio. However, this correspondence is broken in the case of segmental signal-to-noise ratio, as the fraction is now inside the expectation. Moreover, WSSNR is ill-posed for use in quantizer design, because the integrand has a singularity at every codepoint. For these reasons, we instead employ a weighted squared-error distortion measure $\mathrm{E}\left[\frac{||e_{fw}||^2}{||s_w||^2}\right]$, which reflects the principle that the distortion in a given subframe should scale with the

subframe power. This distortion measure can be related to WSSNR through the use of Jensen's inequality:

$$\mathrm{E}^{-1}\left[\frac{||s_w||^2}{||e_{fw}||^2}\right] \leq \mathrm{E}\left[\frac{||e_{fw}||^2}{||s_w||^2}\right], \tag{4.1}$$

The weighted squared error distortion measure can be incorporated into GMVQ design using the same weighted EM approach as in the case of LSD on LSF vectors. The only new element in this instance is the fact that the sensitivity matrix is parameterized by a side-information $||s_w||$ instead of deterministically varying with e_{aw}. Details of high-rate quantization for this case can be found in [54]; one particular modification here is that the expectations in Eq. (4.1) must be regarded as over both s_w and e_{aw}. Moreover, note that the sensitivity matrix in this case is a scalar, implying that an MSE encoder is still optimal. However, it is still necessary to redesign the codebook according to the distortion measure, which requires placing more of the codepoints in regions correlated with small values of $||s_w||^2$ (i.e., low signal power).

With this setup, an experiment similar to that performed for LSF coding was performed to quantify the speaker-dependent gains available in the fixed codebook. The results are pictured in Figure 4.4. A wide variety of coding rates were utilized, reflecting the large range of fixed codebook sizes used to operate at different rates. While the performance gains depend on the exact rate of operation, note that gains 10-15 bits per subframe, in terms of WSSNR, hold over most of the range of interest. This corresponds to a gain of 40-60 bits per frame or 2-3 kbps, a significant proportion of the typical wideband speech coding rate.

4.2 Exploiting Speaker-Dependence

There are a variety of ways in which the performance gains of speaker-dependent coding can be exploited. In the simplest case, one would simply reduce the operating rate, resulting in the same quality as a speaker-independent system

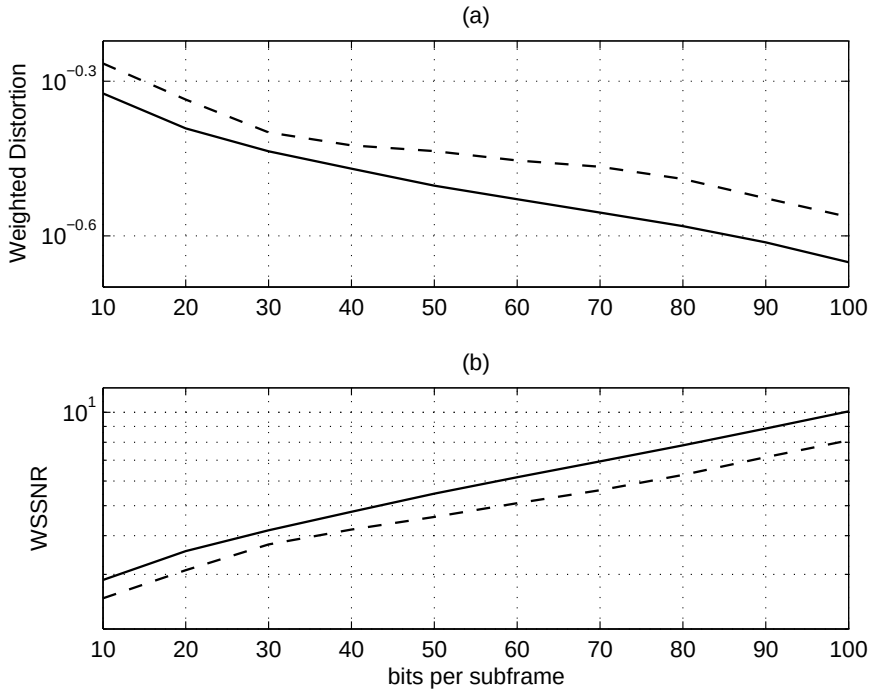

Figure 4.4: Performance of Fixed Excitation Quantization in terms of a) Weighted Distortion and b) Weighted Segmental Signal-to-Noise Ratio. The solid lines indicate speaker-dependent performance, while the dashed lines indicate speaker-independent systems.

on a reduced budget. The extra bits could then be assigned to other portions of the speech coder in order to improve quality, or the overall bitrate of the coder could simply be reduced. If speaker-dependent rates are allowed, more interesting schemes become possible. For example, one could achieve uniform quality over all speakers. That is, in speaker-independent coding, the rate is set such that the mean distortion and outliers (averaged over all speakers) meets an appropriate transparency criterion. However, it is the case that some speakers are much "harder" to code than others. This is illustrated in Figure 4.5, where it can be seen that the standard deviation of the mean operating rate for LSF coding is around 2 bits, in both the speaker-dependent and -independent cases. Thus, with

a single, speaker-independent setting of the operating rate, some speakers enjoy substantially better than transparent quality, while others suffer from much worse quality. Using speaker-dependent rates, bits can be shaved from "easy" speakers and dedicated to "difficult" ones, making the transmission quality more consistent.

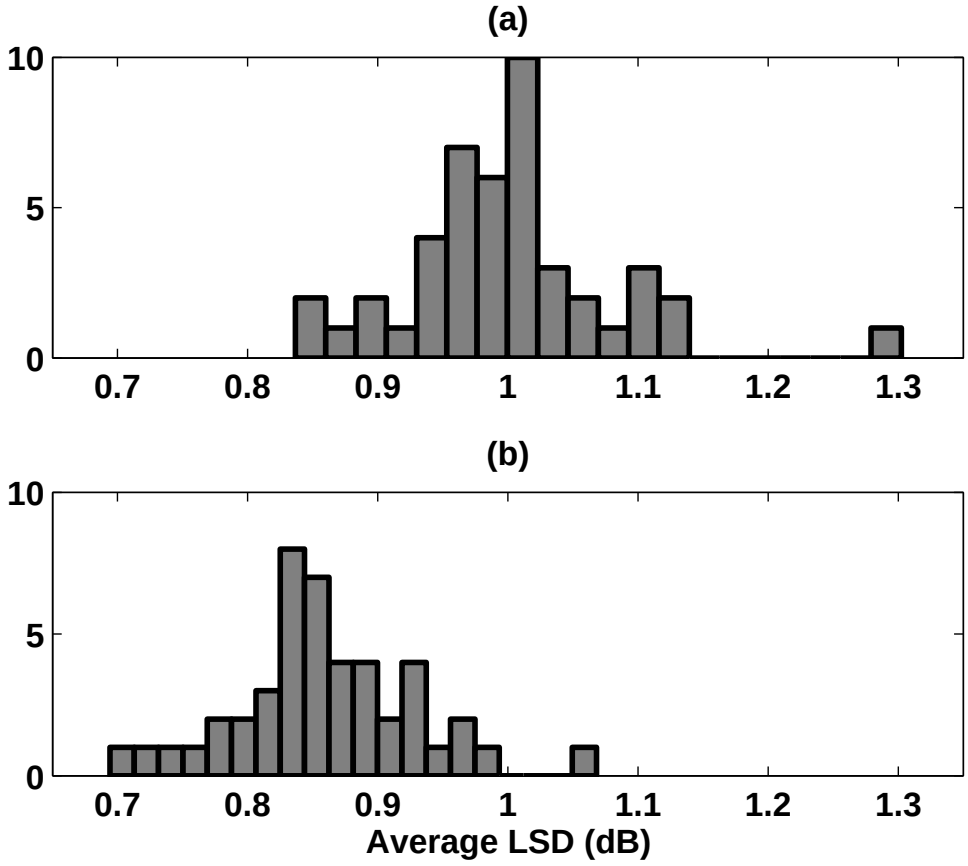

Figure 4.5: Variation of Average Distortion over Speakers. Plot (a) illustrates the speaker-independent case, while plot (b) shows speaker-dependent, both at a rate of 42 bits per frame. A change in average distortion of 0.04 dB corresponds to 1 bit per frame.

Speaker dependence can also be exploited to reduce complexity, rather than improve performance in the rate-distortion sense. That is, since speaker-dependent systems achieve better rate-distortion performance as speaker-independent systems of the same complexity, one would expect that they could also achieve comparable quality with a much lower complexity. To test this hypothesis, the LSF

coding experiments of the previous section were repeated for the lower model orders $M = 4$ and $M = 8$ (recall that the complexity of the GMVQ is proportional to M). Figure 4.6 illustrates the performance of speaker-dependent coding at a variety of complexities, and compares it to speaker-independent coding with $M = 16$. Notice that, even with $M = 4$, the speaker-dependent system still outperforms a speaker-independent coder with four times the complexity. Lastly, note that all of the methods for exploiting speaker-dependence can be combined in whichever proportions are deemed attractive.

