Latent Variable Framework for Modeling and Separating Single-Channel Acoustic Sources

Madhusudana Shashanka

Dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

BOSTON

UNIVERSITY

BOSTON UNIVERSITY

GRADUATE SCHOOL OF ARTS AND SCIENCES

Dissertation

LATENT VARIABLE FRAMEWORK FOR MODELING AND SEPARATING SINGLE-CHANNEL ACOUSTIC SOURCES

by

MADHUSUDANA SHASHANKA

B.E. (Honors), Birla Institute of Technology and Science, Pilani, India, 2003

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2008

Approved by

First Reader

Barbara G. Shinn-Cunningham, Ph.D.
Associate Professor of Cognitive and Neural Systems
Associate Professor of Biomedical Engineering
Boston University

Second Reader

Paris Smaragdis, Ph.D.
Research Scientist
Mitsubishi Electric Research Laboratories

Third Reader

Frank Guenther, Ph.D.
Associate Professor of Cognitive and Neural Systems
Boston University

The first mark of intelligence, to be sure, is not to start things; the second mark of intelligence is to pursue to the end what you have started.

Panchatantra

Acknowledgments

Several people have contributed towards the completion of this work. First of all, I would like to express immense gratitude to my thesis advisor, Prof. Barbara Shinn-Cunningham. She has been a wonderful guide and mentor giving me constant support and encouragement. I thank her for helping me crystallize my academic and professional goals and providing me great freedom to explore and pursue my interests. She has actively helped me in developing collaborations and professional contacts for which I will remain forever grateful.

Secondly, this thesis would not have been possible without Dr. Bhiksha Raj and Dr. Paris Smaragdis at Mitsubishi Electric Research Labs. They provided technical supervision for this work and have helped me in every way, from ideas and theoretical discussions to helping me with parts of the implementation code. And they have been great mentors, with generous advice on any topic I needed help with, especially with my career plans. I have learned a great deal during my collaboration with them. I cannot thank them enough!

I would like to thank the rest of my thesis committee members - Profs. Frank Guenther, Daniel Bullock and Eric Schwartz - for reviewing this work and providing helpful feedback.

I also want to extend my thanks to the faculty of the Department of Cognitive and Neural Systems as well as faculty in the Hearing Research Center, especially within the "binaural gang," for creating a great environment conducive for research. Special thanks to all the staff at CNS - Brian, Susanne, Cindy, Robin and especially Carol - for helping me navigate through the bureaucracy of grad school. Credit is due to many others but I refrain from listing all of them. Suffice to say that I'm thankful to all my friends and well-wishers without whom my journey through grad-school would have been much harder.

Finally, I dedicate this thesis to my parents, whose unconditional support for all my endeavors is what made this thesis work possible. Thank you!

LATENT VARIABLE FRAMEWORK FOR MODELING AND SEPARATING SINGLE-CHANNEL ACOUSTIC SOURCES

(Order No.)

MADHUSUDANA SHASHANKA

Boston University Graduate School of Arts and Sciences, 2008

Major Professor: Barbara G. Shinn-Cunningham, Associate Professor of Cognitive and Neural Systems and Biomedical Engineering

Abstract

Auditory Scene Analysis refers to the human ability to extract different perceptual objects from a sound mixture. Replicating this ability in artificial systems has been an active area of research, related both to how one characterizes acoustic sources and separates sources from mixtures. The focus of this thesis is to develop models and algorithms that provide a framework to address these questions. The framework comprises latent variable models that employ hidden variables to model unobservable quantities. Such models are appropriate for obtaining representations of data that make hidden structure explicit. This work shows how one can utilize these ideas for the problem of source separation using single-channel audio signals.

The proposed framework focuses on learning the time-frequency (TF) structure in a data-driven manner. TF representations of sounds are modeled by treating the energy in every TF bin as histogram counts of multiple draws. This formulation allows the extraction of the characteristic frequency structure of individual sources as latent components and models the sources as additive combinations of these components. The framework is then extended to incorporate the idea of sparse coding to overcome an important limitation of the basic model: an upper bound on the number of extractable components. Sparsity, imposed in the form of an entropic prior distribution, allows extraction of overcomplete sets of components that are more expressive and better characterize the sources. The statistical foundation of the framework makes it amenable to other extensions where known

or hypothesized structure about the data can be easily incorporated by imposing appropriate prior distributions. Theoretical analysis of the proposed methods and algorithms for parameter inference are presented.

Applications of the models to real-world problems are evaluated and discussed. The latent components learned from acoustic sources are used in a supervised setting for source separation and in a semi-supervised setting for denoising. Unlike approaches based on time-frequency masks that reconstruct partial spectral descriptions of sources by identifying time-frequency bins in which a source dominates, this approach reconstructs entire spectral descriptions of all sources. Various experimental results demonstrate the utility of the proposed framework.

Contents

List of Tables

List of Figures

List of Abbreviations

ASA	.....	Auditory Scene Analysis
BSS	.....	Blind Source Separation
CASA	.....	Computational Auditory Scene Analysis
dB	.....	decibels
EM	.....	Expectation-Maximization
FFT	.....	Fast Fourier Transform
GMM	.....	Gaussian Mixture Model
HMM	.....	Hidden Markov Model
ICA	.....	Independent Component Analysis
kHz	.....	kilohertz
KL	.....	Kullback-Leibler
LSA	.....	Latent Semantic Analysis
MAP	.....	Maximum a Posteriori
ML	.....	Maximum Likelihood
NMF	.....	Non-negative Matrix Factorization
PCA	.....	Principal Component Analysis
PLCA	.....	Probabilistic Latent Component Analysis
PLSA	.....	Probabilistic Latent Semantic Analysis
SER	.....	Speaker Energy Ratio
SNR	.....	Signal-to-Noise Ratio

STFT	.....	Short-Time Fourier Transform
SVD	.....	Singular Value Decomposition
TF	.....	Time-Frequency
USPS	.....	United States Postal Service
WSJ	.....	Wall Street Journal

Chapter 1

Introduction

1.1 Introduction

The study of human perception is a fascinating subject. Much research has been devoted to understand this phenomenon for different modalities such as vision, audition and olfaction. The domain of audition - the focus of this thesis, however, did not receive as much attention as vision until the last few decades. This is captured in a remark by Metzger (1953) (abridged in English and quoted by Plomp, 2002):

> The achievements of the ear are indeed fabulous. While I am writing, my elder son rattles the fire rake in the stove, the infant babbles contentedly in his baby carriage, the church clock strikes the hour, a car stops in front of the house, next door one of the girls is practicing on the piano, at the front door her mother converses with a messenger, and I can also hear the fine scraping of the point of the pencil and my hand moving on the paper. In the vibrations of air striking my ear, all these sounds are superimposed into a single extremely complex stream of pressure waves. Without doubt the achievements of the ear are greater than those of the eye. Why do the psychologists, particularly the Germans, stick so stubbornly to vision research?

The comment about German psychologists aside, this quote highlights the effortlessness with which the human auditory system accomplishes this feat of separating the input into distinct auditory *objects* with the listener barely noticing the process. Cherry, in a classic study published the same year (Cherry, 1953), christened this process as the *Cocktail Party Effect* referring to our ability to follow one speaker in the presence of others.

The Cocktail Party Effect is a very challenging problem when viewed from a computational perspective. An analogy due to Bregman (1990) illustrates the difficulties involved in the process. Consider that you are on the edge of a lake and have dug two narrow channels (a few feet long and a few inches wide, spaced a few feet apart) up from the side of the lake. If you stretch a handkerchief across either channel, waves reaching the side of the lake will travel up the channel and cause the handkerchief to go into motion. By looking at the handkerchief, you should be able to infer information about the activities in the lake such as the number, positions, directions of travel and other attributes of various objects on the lake (boats, swimmers, etc.). This seems like an impossible task but is a strict analogy to the problem faced by the auditory system. The lake represents the air that surrounds us, the canals represent the ear canals and the handkerchief represents the ear drum. We barely notice the complexity of hearing but it becomes obvious when presented this way.

The problem has been considered before (Helmholtz, 1863) but since Cherry's study, there has been a spurt of interest in understanding the cocktail party effect and replicating this ability in machines. See (Haykin and Chen, 2005; Bronkhorst, 2000) for reviews of recent developments in the field. Haykin and Chen (2005) list three fundamental questions pertaining to the cocktail party phenomenon:

- What is the cocktail party problem?
- How does the brain solve it?
- Is it possible to build a machine capable of solving it in a satisfactory manner?

The research work presented in this thesis addresses the last question. We are interested in the question of whether it is possible to derive computational algorithms that can solve the problem.

Let us consider a mathematical formulation of the problem. Let $x_1(t)$ and $x_2(t)$ represent the acoustic signals arriving at the two ears at time instant t. Let there be N sources

$s_1(t), \ldots, s_N(t)$ where $N \geq 2$. We can write the mixtures $x_k(t)$[1] as

$$x_k(t) = \sum_{j=1}^{N} a_{kj} s_j(t - \delta_{kj}), \qquad k \in \{1, 2\}, j \in \{1, \ldots, N\} \tag{1.1}$$

where parameters a_{kj} and δ_{kj} are the attenuation coefficients and the time delays associated with the path from the jth source to the kth receiver (ear). The human auditory system analyzes the mixture signals $x_k(t)$ such that the resulting auditory perceptual objects often have a one-to-one correspondence with the actual sound sources $s_j(t)$ making up the mixture. Most artificial systems formulate the source separation problem in a similar way. The problem is to estimate the sources $s_j(t)$ from observed signals $x_1(t)$ and $x_2(t)$. They can be categorized into two groups – systems that work with multiple microphone (multi-channel) recordings (e.g., Brandstein and Ward, 2001), and systems that work with single microphone (single-channel) recordings (e.g., Roweis, 2001). Equation (1.1) represents the multi-channel case with two microphones. In the case of single-channel mixtures, the equation can be simplified by subsuming the delays and attenuation coefficients within the source signals (without loss of generality) and can be written as

$$x(t) = \sum_{j=1}^{N} s_j(t). \tag{1.2}$$

The focus of this thesis will be the case of single-channel audio signals, exemplified by the formulation in the above equation.

The main difficulty in solving the cocktail party problem lies in the fact that the system is usually *under-determined*. In other words, there is no one unique way in which source signals can be reconstructed from the available information. For example, in the single-channel case one can choose several distinct sets of values for $s_j(t)$ such that the relation (1.2) is satisfied. There is not enough "information" in the mixture signal to reconstruct the sources exactly. However, it is possible to *estimate* or obtain *approximate* solutions by utilizing some information about the problem. For example, if we know that we are trying to separate a

[1] This is a simple formulation used for illustration. One has to consider the frequency dependency of the delay term to accurately model the mixing process.

male speaker from a female speaker, we could use the fact that the female speaker usually has a higher pitch. To get a computer accomplish the separation task, we will have to let the computer "know" about this information. There are several methods researchers have used, but most are based on two underlying approaches. The first approach is to understand how the human auditory system solves this problem and utilize similar rules and heuristics in the artificial system. The second approach is an engineering approach where the idea is to utilize probability and signal processing theories to take advantage of known or hypothesized structure/statistics of the source signals and/or the mixing process to estimate the sources.

Research on auditory perception focused on how humans solve this puzzle. This culminated with the seminal work of Bregman (1990). Bregman outlines many rules and heuristics that the auditory system uses to understand and organize sound, or to perform *auditory scene analysis* (ASA). Since then, there have been many attempts to build machines that are capable of aspects of ASA, a discipline known as *Computational Auditory Scene Analysis* (CASA; Brown and Cooke, 1994; Rosenthal and Okuno, 1998; Divenyi, 2005). Many attempts have been made to build such systems (e.g., see work by Vercoe and Cumming, 1988; Duda et al., 1990; Mellinger, 1991; Cooke, 1991; Brown, 1992; Brown and Cooke, 1994; Ellis, 1991, 1992, 1996; Grossberg et al., 2004; Roman, 2005, among others). These systems include both monaural (single-channel) and binaural (two-channel) systems. Most of these CASA attempts can be characterized as descriptions of computational implementations of the views outlined by Bregman. They include substantial knowledge of the psychophysical characteristics of the human auditory system and the heuristics used by it. As Smaragdis (2001) points out, this approach has inherent limitations, mainly due to the difficulty in reconciling subjective and fuzzy concepts used by Bregman such as "similarity", "proximity" and "continuity" and the strictly deterministic platform of computer implementations.

On the other hand, researchers in the statistical signal processing community have approached a computationally equivalent problem from a different perspective. This problem,

usually termed as Blind Source Separation (BSS; Choi et al., 2005), involves finding the set of source signals that combine to form the observed mixture of signals in a blind (i.e. unsupervised) manner. There are two categories – beamforming techniques and Independent Component Analysis (ICA). Beamforming (Brandstein and Ward, 2001) utilizes information about the directions of sources, differences in the level and times of arrival at different sensors, and other sensor-configuration based information to estimate the sources. ICA (Hyvarinen, 1999) uses statistical information and assumptions about the nature of source signals to estimate them. Specifically, it assumes that the source signals are generated by statistically independent random processes. Both of these approaches, however, require at least two different mixture signals, making them unsuitable for the single-channel case.

We take a machine learning approach and formulate the problem in a supervised setting. We assume that one or more of the sources present in the mixture are "known." In other words, sample waveforms of the known sources (recorded in the absence of other interfering acoustic objects) are available for analysis before we tackle the problem of separating the mixture. The idea is to analyze the available "training data" to extract characteristics unique to each known source and then utilize the learned information for applications such as source separation. The following section provides an overview of the contributions of this thesis.

1.2 Overview

This thesis explores modeling single-channel acoustic signals. The focus is on providing a probabilistic framework to model the sounds so that one can either extract the underlying structure and understand a particular class of sounds (e.g. analysis of polyphonic music) or use these models for applications such as source separation. The proposed work considers the problem from a strictly computational perspective and does not take into account how the human auditory system solves the problem. The aim is not simply to build *a* system capable of source separation but to provide a computational framework grounded in theoretical principles with which one can attempt to solve such problems. The models

we present use a time-frequency representation of audio signals. This kind of representation allows us to view the sound in terms of energy present at every frequency component and time frame.

There are two main themes in this work. First, the focus is the statistical model that underlies the computational framework. As mentioned above, sounds rarely occur in isolation. Even for a given source, the sound at any instant is usually composed of many different underlying components or building blocks. For example, a guitar chord contains many notes, which can be thought of as the underlying components. This implies that

- energy in a particular time-frequency bin for a signal has contributions from all sources/components that combine to compose the signal.

We would like to learn these underlying components or building blocks using the developed probabilistic framework. We use models that employ latent variables, which allows us to explicitly express the energy in a time-frequency bin as arising from many components. The latent variables correspond to the underlying components that are unobservable. We use statistical techniques to estimate the parameters of the model, and thus "learn" the components from training data. We present theoretical analysis and provide experimental results that demonstrate applications.