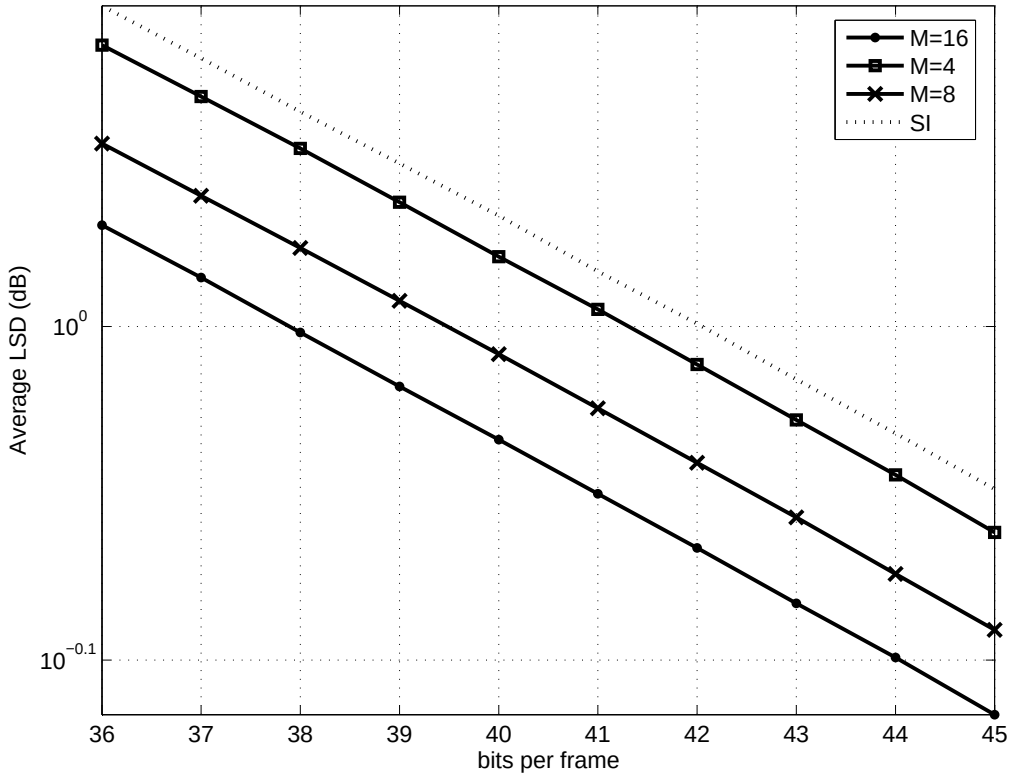

Figure 4.6: Illustration of Speaker-Dependent Performance at a Variety of Complexities

4.2.1 Safety-Net Systems

As illustrated in Table 4.1, a large penalty is incurred when an incorrect speaker-dependent model is employed. This represents a serious practical problem: for example, a user might hand his phone to a friend in the middle of conversation, causing a wild fluctuation in quality. While perfect speaker robustness is incompatible with exploiting speaker-dependent gains, the losses due to speaker errors can be limited by employing a safety-net approach, as in [17]. Safety-net quantizers ensure robustness by operating two coders in parallel, one of which is fixed. That is, speaker robustness can be obtained by operating a speaker-independent coder in parallel with the speaker-dependent coder, and then selecting the final codepoint by taking the better of the two outputs. It is necessary to reduce the rates of the two coders such that the overall number of codepoints corresponds to the desired rate. In the simplest case, this is accomplished by evenly dividing the codepoints between the two coders. This results in a simple index structure with a status bit indicating which component coder is active, and the remaining $r-1$ bits indexing the codeword from that coder. Such a system, then, should lose 1 bit of performance when operating on the correct speaker and, in return, ensure that the performance is within 1 bit of the speaker-independent case when operating on the wrong speaker.

In order to make a fair comparison between safety-net systems and unconstrained coders, it is also necessary to match the complexities of the two types of coders. In the context of GMVQ, this means that the total number of Gaussian coders employed should match in each case. So, for an unconstrained coder of order M, the safety-net system should be composed of, for example, a speaker-dependent GMVQ of order $M/2$ and a speaker-independent GMVQ of order $M/2$. Note that, due to the structure of the GMVQ (see Figure 4.1), the two component GMVQs of the safety-net system can then be combined into a single GMVQ of order M. That is, suppose that the parameters $\left\{\widehat{\alpha}_m, \widehat{\mu}_m, \widehat{\Sigma}_m\right\}_{m=1}^{M/2}$ represent a speaker-independent coder. These parameters can be trained ahead of time and

fixed. The speaker-independent parameters can then be combined with a set of speaker-dependent parameters $\{\alpha_m, \mu_m, \Sigma_m\}_{m=1}^{M/2}$ to construct a safety-net system of order M as follows:

$$\alpha_m^{\mathsf{sn}} = \begin{cases} \frac{1}{2}\widehat{\alpha}_m & , \quad 1 \leq m \leq \frac{M}{2} \\ \frac{1}{2}\alpha_{m-M/2} & , \quad \frac{M}{2}+1 \leq m \leq M \end{cases} \tag{4.2}$$

$$\mu_m^{\mathsf{sn}} = \begin{cases} \widehat{\mu}_m & , \quad 1 \leq m \leq \frac{M}{2} \\ \mu_{m-M/2} & , \quad \frac{M}{2}+1 \leq m \leq M \end{cases} \tag{4.3}$$

$$\Sigma_m^{\mathsf{sn}} = \begin{cases} \widehat{\Sigma}_m & , \quad 1 \leq m \leq \frac{M}{2} \\ \Sigma_{m-M/2} & , \quad \frac{M}{2}+1 \leq m \leq M \end{cases} \tag{4.4}$$

The factor of $\frac{1}{2}$ in Equation (4.2) is introduced in order to assign approximately half of the codepoints to the speaker-independent portion. Notice that, if it desired to emphasize the speaker-independent portion of the safety-net system over the speaker-dependent portion, this factor could be increased to assign more bits. Likewise, one could dedicate a larger share of the M total Gaussian coders to the speaker-independent portion to achieve a similar effect.

The next question is how the speaker-dependent portion of the safety-net system should be trained. The simplest method is to utilize the same speaker-dependent designs from the previous section, which were designed without reference to the safety-net system. However, this approach sacrifices performance on the correct speaker in exchange for better performance under speaker errors. The reason for this is that, since the speaker-dependent portion of the coder is trained without reference to the safety net, it expends a portion of its modeling power on features that are common to all speakers. While this is required for an unconstrained speaker-dependent system, this effort is wasted in the safety-net context, since the speaker-independent portion of the coder already models the common features. In order to achieve the desired 1 bit penalty/robustness, the training should be carried out using the entire safety-net GMVQ, which will allow the speaker-dependent portion to focus its efforts on those speaker-specific features

not captured by the speaker-independent subcoder. Supposing one already has the speaker-independent parameters, this can then be accomplished by applying a slight variation on the EM algorithm to a single-speaker database. The E-step is unchanged, which is to say it consists of computing $r_{mn} = \alpha_m^{\mathsf{sn}} N(x_n|\mu_m^{\mathsf{sn}}, \Sigma_m^{\mathsf{sn}})$ for $m = \{1, \ldots, M\}$ and all time steps n, and then normalizing appropriately. Note that this step uses the entire set of M parameters, with no distinction between the speaker-dependent and speaker-independent portions. The M-step, however, must be modified in order to leave the speaker-independent portion of the parameters unchanged. To accomplish this, the regular M-step updates are applied only to the last $M/2$ parameters, which correspond to the speaker-dependent portion. Lastly, the speaker-dependent mixture weights must be normalized to sum to $\frac{1}{2}$, as in Eq. (4.2), in order to meet the constraint.

To demonstrate the performance of safety-net systems, an experiment similar to the previous one was performed. A safety-net GMVQ of order $M = 16$ was trained for each speaker in the database. Then, for each speaker's test database, the correct safety-net coder was applied, followed by 10 randomly selected incorrect coders. The results were then averaged as above to show the average performance of safety-net GMVQ, which is summarized in Table 4.2. Comparing with Table 4.1, it is clear that the performance of safety-net systems under speaker error is 1 bit behind speaker-independent coding, and the performance with the correct speaker is 1 bit behind unconstrained speaker-dependent coding, as expected. The penalty for a speaker error has been cut from 8 bits in the unconstrained case to 4 bits in the safety-net case, as desired.

4.3 On-line Training

Because it is impractical to compile training databases of individual speakers prior to deployment, speaker-dependent systems must be designed in an on-line fashion. This introduces two new considerations into the system de-

Table 4.2: Spectral Distortion Performance of Safety-Net Systems Around Operating Point

bits/frame	Correct Speaker			Speaker Error		
	Avg. LSD (in dB)	Outliers (in %)		Avg. LSD (in dB)	Outliers (in %)	
		2-4 dB	> 4 dB		2-4 dB	> 4 dB
36	1.117	0.741	0	1.289	3.961	0.001
37	1.078	0.543	0	1.246	2.999	0.001
38	1.039	0.384	0	1.200	2.274	0.000
39	1.001	0.295	0	1.156	1.748	0.001
40	0.965	0.208	0	1.116	1.334	0.001
41	0.930	0.152	0	1.076	1.028	0.000
42	0.895	0.108	0	1.037	0.790	0.001
43	0.862	0.083	0	0.999	0.611	0.000
44	0.830	0.066	0	0.962	0.456	0.000
45	0.798	0.040	0	0.924	0.340	0.000

sign: computational resources are required for carrying out the training algorithms, and communications resources are required in order to disseminate the resulting speaker-dependent designs to the relevant end-users. This section considers a variety of different training configurations, each of which strikes a different balance between computational requirements, communications requirements, and performance. One feature shared by all of the schemes is that the system initially operates in a speaker-independent mode. Then, as suitable training data becomes available, speaker-dependent designs are produced and disseminated, allowing a transition to speaker-dependent operation. The schemes differ in how they balance end-user resources, communications resources, and how much of the performance advantages described in Section 4.1 they are able to realize. Three different training configurations are presented in Section 4.3.1, and their relative strengths and weaknesses are discussed. Section 4.3.2 discusses learning using quantized data, which is required to enable training configurations with remote learning. A modification to the GMVQ decoder is presented that avoids singularities in the training process, and the performance losses due to training on quantized data are experimentally quantified. Lastly, Section 4.3.3 discusses recursive learning, wherein the training is conducted in a sample-by-sample manner, eliminating the need to store large training databases. Recursive learning can be applied to any of the training configurations in order to reduce storage requirements, or to enable adaptive operation.

4.3.1 Training Configurations

Consider first the most straightforward training scheme, illustrated in Figure 4.7. In this Local Learning approach, each end user's equipment first stores up a suitably large database of unquantized training data. Then, the standard training schemes described in Section 4.1 are applied to produce the speaker-dependent coder design. Finally, the design is transmitted to the required receivers as explicit side information (either directly or via a centralized database). The primary advantage to this approach is that training is carried out on unquantized data,

and so achieves the full measure of performance improvement. Local Learning has a number of drawbacks: it requires each end-user's equipment to have sufficient memory and processing power for the training process (which is generally much larger than the coding complexity). It also requires the resulting coder parameters to be sent explicitly as side-information. As discussed in the introduction, the number of scalar parameters for a GMVQ of order M, operating in dimension d, is $M(1 + d + d(d + 1)/2) - 1$. Supposing these parameters are represented with 16-bit fixed-point words, the overhead for transmitting a speaker-dependent LSF quantizer with $M = 16$ and $d = 16$ works out to 38,656 bits, corresponding to a few seconds worth of wideband speech at typical coding rates. For a speaker-dependent excitation codebook, with $d = 80$ and $M = 16$, the resulting overhead is 850,160 bits, corresponding to a little more than one minute of wideband speech data. Note that this overhead is incurred multiple times for each user, in order to distribute it to the other users he wishes to communicate with. Also note that, if a safety-net coder is employed, the overhead for speaker-dependent parameters is reduced by one half. Furthermore, the training and distribution of speaker-dependent coders need not take place during an actual phone call. For example, the system could train the speaker-dependent coder and then transmit it to the entries in the user's address book during times of inactivity.

In situations where the end user's equipment is not powerful enough to implement the learning process locally, the design can instead be carried out at remote locations, as illustrated in Figure 4.8. To enable Remote Learning, training must be performed on data that has already been quantized (presumably by a speaker-independent system). Details of learning on quantized data are presented in Section 4.3.2. The primary advantage of remote learning is that it eases the complexity requirements for the end users. The only additional complexity required in the end user's equipment is the ability to receive and store new coder parameters and load the resulting systems. In exchange for relocating the training complexity, a performance loss is incurred due to the use of quantized data in the training phase.

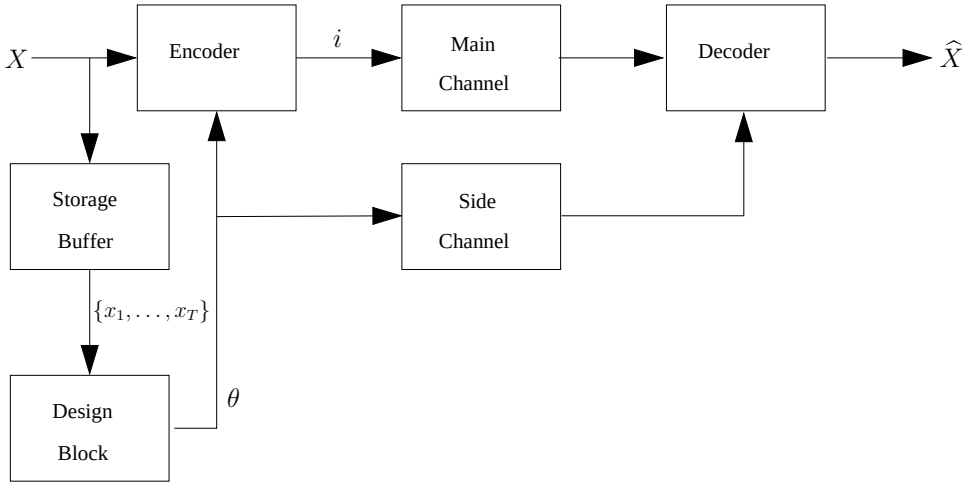

Figure 4.7: Local Learning. This configuration avoids performance penalties associated with training on quantized data.

Additionally, one extra side information transmission is necessary to communicate the speaker-dependent coder design back to the encoder.

In scenarios where the transmission overhead for distributing the speaker-dependent models is the limiting factor, Synchronized Learning can be used (see Figure 4.9). In this method, both the encoder and decoder perform the learning process in parallel. This allows both ends of the communications system to update their coders in a synchronized manner, without sending any side-information. In order to maintain synchronization between the two ends, the encoder must perform the learning process on quantized data. This method requires every end user to have sufficient computational power and storage to implement the training process. Also, note that the cost of adding a decoder to the transmitter side is not onerous in the case of GMVQ, which computes the output vectors in the encoding process. Another disadvantage is that the training process would have to be repeated every time a new user is contacted. Thus, this method represents the other extreme of the complexity/overhead trade-off: large, redundant training complexity in exchange for zero transmission overhead. It should be noted, however, that such an approach is sensitive to transmission errors, which can cause the learning processes to lose

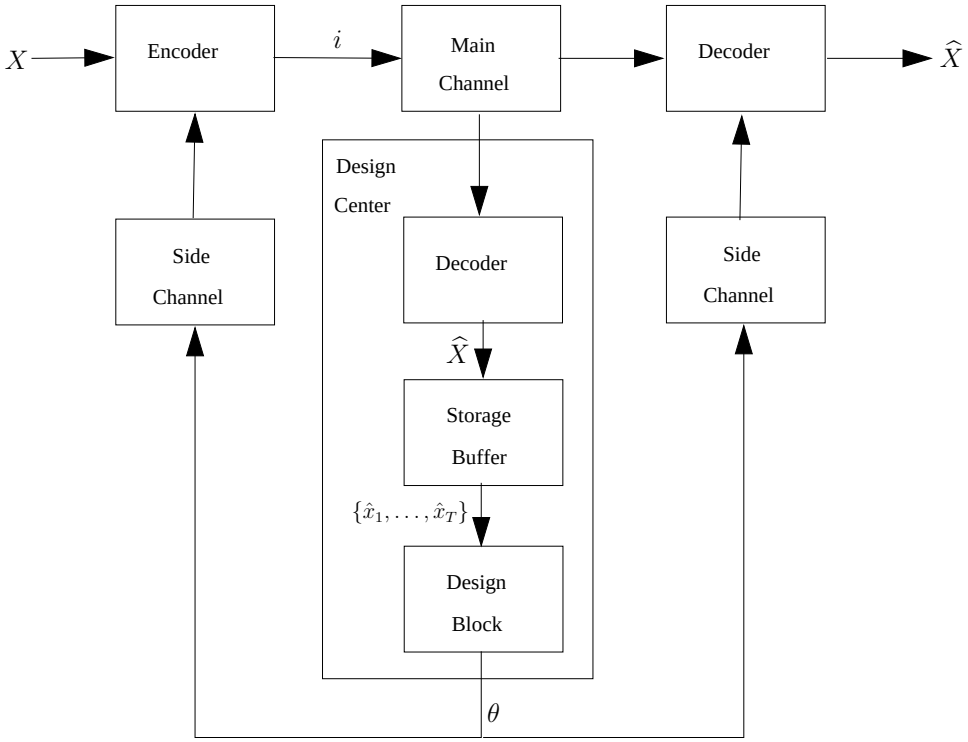

Figure 4.8: Remote Learning. This configuration minimizes the required end-user complexity.

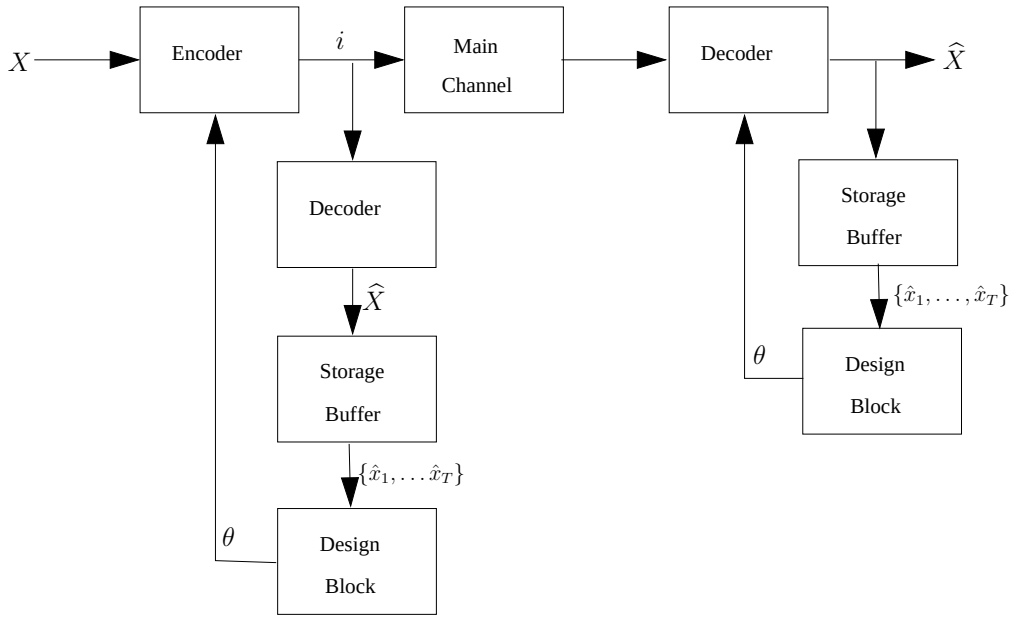

Figure 4.9: Synchronized Learning. This configuration avoids the use of side information.

synchronization. In order to overcome this, it will still be necessary, in practice, to transmit some side-information to guard against loss of synchronization.