The second theme of this thesis is to investigate sparse coding. We consider models in which the aim is to represent observed data as an additive mixture of a set of canonical components. In this context, sparse coding refers to a scheme in which only a small number of components are required to represent any particular instance of data. In an overcomplete code, there are more components than the dimensionality of the data. A sparse overcomplete code is one that combines notions of both sparsity and overcompleteness. In the context of modeling acoustic signals, this concept has significant implications. A given class of sounds that we want to analyze can have an arbitrary number of building blocks. However, mathematics constrains us so that the number of components extracted is equal to or less than the dimensionality of the time-frequency representation (i.e., the number of frequency bins). Extracting more components will lead to trivial solutions or indetermi-

nacy. But *the number of underlying components (ground truth) does* not *depend on the representation.* A sparse overcomplete code allows us to get around the problem - we can have a large set of components to explain the entire signal; however, any particular instant will have contributions from only a few components. The proposed work shows how this computational principle can be utilized in a probabilistic framework. Again, we present the theory and show experimental results that demonstrate the efficacy of the proposed methods.

The thesis is organized as follows. Chapter 2 provides background about modeling single-channel audio, reviewing time-frequency representations of sound and set the stage for later chapters. Previous approaches that have been proposed for single-channel source separation are then reviewed. Chapters 3 and 4 represent the core part of this thesis research. Chapter 3 presents the latent variable framework, while Chapter 4 extends the framework to incorporate the concept of sparse coding. We present conclusions and avenues for future work in Chapter 5.

Chapter 2

Modeling Time-Frequency Structure from Audio

Sound is the vocabulary of nature.

Pierre Schaeffer

2.1 Introduction

This chapter presents background about modeling structure from acoustic signals. Time-frequency representations of sounds are briefly reviewed and conventions used in the remaining chapters are defined.

2.2 Representation

Sound consists of pressure variations propagating through a medium such as air. The common digital representation of an acoustic signal is the sampled waveform, where each sample represents the sound pressure level at a particular time instant. Figure 2·1 shows the time-domain pressure signal of a speech sound.

Real-world sounds are time-varying, and all of their meaning is encoded in these variations in frequency content over time. A time-domain waveform does not represent the information present in a sound in an explicit way. We can instead utilize a time-frequency representation, which explicitly represents the energy in every time-frequency bin. The time dimension corresponds to a sequence of time-frames (successive fixed-width snippets of the waveform, possibly windowed and overlapping) representing each frequency dimension, corresponding to the output of one of a bank of filters. This is consistent with auditory

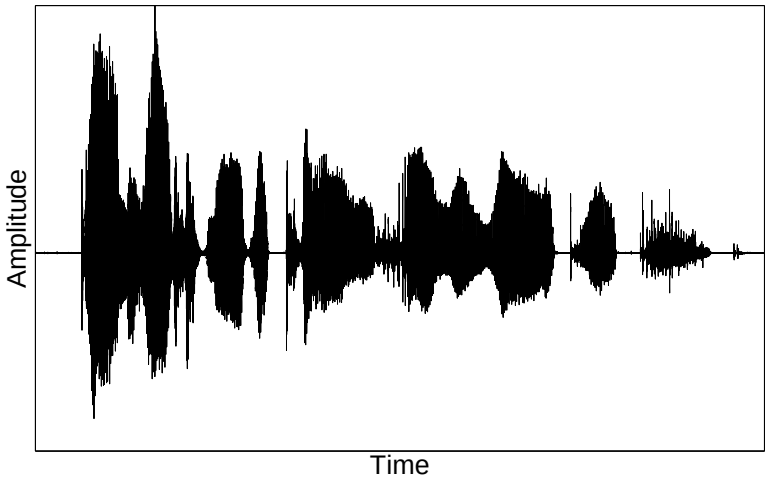


Figure 2·1: Time representation of a female speech utterance "Don't ask me to carry an oily rag like that."

physiology. The first stage of analyzing sounds in biological systems is to decompose the signal into multiple frequency bands through an intricate mechano-neural interaction in the cochlea. As a result of this processing, the initial neural representation of sound is well approximated as a process that takes an incoming acoustic signal and decomposes it into an ongoing time-frequency representation.

A useful method to obtain a time-frequency representation is the Short Time Fourier Transform (STFT), introduced by Gabor (1946). The incoming time signal is multiplied by a time windowing function that is non-zero for a short period of time. The Fourier transform of the output of the window is taken as the window slides along time axis. This results in a two dimensional time-frequency representation of the signal that shows how frequency content changes with time. Results are often displayed as *spectrograms* that show energy (using color or grayscale) as a function of time and frequency. Figure 2·2 shows an example spectrogram of the waveform shown in Figure 2·1[2]. In the presence of

[2]The STFT of a time-signal produces a complex number at every time-frequency bin. We only consider the *magnitude* of the STFT to create a spectrogram. Unless otherwise stated, spectrograms represent only the magnitude of the STFT. Also, all figures display the *logarithm* of the magnitude spectrogram for enhanced contrast between regions of high and low energy.

multiple sounds, energy is combined from all sources at every time-frequency component in the two-dimensional representation of the sound mixture.

The STFT is just one of the many ways of obtaining a time-frequency representation. The STFT can be thought of as a representation of the output of a bank of filters that slices the spectrum into equal width (in Hz), non-overlapping slices. Instead, one can choose a different filterbank to obtain the frequency bin indices. Examples include the constant-Q transform (Brown, 1991) or filter banks based on psychoacoustic measurements such as the *gammatone* (Patterson-Holdsworth) (Patterson et al., 1995) and the *gamma-chirp* filter banks (Irino and Patterson, 1997). The framework we will propose will be applicable on any time-frequency representation as long as the entries in all time-frequency bins are non-negative and represent an "energy-like" quantity that can be approximated to combine additively in the case of sound mixtures. This thesis only considers the magnitude of the STFT to generate the time-frequency representation that will be considered. Also, the term *spectral vector* will be used to denote a particular analysis frame (corresponding to a time bin) of such a spectrogram.

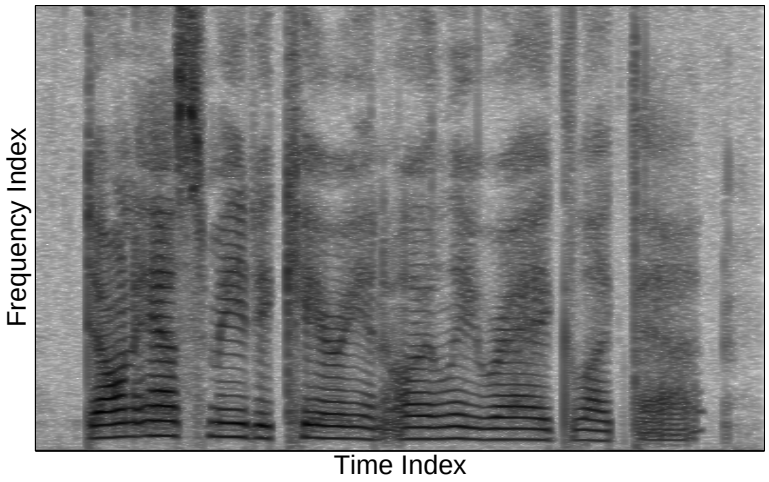


Figure 2·2: Time-frequency representation of the waveform of Figure 2·1. The figure plots the log-magnitude of the short-time Fourier transform of the signal.

2.3 Modeling Time-Frequency Structure

The work proposed in this thesis focuses on learning time-frequency structure in a purely data-driven fashion from audio data. This section reviews other methods that have been proposed for audio source separation.