4.3.2 Learning from Quantized Data

A number of issues arise when considering learning from quantized data. First, since the learning process does not have access to clean data, some degradation in the resulting design is expected. This loss, as a function of encoding rate, is quantified later in this subsection. Another issue that arises in the context of GMVQ results from the nature of the quantization error (as opposed to its magnitude as such). To see this, recall that the GMVQ utilizes scalar transform coders to implement each component Gaussian coder. The covariance matrices of the individual Gaussians tend, in practice, to be fairly oblong. This results in transform codebooks that, at standard operating rates, lie in subspaces of $\mathbb{R}^d$. That is, only a single codepoint is allocated to the least significant transform dimension(s), as

Table 4.3: Level Allocation for Speaker-Independent GMVQ with $M = 16$ at a rate of 43 bits per frame (transparent quality). Note that all of the component transform coders assign a single level to the least significant transform component.

Transform Coefficient \ Cluster Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	2	2	2	3	1	2	2	2	2	2	2	2	2	2	1
3	3	2	3	3	3	3	2	3	2	2	3	2	4	4	4	3
4	4	4	4	4	3	4	4	3	4	4	4	3	4	4	4	3
5	4	4	5	4	4	4	4	4	4	4	4	4	4	4	5	4
6	4	5	5	4	4	5	4	4	4	5	4	4	5	4	5	4
7	5	5	5	5	4	5	4	5	5	6	4	5	5	5	5	5
8	6	5	5	5	4	6	5	5	5	6	4	6	5	5	6	6
9	7	6	5	5	5	6	6	6	6	7	5	6	5	5	6	6
10	7	7	5	6	5	7	7	6	7	7	6	7	5	6	6	7
11	8	8	6	6	6	7	7	7	7	7	7	8	6	7	6	8
12	10	9	6	6	8	9	8	8	7	8	8	9	6	7	8	10
13	10	10	9	8	9	10	9	10	9	9	10	12	7	9	10	10
14	11	11	11	11	11	11	16	13	10	12	12	13	8	11	10	12
15	15	16	18	14	17	15	17	16	16	13	14	15	10	13	12	16
16	17	16	23	32	37	17	17	16	21	17	31	17	19	16	14	19

illustrated in Table 4.3. Note that the entire GMVQ codebook does not lie in a subspace, as the subspaces of each component Gaussian coder do not typically coincide. Nevertheless, attempting to learn a GMVQ of comparable order as was used to quantize the data results in each component "locking on" to a corresponding subspace. This leads to a numerical instability wherein the covariances shrink without bound, derailing the learning process.

To circumvent this problem, a postprocessing can be applied to the quantized data before it is utilized in the learning process. This postprocessing consists of adding Gaussian noise to the quantized data in order to ensure that the data has full rank. Specifically, noise is added only to those coefficients of the

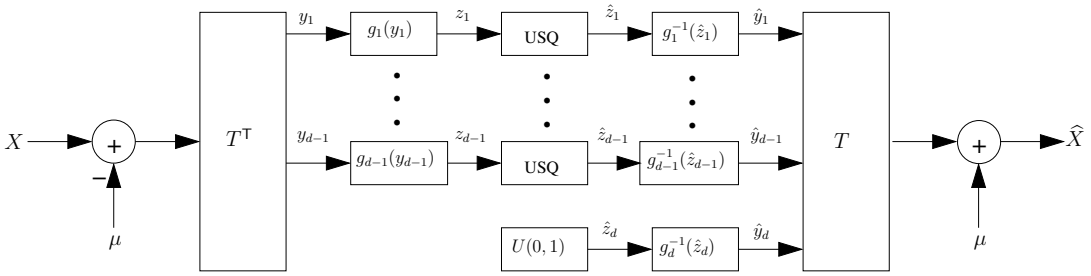

Figure 4.10: Transform coder with decoder modified for use in learning. In this example, the d-th transform component received an allocation of 0 bits; all other components are coded as normal. The functions g_i are the compressor functions (i.e., cdf's of Gaussians), the "USQ" blocks are uniform scalar quantizers (on $[0, 1]$) and the $U(0, 1)$ block is a random number generator, uniformly distributed on $[0, 1]$.

(transformed) quantized vector which are degenerate (i.e., the dimensions which received an allocation of 1 codepoint). The encoder-decoder pair for a transform coder with postprocessing is illustrated in Figure 4.10. The variance of the noise for each coefficient is set according to the GMVQ used to quantize it. Note that this postprocessing should not be applied to the actual output of the quantizer in the operational speech coder, as it amounts to adding extra noise: it is only intended to be applied to data for use in the learning process.

In order to demonstrate the effectiveness of the postprocessing scheme, and to quantify the loss due to learning on quantized data, a set of experiments on LSF quantization were performed. First, the training set for each speaker was quantized using a speaker-independent GMVQ of order 16 (trained on clean data). The data was quantized at a rate of 43 bits per frame, resulting in transparent quality (see Table 4.1). Postprocessing as shown in Figure 4.10 was employed to ensure that the quantized data was full rank (every cluster had at least one dimension with an allocation of 0 bits). For each speaker, then, two coders were trained: an unconstrained speaker-dependent coder, and a safety-net coder, both of order $M = 16$. To assess the performance of the models from quantized data, all

three coders (speaker-independent, speaker-dependent and safety-net) were then operated on each speaker's test set. Additionally, 10 randomly selected "incorrect" safety-net coders were operated on each test set in order to examine the performance under speaker error. The results are illustrated in Figure 4.11. Note that, as before, the safety-net system imposes a 1-bit penalty on speaker-dependent performance and in turn ensures that the performance under speaker error is limited to 1 bit worse than the speaker-independent case. Comparing with the previous results using unquantized training data (i.e., Tables 4.1 & 4.2), it becomes apparent that the price of learning from quantized data is about 1 bit. That is, the performance of the speaker-dependent and safety-net systems have worsened by a margin of 1 bit. Notice, however, that the performance of the speaker-independent system, and the safety-net system under a speaker error, have not changed. This is because the speaker-independent system was trained on unquantized data, and, because the safety-net system is built around the speaker-independent parameters, it inherits this performance advantage.

Next, in order to characterize the effects of the encoding rate upon the learning process, the previous experiment was repeated using a wide variety of quantization rates. The results are illustrated in Figure 4.12. Note that the operating point for transparent quality (around 40 bits) lies in a steep section of the performance curve, which is to say that large gains in the performance of speaker-dependent systems can be obtained by increasing the bit rate during training. As the training rate approaches 70-80 bits per frame, the slope levels off. This reflects the fact that such rates are sufficiently large as to make postprocessing unnecessary (i.e., all components of all clusters are allocated at least 1 bit). On the other hand, at very low rates, only a small improvement is possible. In this regime, many components of every cluster receive allocations of 0 bits, and so postprocessing is applied to a large proportion of the components. Since the postprocessing is based on the speaker-independent model, it imposes, to some degree, the speaker-independent statistics onto the training data, resulting in performance

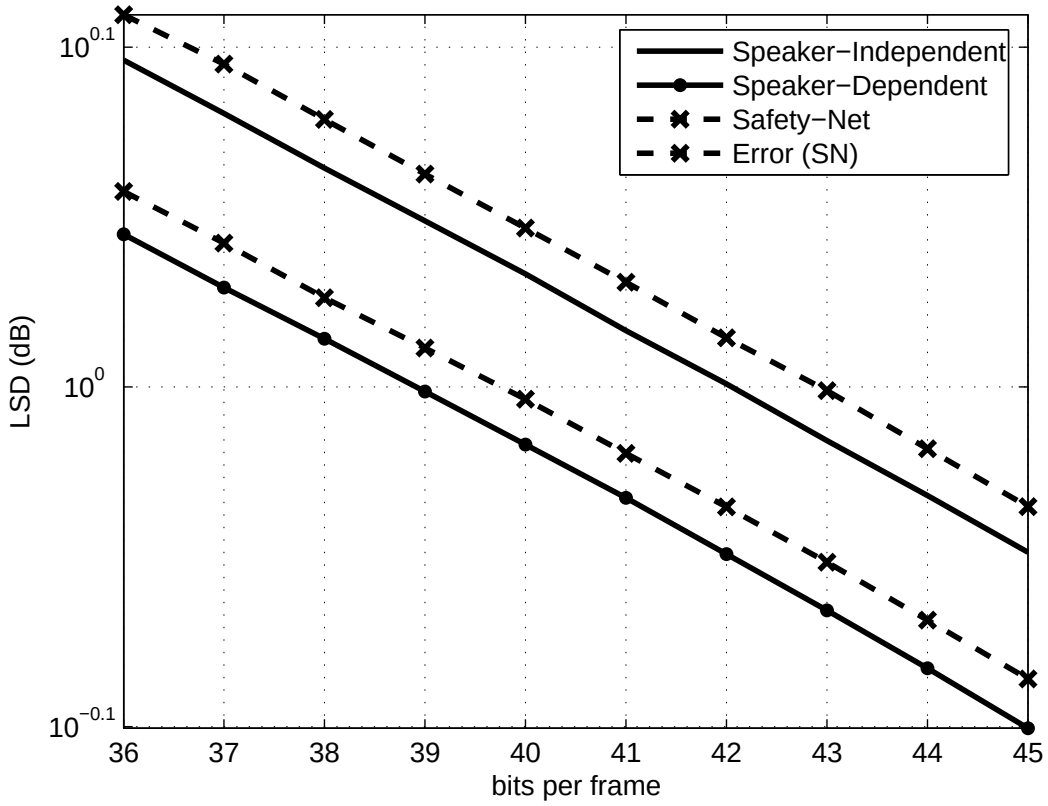

Figure 4.11: Performance of Speaker-Dependent and -Independent LSF Quantization When Learning on Quantized Data.

very close to the speaker-independent case. Moreover, this curve suggests that it may be beneficial to boost the quantization rate during training, in order to avoid performance penalties. If the rate at which LSFs are coded can be doubled for a time, models trained on quantized data will show very little loss. If desired, this temporary increase in rate could be accomplished without raising the overall rate of the coder by rededicating bits from the fixed codebook during the LSF learning phase. A similar reshuffling could then be applied to allow learning of the other parameters. While this would result in degraded audio quality during the training phase, it would also result in a much better speaker-dependent model once training was completed (at which time, the normal bit allocation could be restored).