2.3.1 CASA Methods

The field of Computational Auditory Scene Analysis (CASA) aims to build sound separation systems that are based on known principles of human perception (Brown and Wang, 2005). Bregman (1990) outlined various rules and heuristics used by the auditory system to perform auditory scene analysis. Since his seminal work, the field of CASA has emerged with the goal of building artificial systems that implement the principles he outlined. Most systems, motivated by psychophysics and physiology to a lesser extent, are binaural systems; however, monaural systems have also been proposed.

An important concept in CASA is that of a time-frequency mask. Consider the problem of separating out a target signal from a mixture. The idea is to assign a higher weight to those time-frequeny regions of the mixture in which the target is dominant (has more energy) and low weight to the rest of the spectrogram. The mask multiplies the mixture spectrogram and the time-domain target signal is reconstructed from the weighted time-frequency representation. Weintraub (1985) was the first to use this approach and many researchers have adopted it since then (Brown, 1992; Brown and Cooke, 1994; Roweis, 2001). The values of the time-frequency mask can be binary or real-valued. In the case of a binary mask, one only retains those time-frequency regions of the mixture where the target is dominant, and discards regions in which the target is weaker than the interference. Specifically, the binary mask has a value of 1 where the target is dominant and 0 elsewhere. The intuition is that the dominant source masks energy of the weaker source in any particular time-frequency bin, and based on the spectrotemporal sparsity of many natural signals, a reconstruction based on the time-frequency bins in which the target dominates is sufficient for relatively accurate reconstruction. The *ideal* binary mask, a binary mask

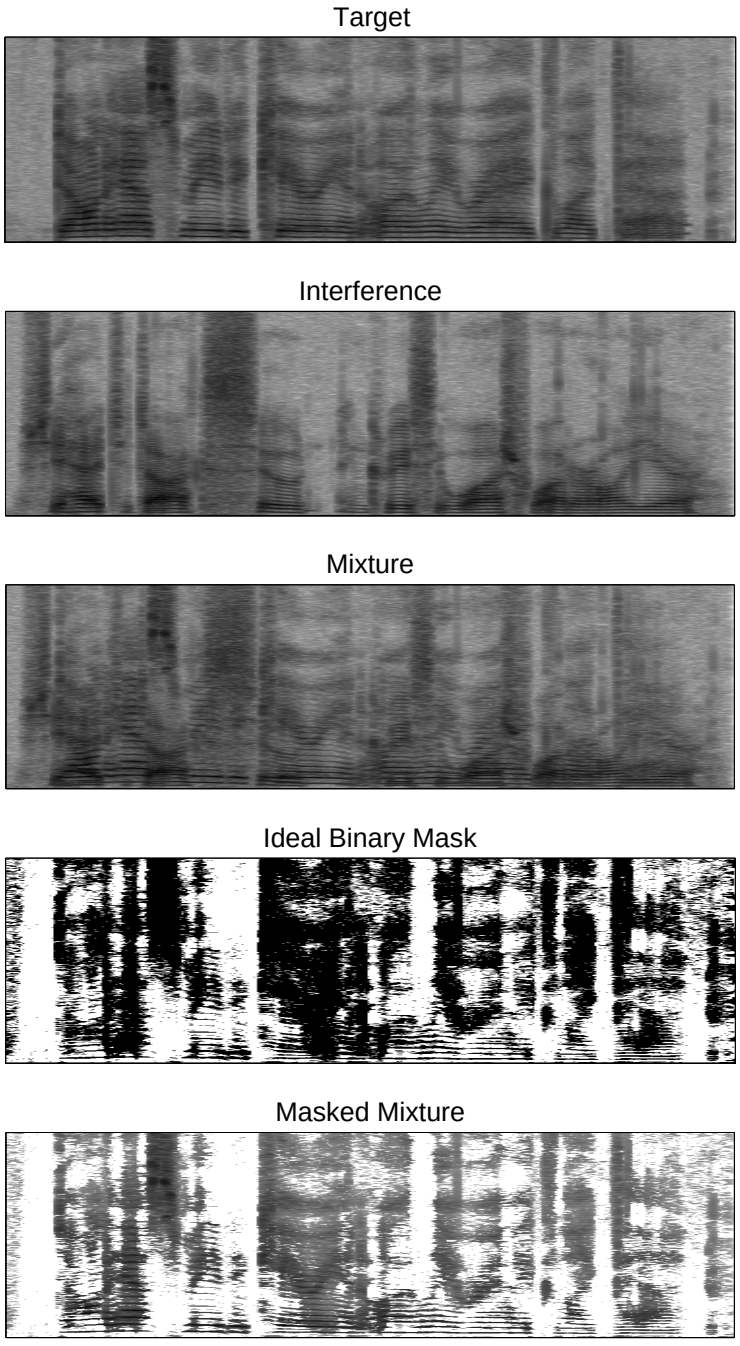


Figure 2·3: Illustration of the ideal binary mask. Target is a female utterance "Don't ask me to carry an oily rag like that" and interference is a male utterance "She had your dark suit and greasy wash water all year." In the Ideal Binary Mask panel, black pixels indicate 1 (allow) and white pixels indicate 0 (discard). The masked mixture is shown which corresponds to the reconstructed target.

one would obtain if the individual source signals of the mixture were available *a priori*, has been proposed as the computational goal of CASA (Wang, 2005). Figure 2·3 illustrates the concept of an ideal binary mask.

The aim of CASA systems is to identify these T-F regions in which the desired target is dominant. The systems utilize various cues, including harmonicity of the sources, fundamental frequency (F0) continuity, common onsets/offsets of energy across frequency bands, smooth transition of energy along time, and so on. Some methods use "mid-level" representations to explicitly encode information about such cues. For example, Cooke (1991) uses a "synchrony strand" representation that makes continuity in time-frequency explicit. Brown and Cooke (1994) extend the approach to create explicit TF "maps" of onset/offset activity, frequency transition, and periodicity. For monaural systems, the important cue is the fundamental frequency of sources. Several systems utilize this cue by constructing "correlogram" representations (Weintraub, 1985; Slaney and Lyon, 1990). Unlike these data-driven systems, approaches have also been proposed that are top-down and more generic in nature (Ellis, 1996; Godsmark and Brown, 1999). See Brown and Wang (2005) for a review of CASA approaches for sound separation.

An important assumption in CASA approaches is that the energy of a single utterance tends to be sparsely distributed, the implication being that different sources are disjoint in their spectro-temporal content (Yilmaz and Rickard, 2004). Indeed, this approach fails when mixtures are composed of sounds that are not spectro-temporally sparse. The approach can result in audible distortions when the composite signals overlap in time/frequency. In a study comparing CASA and ICA approaches, van der Kouwe et al. (2001) found that CASA performed well only on mixtures that exhibited well defined regions in the TF plane corresponding to the various sound sources. The performance for speech separation was best in conditions in which the interferer was tonal or locally narrowband. When there was substantial spectral overlap between target and interference, performance was poor. Despite the limitations, the idea of time-frequency masks and ideal binary masks continue to be dominant in CASA research.