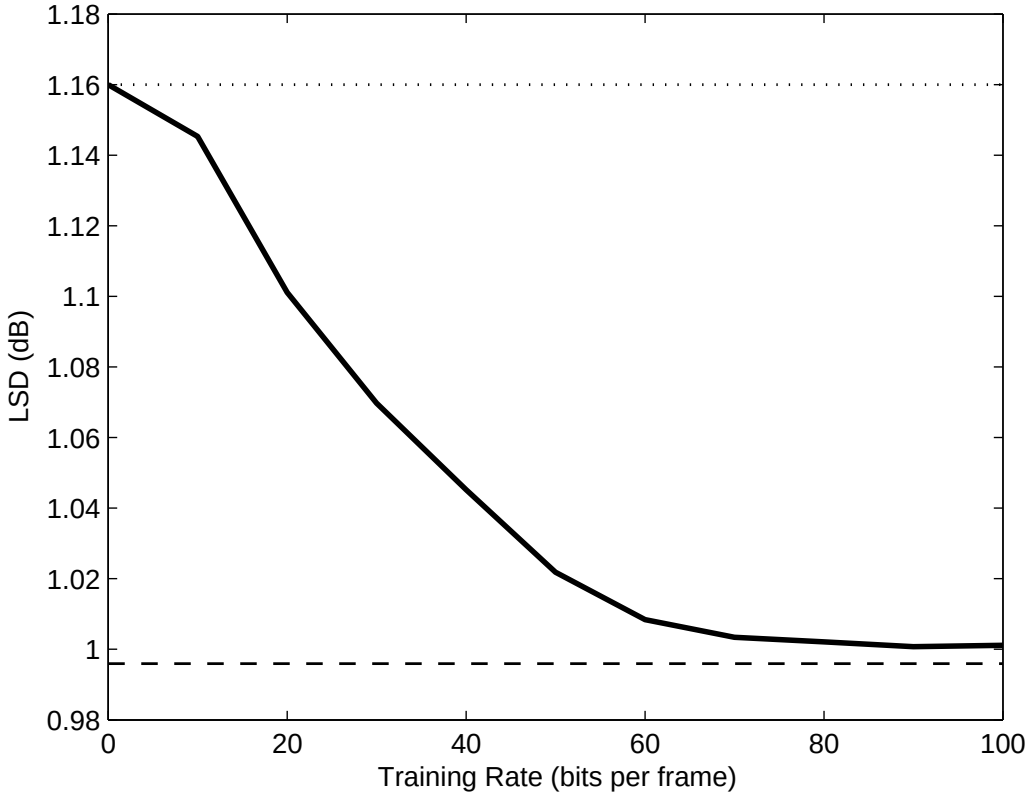

Figure 4.12: Performance of Speaker-Dependent LSF Quantizers Trained on Quantized Data, as a function of bit rate. The dashed line shows the performance of speaker-dependent systems trained on clean data, while the dotted line shows speaker-independent performance. All of the systems illustrated operated at a rate of 38 bits per frame.

4.3.3 Recursive Learning

All of the training strategies discussed above have assumed that the training process, wherever it is carried out, has access to a suitably large database of training data. That is, the systems operate in a speaker-independent mode until they have amassed enough user-specific training data to perform the training process. The design is then carried out in a single batch, resulting in the desired speaker-dependent coders. However, it may be that the requirement of storing a

speaker-dependent database is prohibitive, particularly when the learning is carried out in the end-user's equipment. In these cases, recursive learning can be employed, wherein the learning process utilizes only a single frame at a time, resulting in a sequence of parameter estimates. Such a scheme has very low storage requirements, needing only the current frame's data, the parameters, and a few auxiliary variables used in the recursion. It can also provide for adaptive operation. In such a case, the training process would continue indefinitely, allowing the coders to track changes in the speaker or acoustic environment. Note that adaptive methods incur repeated communications overhead costs in schemes using explicit coder transmission, and so may be more suited to synchronized learning approaches.

In the context of learning a GMVQ, the Recursive EM algorithm can be employed for this purpose. This algorithm, presented by Titterington in [50] and [51], is based on Stochastic Approximation. That is to say, the parameter estimate at time step n takes the form:

$$\theta_n = \theta_{n-1} + \eta(n) T_n^{-1} \left[\frac{\partial}{\partial \theta} \log f_\theta(x_n) \right]_{\theta=\theta_{n-1}} \tag{4.5}$$

where $\eta(n)$ is a step-size parameter and T_n is a conditioning matrix. Given certain technical conditions, stochastic approximation theory guarantees that such an estimator is consistent. In particular, it is required that $\sum_n \eta(n) = \infty$ and $\sum_n \eta^2(n) = 0$, i.e., $\eta(n) = o(\frac{1}{n})$. Beyond its effects on consistency, the choice of $T(n)$ determines the relative asymptotic efficiency of the estimation procedure, with optimal performance achieved by employing the Fischer Information matrix (i.e., the Hessian of the log-likelihood). However, in nontrivial problems such as estimation of the parameters of a multivariate GMM, it is very expensive to compute and, particularly, invert the Fischer Information. A popular alternative, then, is to use the "complete-data" Fischer Information matrix, which is much simpler to compute and invert, although it results in decreased relative efficiency. This is referred to as the Recursive EM Algorithm, and results in the following

update procedure for the case of GMM:

$$
\begin{aligned}
r_{mn} &\propto \alpha_{m(n-1)} N(x_n|\mu_{m(n-1)}, \Sigma_{m(n-1)}) && (4.6)\\
\alpha_{mn} &= (1-\eta(n))\alpha_{m(n-1)} + \eta(n) r_{mn} && (4.7)\\
&= \tau_{mn} + \rho_{mn} && (4.8)\\
\mu_{mn} &= \frac{\tau_{mn}\mu_{m(n-1)} + \rho_{mn} x_n}{\tau_{mn} + \rho_{mn}} && (4.9)\\
\Sigma_{mn} &= \frac{\tau_{mn}\Sigma_{m(n-1)} + \frac{\tau_{mn}\rho_{mn}}{\tau_{mn}+\rho_{mn}} \left\langle x_n - \mu_{m(n-1)} \right\rangle}{\tau_{mn} + \rho_{mn}} && (4.10)
\end{aligned}
$$

where $\langle . \rangle$ denotes the outer product and τ_{mn} can be thought of as the "prior strength" assigned to the m-th old estimate (as of time $n-1$) and ρ_{mn} represents the new information for cluster m at time n. Notice the similarity between these recursions and the expressions that arise in Sequential MAP estimation of a GMM with a complete-data conjugate prior (see [48]). In the case that $\eta(n) = \frac{1}{n}$, then, the two approaches are equivalent. However, as will be seen shortly, other choices of $\eta(n)$ are more appropriate to the recursive learning problem, in which case the equivalence with MAP estimation does not apply. Also note that the inverse and determinant of Σ_{mn} will also be required in order to compute r_{mn} at the each time step. Because the update to Σ_{mn} in Eq. (4.10) is a rank-one update, these quantities can be efficiently computed in a recursive manner by applying the Matrix Inversion Lemma:

$$
\begin{aligned}
\Sigma_{mn}^{-1} &= \frac{\tau_{mn} + \rho_{mn}}{\tau_{mn}} \left(\Sigma_{m(n-1)}^{-1} \left(I - \frac{\frac{\rho_{mn}}{\tau_{mn}+\rho_{mn}} \langle x_n - \mu_{m(n-1)} \rangle \Sigma_{m(n-1)}^{-1}}{1 + \frac{\rho_{mn}}{\tau_{mn}+\rho_{mn}} \|x_n - \mu_{m(n-1)}\|^2_{\Sigma^{-1}_{m(n-1)}}} \right) \right) && (4.11)\\
|\Sigma_{mn}| &= \left(\frac{\rho_{mn}}{\tau_{mn} + \rho_{mn}} \right)^d \left(1 + \frac{\rho_{mn}}{\tau_{mn} + \rho_{mn}} \|x_n - \mu_{m(n-1)}\|^2_{\Sigma^{-1}_{m(n-1)}} \right) \left| \Sigma_{m(n-1)} \right| && (4.12)
\end{aligned}
$$