2.3.2 Basis Decomposition Methods

We now briefly review a different class of approaches that we call basis decomposition methods. The main idea is that an observed data vector can be expressed as a linear combination of a set of "basis components." We are interested in methods that analyze time-frequency representations of audio to extract structure that can be used for applications like source separation in later stages. In other words, we focus on basis decomposition methods that analyze time-frequency representations as linear combinations of *source-dependent* components[3]. The intuition is that every source exhibits characteristic structure across frequency that can be captured by a finite set of components. Mathematically, the model can be written as

$$\mathbf{v}_t = \sum_{k=1}^{K} h_{kt}\mathbf{w}_k, \quad t = \{1, \ldots, T\}, \tag{2.1}$$

where $\mathbf{v}_t$ is the t-th frame of the observed spectrogram, K is the number of components, $\mathbf{w}_k$ is the k-th component vector and h_{kt} is the gain of the k-th component in the t-th frame. Writing the spectrogram as $F \times T$ matrix $\mathbf{V}$, basis components as $F \times K$ matrix the $\mathbf{W}$ ($[\mathbf{w}_1, \ldots, \mathbf{w}_K]$), and the gains as $K \times T$ matrix $\mathbf{H}$, the above formulation can be written as

$$\mathbf{V} = \mathbf{WH}. \tag{2.2}$$

Consider a simplistic example that illustrates this idea. The bottom-right panel of Figure 2·4 shows the spectrogram of a sound signal corresponding to two tones coming on and off intermittently. At various times during the signal, there is either silence, or one of the two tones is on, or both tones are on simultaneously. And yet, the entire signal can be represented as a linear combination of just two components, corresponding to the tones. The proportions with which the components combine indicates the extent to which they are present in the signal in each time frame. This is illustrated in the left and top panels of the figure. In this example, the two components have non-overlapping frequency

[3]Basis decomposition can also refer to source-independent time domain decompositions such as Fourier and Wavelet bases. The term is used in a restricted sense here. We should also point out that there are time-domain methods that extract source-dependent components (e.g., Jang and Lee, 2003) for monaural source separation but they will not be considered here.

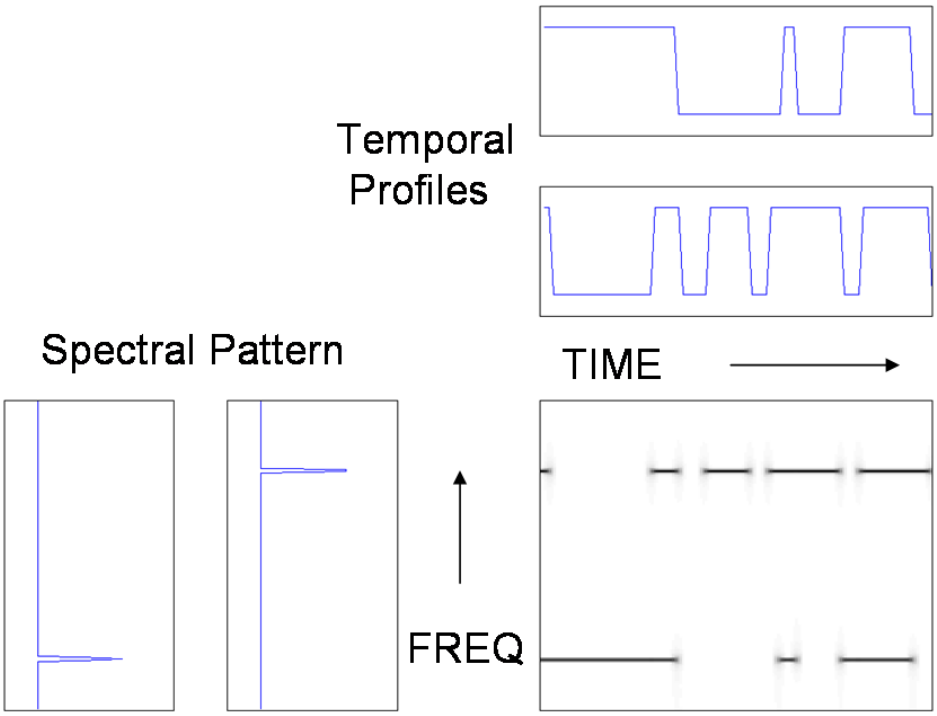


Figure 2·4: Illustration of Basis Decomposition. The bottom-right panel represents a mixture of two intermittent tones. The left panel indicates the two "basis components" corresponding to the frequencies of the two tones and the top-panel shows their time profiles. Non-negative Matrix Factorization was used to derive the spectral and temporal profiles (after Smaragdis, 2004).

content but this need not be true in general. In the context of source separation, the idea is that one could "learn" these components for every source present in a mixture signal (from clean training data) and use this information for source separation. If the components characterize the sources well, the separation quality should be high.

Several methods can be used for estimating the components to be used as bases. The formulation of equation (2.2) points to matrix factorization methods such as Principal Component Analysis (PCA) and Independent Component Analysis (ICA)[4]. The trouble with such standard methods is that the resulting matrices corresponding to the components and gains are real valued with both positive and negative entries. Therefore, components add and can cancel each other to approximate the input. For example, bases components extracted by PCA, shown in Figure 2·5 contain both positive and negative values, and the resulting approximation of the input can contain negative entries. However, the entries of spectrogram represent energies and thus should have only positive entries.

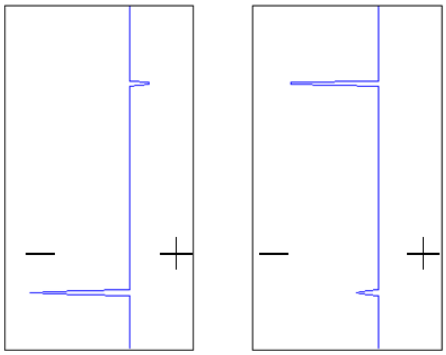


Figure 2·5: Two PCA basis components extracted from the spectrogram shown in Figure 2·4. Notice that the components have positive and negative values, which is hard to interpret in the context of spectrograms (energy cannot be negative). The ordinate represents frequency.

Non-negative Matrix Factorization (NMF; see section 3.3.4) was introduced by Lee and

[4]ICA uses a different formulation in which each row of the input $\mathbf{V}$ corresponds to time samples of the mixture signal from one sensor, instead of being a frequency index in a TF representation. ICA works well only when there are more sensors than sources. However, it has been extended for monaural sound separation recently as Independent Subspace Analysis, (see Casey and Westner, 2000)

Seung (1999) to explicitly enforce non-negativity constraints on all the entries of factored matrices. Researchers have since used this method to model acoustic sources and for source separation with good results (Smaragdis and Brown, 2003; Smaragdis, 2004; Virtanen, 2007). Recently, there has been a lot of interest in this approach and there have been several studies regarding its applicability in modeling acoustic signals (Virtanen, 2006; O'Grady, 2007). Despite its wide use, a weakness of NMF is the lack of theoretical motivation. Much of its appeal comes from its empirical success in learning meaningful components (Hoyer, 2004, pp. 1459), but there is no theoretical justification for why it works in separation (Virtanen, 2006, pp. 28). However, NMF based approaches have been widely used in various machine learning applications, including audio source separation, and research interest in this field continues to grow.

2.4 Spectrograms as Histograms - A Generative Model

The framework proposed in this thesis is based on a different approach to modeling spectrograms. We follow the basis decomposition approach and wish to learn characteristic components for acoustic sources that capture distinctive frequency structures. This is similar to NMF but it overcomes a significant limitation of NMF - the lack of a statistical generative model. We hypothesize a statistical model for how each spectral vector is generated and the framework attempts to characterize the underlying generative random process. In this section, we describe this generative model and set the stage for the rest of the thesis.

The value of a particular time-frequency bin in a spectrogram represents the amount of acoustic energy in the signal at the particular time frame and frequency band. We can thus consider this value as a count – a value which signifies the number of "energy quanta" observed at that particular bin. Consider the hypothesized process which generates a spectral vector. It is generated as a result of multiple draws from a random process. A given draw corresponds to an observation of one "unit energy quantum" at one of the F frequency bins. The process is repeated multiple times and the number of energy quanta observed in each bin is noted. This histogram of results corresponds to the observed spectral

vector. The total energy of the spectral vector corresponds to the total number of draws that generated it. Figure 2·6 illustrates the approach.

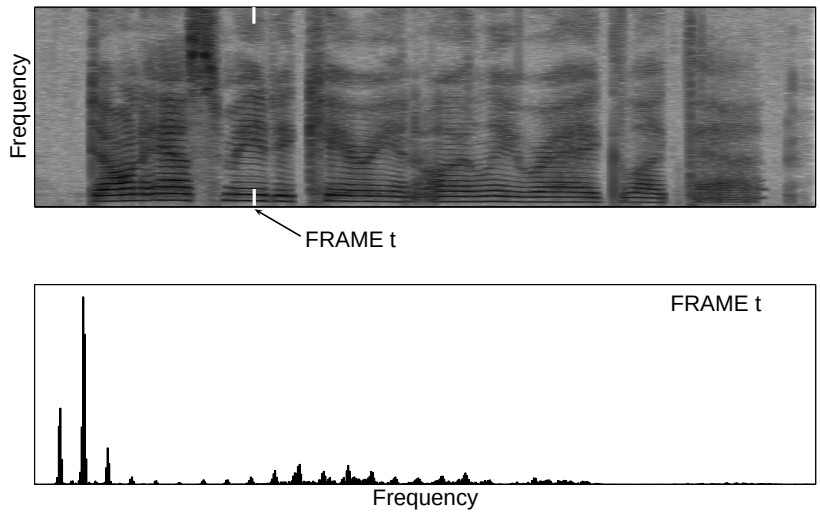


Figure 2·6: Illustration of the "spectrogram as a histogram" perspective. Each spectral vector is thought of as a histogram of multiple draws from an underlying random process.

In this approach, we can model a mixture spectrogram as the histogram of draws from multiple random processes, one each for every source present in the mixture. We show that the perspective allows us to model and reconstruct entire spectrograms for every underlying source rather than building partial spectral descriptions (as is done in the binary mask approach).

Before we proceed, we point out the applicability of the framework (to be proposed in future sections) to time-frequency representations that are explicitly modeled as probability distributions. For example, Loughlin et al. (1994) present a method to construct a joint time-frequency distribution to represent acoustic signals. For such representations, the framework analyzes the given input instead of modeling it as a histogram.

Chapter 3

Latent Variable Decomposition: A Probabilistic Framework

Although to penetrate into the intimate mysteries of nature and thence to learn the true causes of phenomena is not allowed to us, nevertheless it can happen that a certain fictive hypothesis may suffice for explaining many phenomena.

Leonhard Euler, 1748

3.1 Introduction

This chapter introduces the probabilistic framework that forms the basis of this thesis. As mentioned previously, spectrograms are modeled as histograms of multiple draws of frequency bin indices from an underlying random process. This allows one to develop a framework grounded in sound statistical principles. The core idea is that the random process that generates a particular spectral vector is modeled by a set of latent or hidden distributions that are characteristic of the source. These latent distributions combine in different proportions to generate different spectral vectors for a given source. The assumption is that these latent distributions capture spectral structure that is characteristic to the source and not to the individual spectral vectors. With this framework, one can utilize learned latent distributions from a set of sources for applications such as source separation and denoising. The fact that the framework is based on a probabilistic foundation allows us to use statistical techniques for parameter estimation. It also makes the framework more amenable to principled extensions and improvements, one of which will be considered in

the next chapter.

The chapter is organized as follows. Section 3.2 presents background about latent variables and latent class models. Latent variables and the concept of conditional independence, which underlie the proposed framework, are discussed. The framework is then proposed in Section 3.3, along with the theory and derivation of inference algorithms. A geometric interpretation of the model is presented in Section 3.4 that provides insight about the workings of the framework. The theory is presented with reference to generic *feature counts*. The framework is general and can be applied to different kinds of data, including images and word-counts data. Section 3.5 shows how the framework can be used for source separation and other acoustic processing tasks. The chapter ends with discussion and conclusions in Section 3.6.

3.2 Background: Latent Variables and Latent Class Models

Latent variables are widely used to understand and explain observed data in the areas of social and behavioral sciences and psychology. Consider the following sentence, used as an example by Borsboom et al. (2003): "Einstein would not have been able to come up with his $e = mc^2$ had he not possessed such an extraordinary intelligence." This sentence relates observable behavior (Einstein's writing $e = mc^2$) to an unobservable attribute (his extraordinary intelligence), and it does so by assigning to the unobservable attribute a causal role in bringing about Einstein's behavior. In psychology, constructs like this often play an important role to scientific theses. Similar situations arise in social science when scientists wish to understand *attitudes* of a population of individuals by observing their responses in a questionnaire. Problems like these can be approached by modeling the *observed actions* as manifest variables and the *hidden attitudes* as latent variables.

One can have a variety of models that employ latent variables. A simple example is as follows:

$$x = u + \delta, \tag{3.1}$$

where x is the manifest variable, u is the latent variable and δ represents measurement

error. The latent variable models that are of relevance to this work are more complex and are referred to as *Latent Class Models*.

3.2.1 Latent Class Models

According to Borsboom et al. (2003), the conceptual framework of latent variable analysis as discussed in this section can be traced back to the work of Spearman (1904), who developed factor-analytic models for continuous variables in the context of intelligence testing. The basic statistical idea of latent variable analysis is simple. If a latent variable underlies a number of observed variables, then the observed variables conditioned on that latent variable should be statistically independent. This is called the *principle of local independence*. The intuition behind this idea is that the common cause of a phenomenon should factor out observed correlations. Suppes and Zanotti (1981) call this principle the *common cause criterion*. For example, if it was found that barometric pressure and temperature were both dropping at the same time, one would look for a common dynamical cause within the theory of meteorology. Similarly, if one found that headaches and fever were positively correlated, he/she would look for a common cause instead of considering one as a cause of the other. Following Spearman's work, this paradigm developed in the 20th century. Models that assume the principle of local independence and employ discrete variables for both latent and observed variables are known as latent class models (Green Jr., 1952; Lazarsfeld and Henry, 1968; Goodman, 1974).

To illustrate the model, consider an example used by Lazarsfeld and Henry (1968). Let us suppose that a survey was conducted about the readership of two magazines x_1 and x_2, and 1000 people responded. The results of the survey are shown in Table 3.1.

	Read x_1	No x_1	Totals
Read x_2	260	240	500
No x_2	140	360	500
Totals	400	600	1000

Table 3.1: Illustrative example for latent class models - readership of magazines x_1 and x_2

From the table, it is easy to see that there is some association between the readerships of magazines x_1 and x_2. A simple indicator is the fact that readers of x_1 tend to read x_2 (260) more often than non-readers of x_1 (240). The magazines have some common appeal to to readers, though they are quite different in the readership.

Now, suppose that additional data on the 1000 respondents are available that indicate whether each individual has obtained higher education or not. The data can then be divided into two groups as shown in Table 3.2.