The last issue to determine is the schedule of the stepsize $\eta(n)$. The simplest approach is simply to utilize $\eta(n) = \frac{1}{n}$, as in [50]. This approach is useful

in, for example, adaptive settings wherein a good estimate is already available, and it is desired to update it using new data (c.f. [52]). In such a scenario, the "prior weight" given to the initial estimate is quite high, and a small stepsize is desired. Thus, a $\frac{1}{n}$ schedule, starting at some fairly large n_0 is appropriate. However, in "from scratch" estimation problems, the simple $\frac{1}{n}$ schedule suffers from its sensitivity to early data. That is, during the early stages of estimation, when the old parameter estimate is very inaccurate, employing a $\frac{1}{n}$ schedule typically causes the estimation procedure to diverge. To avoid this problem, one can employ a modified learning schedule as suggested by Sato in [53]:

$$\eta(n) = \left(\sum_{t=1}^{n} \prod_{s=t+1}^{n} \lambda(s) \right)^{-1} \tag{4.13}$$

$$\lambda(n) = 1 - \frac{1}{(n-2)\gamma + \frac{1}{\epsilon_0}} \tag{4.14}$$

where $\lambda(n)$ is a "forgetting factor," whose schedule is parameterized by γ, which controls the asymptotic decay rate of $\eta(n)$, and ϵ_0, which sets the initial length of the "memory window." Notice that $\eta(n)$ can be computed recursively:

$$\eta(n) = \frac{1}{1 + \frac{\lambda(n)}{\eta(n-1)}} \tag{4.15}$$

Thus, in this approach, the stepsize schedule is separated into three regions. In the range $1 < n < \frac{1}{\epsilon_0}$, a rough (but reliable) estimator is formed using a short memory window. In the second phase, $\frac{1}{\epsilon_0} < n < \frac{1}{\gamma\epsilon_0}$, learning is carried out with a window length of $\frac{1}{\epsilon_0}$ (i.e., $\eta(n) \approx \frac{1}{\epsilon_0}$). In the final phase, $\eta(n)$ decays as $\frac{\gamma+1}{\gamma n}$, ensuring that that the estimator meets the consistency requirements set forth by stochastic approximation.

In our experiments on learning of speaker-dependent GMVQs for LSF quantization, we found that the values $\gamma = 0.05$ and $\epsilon_0 = 0.001$ resulted in reliable estimation. The learning curves for this problem, averaged over all 45 speakers, are seen in Figures 4.13. Note the small disparity between the likelihood performance

and the actual quantization performance. This results from the assumption that the inertial profile of the quantizers is independent of the parameters (see Chapter 3), which is not really accurate. This is not a major impediment, in that the variation in the inertial profile is small compared to the changes in likelihood that are achieved.

4.4 Discussion

This chapter considered the problem of speaker-dependent wideband speech coding. A simplified CELP framework was considered, which has three types of parameters: spectrum parameters (here, LSFs), adaptive codebook parameters, and fixed codebook parameters. In order to quantify the speaker-dependent gains in each type of parameter, the GMVQ framework was utilized, which is able to represent the statistics of individual speakers. First, it was found that gains of 4 bits per frame can be realized in the case of spectrum coding under LSD. Next, the adaptive codebook parameters were considered. While statistical variation between speakers was evident in the pitch lags in voiced frames, the corresponding speaker-dependent gains were still modest. Moreover, it was found that the gains for pitch lags were negligible in unvoiced frames, and that the pitch gains showed insignificant gains in all types of frames. Thus, there is little to be gained from speaker-dependent coding of adaptive codebook parameters. Finally, the coding of the fixed excitation was considered. In order to quantify speaker-dependent gains, quantization of the fixed excitation was considered in the weighted signal domain, where the search over fixed codebook parameters normally takes place. A weighted squared-error measure was used, which corresponds to WSSNR. It was shown that gains of 10-15 bits per subframe are achievable using speaker-dependent quantizers, corresponding to a significant portion of a typical wideband speech coder's bit budget. These savings can be leveraged in a variety of ways, for example by reducing complexity rather than operating rate. Also, it was shown that significant

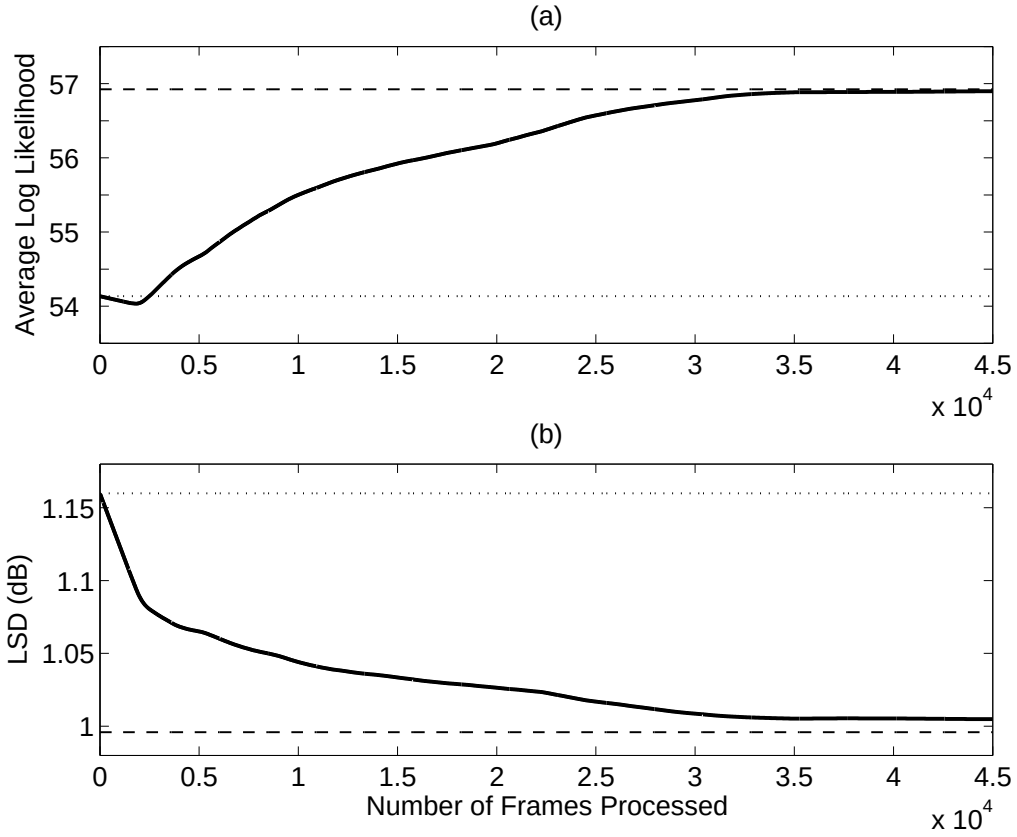

Figure 4.13: Convergence of Online EM Algorithm. Subfigure (a) shows the average log likelihood while subfigure (b) depicts the average Log Spectral Distortion. The dashed lines show the performance of speaker-dependent systems trained using batch methods, while the dotted lines show speaker-independent performance. Note that the average change in log likelihood is negative at the beginning of the estimation process, while the average distortion monotonically decreases. Also note that the final performance achieved by the recursive procedure within 1/4 of a bit per frame of the batch learning performance.

variations exist in how "difficult" it is to code a particular speaker, implying that speaker-dependent settings of the coding rates can result in more uniform quality.

Next, a variety of techniques needed for operation of a speaker-dependent coding system were presented. First, safety-net systems were considered, which

attain robustness against incorrect speakers by operating a speaker-independent quantizer in parallel with the speaker-dependent one. This approach can be incorporated naturally into the GMVQ framework, which itself functions by choosing between the outputs of candidate coders. In order to allow precise trade-off between robustness and performance, a slightly modified EM algorithm was presented, which includes the effects of the speaker-independent coder.

Next, a variety of online learning architectures were presented, which strike different balances between complexity, communications costs and performance. In Local Learning, the training is carried out at the encoder side using unquantized data, and then the resulting speaker-dependent coder design is transmitted as side information to the decoder. This configuration requires the end user's equipment to have sufficient resources to carry out training, but, because clean data is used, suffers no performance penalty. Next, Remote Learning was considered, wherein the learning is carried out at a remote location (for example, a base station). This removes the onus of training complexity from the end-user, but incurs a performance penalty because training must be carried out on quantized data. Finally, Synchronized Learning was considered, wherein both the encoder and decoder carry out the learning process in tandem, obviating the requirement to transmit the designs as side-information. In addition to the performance penalty due to quantized data, this approach also requires redundant training complexity at both ends of the system, and is sensitive to transmission errors.

Next, the performance impact of training speaker-dependent LSF quantizers using quantized data was considered. In order to permit training on quantized data in the context of GMVQs, a modified decoder was presented which prevents the training data from having deficient rank. For training data that has been quantized by a speaker-independent GMVQ at transparent quality, the penalty is approximately 1 bit per frame. The penalty becomes very small at encoding rates above 60 bits per frame, suggesting that transmitting parameters at elevated rates during the training phase can erase most of the penalties.

Finally, recursive learning was considered. In this approach, the learning process is carried out in a frame-by-frame fashion, obviating the need to store up large training databases. The approach used here, for GMVQ design, was carried out using the Recursive EM algorithm, along with robust learning schedule developed in [53]. It was shown that there is essentially no penalty for using online learning, implying that the storage required for the training process need not be significant. It was found that the LSF learning process required approximately 30,000 frames (or 10 minutes worth) of training data to achieve comparable performance to batch methods.

The material in this chapter is in preparation for a submission, coauthored with Bhaskar D. Rao, for publication in *IEEE Transactions on Audio, Speech and Language Processing* under the title *"Speaker-Dependent Wideband Speech Coding"*. The dissertation author was the primary researcher and author, and the co-author contributed to or supervised the research which forms the basis for this chapter.

5 Conclusions and Future Work

Quantization for modern coding applications provides a variety of challenges, notably coding of sources with high dimensions, complicated statistics and diverse distortion measures. On top of this, complexity must be kept under control. The first part of this dissertation considered a variety of quantization structures aimed at striking different balances between performance and complexity, and considered training techniques designed to handle the diverse factors impacting performance. Armed with these techniques, the problem of speaker-dependent wideband speech coding was considered. This consisted of quantifying performance gains due to speaker-dependence, and providing solutions to a variety of implementational issues. The various results are discussed in further detail below.

5.1 High-Rate Design of Transform Coders with Gaussian Mixture Companders

Chapter 2 considered the design of scalar transform coders under unknown source statistics and input-weighted mean-squared error. While the performance of scalar transform coders is necessarily somewhat limited, the very low, rate-independent complexity they offer has nevertheless insured that they remain popular in a variety of applications. Contributions in this area include the following:

- The high-rate analysis for scalar transform coders was extended to the case of input-weighted mean squared error.

- A flexible scalar quantizer called the Gaussian Mixture Compander was presented, and details of its implementation were explored. In particular, an iterative decoder is required, and we presented a scheme for insuring its rapid convergence.

- A data-driven design algorithm based on the high-rate approximation was derived, which automatically optimizes the system. As a component of this algorithm, an extension to the EM algorithm was derived for the problem of learning scalar point densities under input-weighted squared error.

- The proposed system was demonstrated for the problem of wideband speech LSF coding. This problem highlights a potential pitfall of high-rate training, which is that the desired operating rate may be below the high-rate regime, particularly for heavily structured quantizers. This results in the gains being significantly smaller than the high-rate estimates would imply. Nevertheless, the high-rate approach led to improvements in outlier statistics at rates of interest.

- In light of the above limitations, modifications to the system for operation at moderate rates were presented. This consists of replacing the GM companders with unstructured scalar quantizers for coefficients that received small allocations. The Lloyd algorithm for scalar quantizer design under input-weighted squared error is presented, and the resulting systems were tested on the wideband speech spectrum coding problem. This approach did indeed improve the average distortion, although it exhibited worse outlier performance than the compander-based system.

5.2 High-Rate Optimized Recursive Quantizers Using Hidden Markov Models

In Chapter 3, a class of more flexible quantizer structures was examined. Based on the GMVQ structure of [12], recursive quantizers were developed using Hidden Markov Models. This chapter revisits the original training techniques developed for GMVQ in light of the high-rate approach, and goes on to demonstrate very good performance on wideband speech spectrum coding. The contributions of this chapter are listed below.

- The High-Rate analysis of GMVQ systems under input-weighted squared error was considered, leading to connections with random coding. In light of this, the GMVQ system was considered in the context of CURTZ coders, which generalize the scalar transform coder and random coders.

- Effects due to large dimensions were considered, in which domain the high-rate distortion integral takes on the form of a weighted cross-entropy.

- Using the above results, the training of GMVQ was considered. The large-dimension approach leads to a weighted EM algorithm, which was compared to the model-based training originally developed for the GMVQ. The convergence of these two approaches was quantified for a variety of example sources.

- The GMVQ was extended using Hidden Markov Models to a recursive system which is able to exploit long-term dependencies in the data. The high-rate training approach was then extended to the recursive case, resulting in a weighted Baum-Welch algorithm.

- The various systems under consideration were demonstrated for the problem of wideband speech spectrum quantization, where it was shown that they lead to significant performance improvements relative to the state-of-the-art.

5.3 Speaker-Dependent Wideband Speech Coding

In Chapter 4, the problem of speaker-dependent coding in a CELP framework was considered. The flexible GMVQ quantizers and design approach developed in the previous chapters were leveraged to quantify the gains available in speaker-dependent coding of the various parameters common to CELP systems. Then, the various implementational issues, such as online training and robustness, were addressed. The specific contributions are listed below.

- Gains from speaker-dependent coding of spectrum, adaptive codebook and fixed excitation parameters were experimentally quantified. It was found that a reduction of 10% in the bitrate are achievable for spectrum coding, and that there is very little gain to be had from the adaptive codebook parameters. Most significant were the gains in the fixed excitation, which amounted to 10-20% of the entire bit budget of a typical wideband speech coder.

- Various methods for exploiting speaker-dependent gains were presented. In addition to simply reducing the bit-rate, it is possible to instead reduce coding complexity. Another possibility is to use speaker-dependent rates to achieve uniform quality over all speakers.

- Safety-net coding was incorporated into the GMVQ framework in order to provide robustness, and the training algorithm was modified to allow precise trade-offs between robustness and performance.

- A variety of architectures for online learning were presented, which strike different balances between performance, training complexity and communications overhead.

- The problem of learning GMVQ systems from quantized data was considered, and a modification to the transform coder was developed to avoid problems in this area. Then, the penalty for learning on quantized data was experimentally quantified.

- Methods for recursive learning of GMVQ systems were presented, which eliminate the storage requirements associated with batch learning and enable adaptive operation. The convergence of this approach was investigated and it was demonstrated that performance equivalent to that of batch learning can be attained.

5.4 Future Work

This section summarizes some of the possible extensions of this dissertation. The first subsection discusses extensions in the realm of structured quantizers, while the second considers extensions related to speaker-dependent coding.

5.4.1 Improved Recursive Quantizers

The recursive quantization structures presented in Chap. 3 operate by changing the parameters of a GMVQ at each time step, based on previous data. However, in order to keep complexity low, the systems under consideration only affect the weights and cluster means, leaving the covariances of each component fixed. A more flexible, and hence higher-performance, recursive scheme would also modify the covariances based on previous data. This is similar to the ARCH methods used in econometrics for predicting changes in volatility (c.f. [58], [59]). Such a method would allow the clusters to expand in volatile segments, and shrink down in very predictable segments. However, because the operation of the component coders in a GMVQ (typically transform coders) depend on the Eigendecompositions of the covariances, the potential complexity penalty for altering the covariances is substantial. A promising avenue would be to consider only certain constrained modifications designed to make the eigendecomposition easy to update. Examples would be simple scaling, or low-rank updates. Training methods using Maximum Likelihood are already available for these sorts of models in the ARCH literature, which could be extended to input-weighted distortion measures through the same

weighting approach as is used in GMVQ training.

5.4.2 User-Dependent Speech Coding

The results in Chap. 4 consider only performance gains due to speaker-dependence. Speaker-dependence is a function of anatomy and speaking style, but ignores other factors such as background noise, acoustic environment and the response of end-user equipment, all of which will impact performance in a real telecommunications setting. A user-dependent system would take all of these factors into account. It is possible for user-dependent gains to be smaller or larger than speaker-dependent gains, depending on the extent to which the other factors make the users more or less similar. However, while we have not attempted to quantify user-dependent gains, the various implementational techniques presented would apply directly to a user-dependent system. Thus, the way has been cleared for larger, more expansive experiments that would be able to quantify all of these various factors. This framework can also be used to aid in developing models of how the various environmental factors vary from user to user, and from time to time. In particular, the performance of adaptive systems is of interest here, as a given user's environment is subject to fairly rapid change in the context of mobile communications.

Bibliography

[1] S. Na and D. L. Neuhoff, "Bennett's Integral for Vector Quantizers", IEEE Trans. on Info. Theory, Vol. 41, (no.4), July 1995.

[2] J.J.Y. Huang and P.M. Schultheiss, "Block Quantization of Correlated Gaussian Random Variables" IEEE Trans. on Comm. Systems, Vol. 11, Issue 3, Sept. 1963.

[3] V. K. Goyal, J. Zhuang and M. Vetterli, "Transform Coding with Backward Adaptive Updates" IEEE Trans. on Info. Theory, Vol. 46, (no.4), July 2000.

[4] J. K. Su and R. M. Mersereau, "Coding Using Gaussian Mixture and Generalized Gaussian Models" International Conf. on Image Proc., Sept. 1996.

[5] M. Effros, H. Feng and K. Zeger, "Suboptimality of the Karhunen-Love Transform for Transform Coding" IEEE Trans. on Info. Theory, Vol. 50, (no.8), August 2004.

[6] C. Archer and T. K. Leen, "A Generalized Lloyd-Type Algorithm for Adaptive Transform Coder Design" IEEE Trans. on Signal Proc., Vol. 52, (no.1), January 2004.

[7] R. M. Gray and D. L. Neuhoff, "Quantization" IEEE Trans. on Info. Theory, Vol. 44, (no.6), October 1998.

[8] A. Gersho and R. M. Gray, "Vector Quantization and Signal Compression" Norwell, MA: Klewer Academic Publishers, 1991.

[9] A. H. Gray, Jr., R. M. Gray and J. D. Markel, "Comparison of Optimal Quantizations of Speech Reflection Coefficients" IEEE Trans. on Acoustics, Speech, and Signal Proc., Vol. ASSP-25, (no.1), February 1977.

[10] R. Viswanathan and J. Makhoul, "Quantization Properties of Transmission Parameters in Linear Predictive Systems" IEEE Trans. on Acoustics, Speech, and Signal Proc., Vol. ASSP-23, (no.3), June 1975.