	High-Ed			Low-Ed		
	Read x_1	No x_1	Totals	Read x_1	No x_1	Totals
Read x_2	240	60	300	20	80	100
No x_2	160	40	200	80	320	400
Totals	400	100	500	100	400	500

Table 3.2: Readership of x_1 and x_2, given education

In the subgroups corresponding to different levels of education, there is no association between the readership. For example, among the High-Ed respondents, 4/5 read magazine x_1 irrespective of whether they read x_2 ($240/300 = 4/5 = 160/200$). Since there is no association between x_1 and x_2 when education is considered, one can say that education *explains* the observed association between the magazines. The observed relation between x_1 and x_2 was due to their common appeal to higher educated people.

Now, the same data can be viewed in terms of probabilities by normalizing all the entries. For example, the probability that a person would read both x_1 and x_2 is equal to $260/1000 = 0.26$. Let us represent the readership of magazines by random variables x_1 and x_2 which take two possible values 0 and 1. Let a value 1 imply that the person reads the magazine while a value of 0 implies that he/she does not. Similarly, let us represent the status of education by a dichotomous random variable z, where a value of 1 implies High-Ed and a value of 0 implies Low-Ed. Then, tables 3.1 and 3.2 can be written in terms of probabilities as shown in tables 3.3 and 3.4 respectively. The observed data can be viewed as histograms of repeated draws from these underlying probability distributions.

	$x_1 = 1$	$x_1 = 0$	
$x_2 = 1$	$P(x_1, x_2) = 0.26$	$P(x_1, x_2) = 0.24$	$P(x_2) = 0.5$
$x_2 = 0$	$P(x_1, x_2) = 0.14$	$P(x_1, x_2) = 0.36$	$P(x_2) = 0.5$
	$P(x_1) = 0.4$	$P(x_1) = 0.6$	1

Table 3.3: Illustrative example for latent class models - probabilities of random variables x_1 and x_2.

	$P(z = 1) = 0.5$		
	$x_1 = 1$	$x_1 = 0$	
$x_2 = 1$	$P(x_1, x_2\|z) = 0.48$	$P(x_1, x_2\|z) = 0.12$	$P(x_2\|z) = 0.6$
$x_2 = 0$	$P(x_1, x_2\|z) = 0.32$	$P(x_1, x_2\|z) = 0.08$	$P(x_2\|z) = 0.4$
	$P(x_1\|z) = 0.8$	$P(x_1\|z) = 0.2$	1

	$P(z = 0) = 0.5$		
	$x_1 = 1$	$x_1 = 0$	
$x_2 = 1$	$P(x_1, x_2\|z) = 0.04$	$P(x_1, x_2\|z) = 0.16$	$P(x_2\|z) = 0.2$
$x_2 = 0$	$P(x_1, x_2\|z) = 0.16$	$P(x_1, x_2\|z) = 0.64$	$P(x_2\|z) = 0.8$
	$P(x_1\|z) = 0.2$	$P(x_1\|z) = 0.8$	1

Table 3.4: Probabilities of random variables x_1 and x_2, given variable z.

Table 3.3 lists both the joint probabilities $P(x_1, x_2)$ and marginal probabilities $P(x_1)$, $P(x_2)$, where the marginals are given by

$$P(x_1) = P(x_1, x_2 = 0) + P(x_1, x_2 = 1),$$

$$P(x_2) = P(x_1 = 0, x_2) + P(x_1 = 1, x_2).$$

If we account for the random variable z corresponding to the education attribute that "explains" the data (as shown in Table 3.4), we also observe the following relation

$$P(x_1, x_2|z) = P(x_1|z) \times P(x_2|z). \tag{3.2}$$

The above relation implies that the random variables x_1 and x_2 are *statistically independent* if they are conditioned on random variable z. This relation enabled us to say that the concept corresponding to the variable z (education) explained the observed associations.

The observed joined probability can be written as

$$P(x_1, x_2) = \sum_{z \in \{0,1\}} P(z)P(x_1|z)P(x_2|z). \tag{3.3}$$

Equations (3.2) and (3.3), which correspond to the *principle of local independence*, define a latent class model. But the underlying variable z which renders the observed variables independent corresponds to a hidden or latent concept that is not directly observable, unlike the "education" attribute used in this example. Data about such additional attributes are rarely available and even when available, are not sufficient to explain the observed associations. The intuition behind latent class models is to explain the associations by invoking a hidden variable. In the general case, we can have multiple variables (say K) $x_1, x_2, \ldots, x_K$ and each variable, instead of taking two values, could take multiple values. In its general form, latent class model expresses a K-dimensional distribution as a mixture where each component of the mixture is a product of one-dimensional marginal distributions. Mathematically, we can write it as

$$P(\mathbf{x}) = \sum_z P(z) \prod_{j=1}^{K} P(x_j|z), \tag{3.4}$$

where $P(\mathbf{x})$ is a K-dimensional distribution of the random variable $\mathbf{x} = x_1, x_2, \ldots, x_K$. Mixture components are indexed by the latent variable z and $P(x_j|z)$ are one-dimensional marginal distributions. Given counts of multiple draws from $P(\mathbf{x})$, the aim is to estimate the parameters of the model $P(z)$ and $P(x_j|z)$, $j \in \{1, 2, \ldots, K\}$.

3.2.2 Latent Class Models as Matrix Decomposition

Consider a latent class model in two variables x_1 and x_2. Let x_1 and x_2 be multinomial variables, where x_1 can take one out of a set of M values in a given draw and x_2 can take one out of a set of N values. A draw can be thought of as rolling dice two with M and N faces, respectively. Observed data can be represented as a matrix $\mathbf{V}$, where the mn-th element V_{mn} represents the number of draws in which x_1 took a value of m *and* x_2 took

a value of n. Let $\mathbf{P}$ represent the matrix of normalized values of $\mathbf{V}$. In other words, $\mathbf{P}$ represents the underlying distribution $P(\mathbf{x})$, where $P_{mn} = P(x_1 = m, x_2 = n)$. Consider a latent variable z that can take values from the set $\{1, 2, \ldots, K\}$.

The latent class model expresses the joint distribution of x_1 and x_2 as

$$P(x_1, x_2) = \sum_{z \in \{1,\ldots,K\}} P(z)P(x_1|z)P(x_2|z). \tag{3.5}$$

We can view the above relation from the perspective of linear algebra. Let us represent the parameters $P(x_1|z)$, $P(x_2|z)$ and $P(z)$ as entries of matrices $\mathbf{W}$, $\mathbf{H}$ and $\mathbf{S}$ as follows:

- $\mathbf{W}$ is a $M \times K$ matrix, where the entry in m-th row and k-th column corresponds to the probability $P(x_1 = m|z = k)$.
- $\mathbf{H}$ is a $K \times N$ matrix, where the entry in k-th row and n-th column corresponds to the probability $P(x_2 = n|z = k)$.
- $\mathbf{S}$ is a $K \times K$ diagonal matrix, where the k-th entry corresponds to the mixture weight $P(z = k)$.

With this matrix notation, we can write the relation of latent class model as follows:

$$\begin{aligned} P(x_1, x_2) &= \sum_{z \in \{1,\ldots,K\}} P(z)P(x_1|z)P(x_2|z) \\ P_{x_1 x_2} &= \sum_{z=1}^{K} W_{x_1 z} S_{zz} H_{z x_2} \\ \mathbf{P} &= \mathbf{WSH}. \end{aligned} \tag{3.6}$$

Figure 3·1 illustrates the latent class model computation using a schematic. Thus, using a latent class model is equivalent to performing a *matrix decomposition*.

With this background, we are ready to introduce the general framework for latent variable decomposition.