[11] F. K. Soong and B.-H. Juang, "Optimal Quantization of LSP Parameters" IEEE Trans. on Speech and Audio Proc., Vol. 1, (no.1), January 1993.

[12] A. D. Subramaniam and B. D. Rao, "PDF Optimized Parametric Vector Quantization of Speech Line Spectral Frequencies" IEEE Trans. on Speech and Audio Proc., Vol. 11, (no.2), March 2003.

[13] P. Hedelin and J. Skoglund, "Vector Quantization Based on Gaussian Mixture Models" IEEE Trans. on Speech and Audio Proc., Vol. 8, (no.4), July 2000.

[14] F. Lahouti and A. K. Khandani, "Quantization of LSF Parameters Using a Trellis Modeling" IEEE Trans. on Speech and Audio Proc., Vol. 11, (no.5), Spetember 2003.

[15] S. Ragot, J.-P. Adoul, R. Lefebvre and R. Salami, "Low Complexity LSF Quantization for Wideband Speech Coding" IEEE Workshop on Speech Coding, 1999.

[16] M. Ferhaoui and S. Van Gerven, "LSP Quantization In Wideband Speech Coders" IEEE Workshop on Speech Coding, 1999.

[17] S. Chi, S. Kang and C. Lee, "Safety-Net Pyramid VQ of LSF parameters for Wideband Speech Codecs" Electronics Letters, Vol. 37, (no.11), May 2001.

[18] E. R. Duni and B. D. Rao, "High-Rate Optimized Recursive Vector Quantization Structures Using Hidden Markov Models" IEEE Trans. on Audio, Speech and Lang. Proc., Vol. 15, (no. 3), March 2007.

[19] T. Z. Shabestary and Per Hedelin, "Vector Quantization by Companding a Union of Z-Lattices" IEEE Trans. on Info. Theory, Vol. 51, (no.2), February 2005.

[20] W. R. Gardner and B. D. Rao, "Theoretical Analysis of the High-Rate Vector Quantization of LPC parameters", IEEE Trans. on Speech and Audio Proc., Vol.3, (no.5), September 1995.

[21] J. Li, N. Chaddha and R. M. Gray, "Asymptotic Performance of Vector Quantizers with a Perceptual Distortion Measure" IEEE Trans. on Info. Theory, Vol. 45, (no.4), May 1999.

[22] T. Linder and R. Zamir, "High-Resolution Source Coding for Non-Difference Distortion Measures: The Rate-Distortion Function" IEEE Trans. on Info. Theory, Vol. 45, (no.2), March 1999.

[23] T. Linder, R. Zamir and K. Zeger, "High-Resolution Source Coding for Non-Difference Distortion Measures: Multidimensional Companding" IEEE Trans. on Info. Theory, Vol. 45, (no.2), March 1999.

[24] A. Gersho, "Asymptotically Optimal Block Quantization", IEEE Trans. on Info. Theory, Vol. IT-25, (no.4), July 1979.

[25] T. D. Lookabaugh and R. M. Gray, "High-Resolution Quantization Theory and the Vector Quantizer Advantage" IEEE Trans. on Info. Theory, Vol. 35, (no.5), September 1989.

[26] H. Le Vu and L. Lois, "Efficient Distance Measure for Quantization of LSF and Its Karhunen-Loeve Transformed Parameters" IEEE Trans. on Speech and Audio Proc., Vol. 8, (no.6), November 2000.

[27] J.-F. Cardoso, "Blind Signal Separation: Statistical Principles" Proc. of the IEEE, Vol. 86, Issue 10, Oct. 1998.

[28] J. H. Manton, "Optimization Algorithms Exploiting Unitary Constraints" IEEE Trans. on Signal Proc., Vol. 50, (no.3), March 2002.

[29] R. M. Neal and G. E. Hinton, "A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants" Learning in Graphical Model, M.I. Jordan (editor).

[30] R.M. Gray, J.C. Kieffer and Y. Linde, "Locally Optimal Block Quantizer Design" Information and Control, Vol. 45, pp. 178-198, May 1980.

[31] A. D. Subramaniam, W. R. Gardner and B. D. Rao, "Low-Complexity Source Coding Using Gaussian Mixture Models, Lattice Vector Quantization, and Recursive Coding with Application to Speech Spectrum Quantization", IEEE Trans. on Speech and Audio Proc.: Accepted for future publication.

[32] G. Ott, "Compact Encoding of Stationary Markov Sources", IEEE Trans. on Info. Theory, vol. IT-13, (no.1), Jan. 1967.

[33] D. M. Goblirsch and N. Farvardin, "Switched Scalar Quantizers for Hidden Markov Sources", IEEE Trans. on Info. Theory, vol. 38, (no.5), Sept. 1992.

[34] T. Z. Shabestary and P. Hedelin, "LSP Quantization by a Union of Locally Trained Codebooks" IEEE Trans. on Speech and Audio Proc., Vol. 13, Issue 5, Part 2, Sept. 2005

[35] J. A. Bucklew, "Companding and Random Quantization in Several Dimensions", IEEE Trans. on Info. Theory, Vol. IT-27, (no.2), March 1981.

[36] P. L. Zador, "Asymptotic Quantization Error of Continuous Signals and the Quantization Dimension", IEEE Trans. on Info. Theory, vol. IT-28, (no.2), March 1982.

[37] A. D. Subramaniam "Gaussian Mixture Models in Compression and Communication", Ph.D. Thesis, UCSD. http://dsp.ucsd.edu/∼anand/thesis.pdf

[38] J. Samuelsson and P. Hedelin, "Recursive Coding of Spectrum Parameters", IEEE Trans. on Speech and Audio Proc., Vol. 9, (no.5), July 2001.

[39] J. N. Kapur, "Entropy optimization principles with applications", Boston : Academic Press, 1992.

[40] G. H. Hardy, "Inequalities", Cambridge University Press, 1934.

[41] L. Rabiner and B.-H. Juang, "Fundamentals of Speech Recognition", New Jersey: Prentice Hall, 1993.

[42] X. Huang and K.-F. Lee, "On Speaker-Indpendent, Speaker-Dependent, and Speaker-Adaptive Speech Recognition" IEEE Trans. on Speech and Audio Proc., Vol. 1, (no,2), April 1993.

[43] H.D. Pfister and H.L. Pfister, "Speaker Dependent Speech Compression for Low Bandwidth Communication" ICASSP 1996.

[44] I. Potamitis, N. Fakotakis and G. Kokkinakis, "Gender-Dependent and Speaker-Dependent Speech Enhancement" ICASSP 2002.

[45] C.-H. Lee, S.K. Jung and H.-G. Kang, "Applying a Speaker-Dependent Speech Compression Technique to Concatenative TTS Synthesizers " IEEE Trans. on Audio, Speech and Lang. Proc., *Accepted for Publication.*

[46] C.M. Ribeiro and I.M. Trancoso, "Application of Speaker Modification Techniques to Phonetic Vocoding" ICSLP 96.

[47] P. Ojala, P. Haavisto, A Lakaniemi and J. Vainio, "A Novel Pitch-Lag Search Method Using Adaptive Weighting and Median Filtering" IEEE Workshop on Speech Coding, 1999.

[48] J.-L. Gauvain and C.-H. Lee, "Maximum *a Posteriori* Estimation for Multivariate Gaussian Mixture Observations of Markov Chains" IEEE Trans. on Speech & Audio Proc., vol. 2, no. 2, April 1994.

[49] Y. Gotoh, M.M. Hochberg and H.F. Silverman, "Efficient Training Algorithms for HMM's Using Incremental Estimation" IEEE Trans. on Speech & Audio Proc., vol. 6, no. 6, Nov. 1998.

[50] D.M. Titterington, "Recursive Parameter Estimation using Incomplete Data" J. R. Statist. Soc. B, vol. 46, no. 2, pp. 257-67, 1984.

[51] D.M. Titterington and J-M. Jiang, "Recursive Estimation Procedures for Missing-Data Problems" Biometrika, vol. 70, no. 3, pp. 613-24, 1983.

[52] P-J. Chung and J. F. Bohme, "Recursive EM and Sage-Inspired Algorithms With Application to DOA Estimation" IEEE Trans. on Signal Proc., Vol. 53, no. 8, Aug. 2005.

[53] M. Sato, "Fast Learning of On-Line EM Algorithm" Unpublished manuscript.

[54] J. Zheng, E. R. Duni and B. D. Rao, "Analysis of Multiple-Antenna Systems with Finite-Rate Feedback Using High-Resolution Quanitization Theory" *To Appear*, IEEE Trans. on Signal Proc., 2007.

[55] J. H. Plasberg and W. B. Kleijn, "The Sensitivity Matrix: Using Advanced Auditory Models in Speech and Audio Processing" IEEE Trans. on Audio, Speech and Lang. Proc., Vol. 15, no. 1, Jan. 2007.

[56] T. Linder, R. Zamir and K. Zeger, "On Source Coding with Side-Information-Dependent Distortion Measures" IEEE Trans. on Info. Theory, Vol. 46, no. 7, Nov. 2000.

[57] W. C. Chu, "Speech Coding Algorithms" New Jersey: John Wiley & Sons, Inc., 2003.

[58] T. Bollerslev, "Generalized Autoregressive Conditional Heteroskedasticity" Journal of Econometrics, vol. 31, pp. 307-327, 1986.

[59] R. F. Engle. "Autoregressive Conditional Heteroscedasticity with Estimates of Variance of United Kingdom Inflation" Econometrica, vol. 50. pp. 987-1008, 1982.