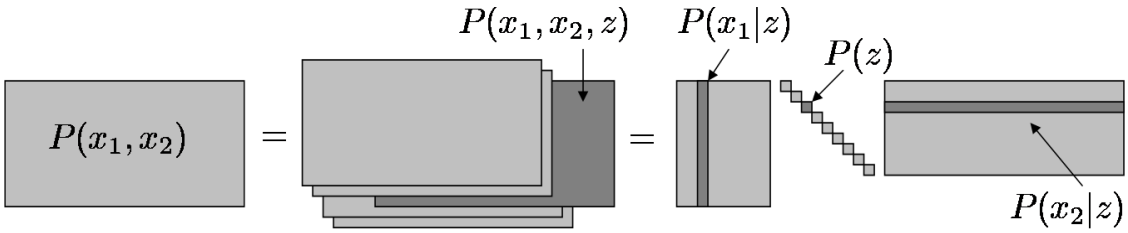


Figure 3·1: Illustration of the Latent Class Model computation. It is equivalent to a matrix decomposition

3.2.3 Probabilistic Latent Component Analysis (PLCA)

Before we present the framework, we first briefly describe Probabilistic Latent Component Analysis (Smaragdis and Raj, 2007; Smaragdis et al., 2006), which is equivalent to latent class models.

Consider a magnitude spectrogram $\mathbf{V}$ of a given sound snippet. Let the dimensions of the matrix be $F \times T$ (i.e., there are F frequency indices and T time frames). As described in Chapter 2, $\mathbf{V}$ can be thought of as a histogram of frequency localized *sound atoms*. The entry in each time-frequency bin V_{ft} describes how much acoustic energy we have at the particular frequency and time-frame. Let the random variable f represent the frequency index and t represent the time-frame. PLCA allows us to characterize the joint distribution $P(f,t)$ as

$$P(f,t) = \sum_z P(z)P(f|z)P(t|z). \tag{3.7}$$

As we showed in equation (3.6) with Latent Class Models, we can write this equation in matrix form as

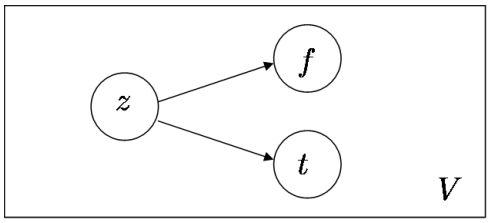
$$\mathbf{P} = \mathbf{WSH}, \tag{3.8}$$

where $F \times T$ matrix $\mathbf{P}$ represents the two-dimensional distribution $P(f,t)$, $\mathbf{W}$ is an $F \times K$ matrix with the f-th entry of the z-th column representing $P(f|z)$, $\mathbf{S}$ is an $K \times K$ diagonal matrix where the z-th diagonal element represents $P(z)$, and $\mathbf{H}$ is an $K \times T$ matrix with the t-th element of the z-th row representing $P(t|z)$. Random variables corresponding to both dimensions are thought of as features and are treated symmetrically. The generative

process for the model is as follows:

- Choose a value for latent variable z according to the distribution $P(z)$,
- Choose a value for f based on $P(f|z)$ and a value for t based on the distribution $P(t|z)$.
- Repeat the above two steps V times, where $V = \sum_{f,t} V_{ft}$ (i.e., the total number of energy "quanta" observed).

Figure 3·2 shows the graphical model for this generative process.



$$P(f,t) = \sum_z P(z)P(f|z)P(t|z)$$

Figure 3·2: Graphical model for two-dimensional latent class model. Circles represent variables, a box surrounding them indicates how many times they should be drawn and arrows indicate statistical dependence. z represents the hidden variable, f and t are the features drawn in the two dimensions in a given draw, and V is the total number of draws.

The objective of the analysis is to evaluate the underlying time-frequency structure of the given sound snippet by characterizing the generative distribution. This is done by estimating the parameters on the right hand side of equation (3.7) from the observed $P(f,t)$. We can accomplish this by using the Expectation-Maximization algorithm (Dempster et al., 1977; McLachlan and Krishnan, 1997; Neal and Hinton, 1998). The algorithm contains two steps - expectation and maximization - which are alternated in an iterative manner until convergence. All parameters are initialized to random values before starting the first iteration. In the expectation step, we estimate the "contribution" of the latent variable z as

$$P(z|f,t) = \frac{P(z)P(f|z)P(t|z)}{\sum_z P(z)P(f|z)P(t|z)}. \tag{3.9}$$

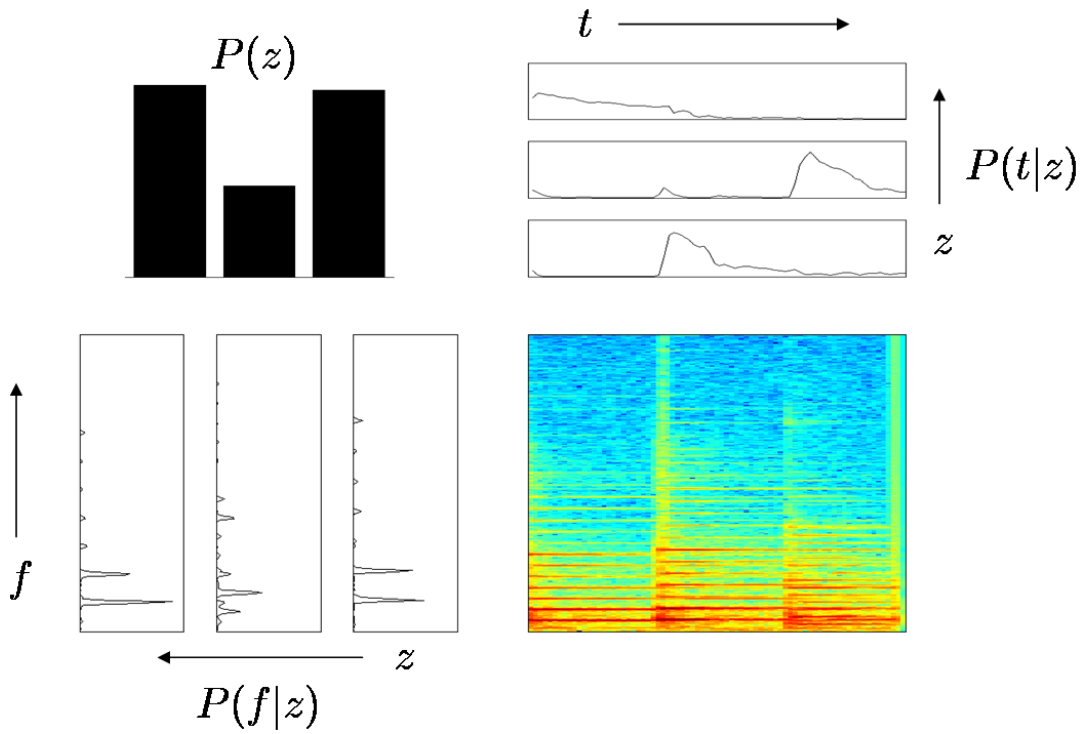


Figure 3·3: Illustration of PLCA applied on a spectrogram of three piano notes (bottom right). The top-right panel displays the extracted time marginals and the bottom-left panel shows the extracted frequency marginals. The top-left plot shows the mixture weights $P(z)$. Notice that the frequency marginals describe the spectra of the notes while the time-marginals describe their energy as a function of time.

In the maximization step, we re-estimate the marginals and the mixture weights using the above weighting to obtain a new and more accurate estimate:

$$P(z) = \frac{\sum_f \sum_t V_{ft} P(z|f,t)}{\sum_z \sum_f \sum_t V_{ft} P(z|f,t)}, \tag{3.10}$$

$$P(f|z) = \frac{\sum_t V_{ft} P(z|f,t)}{\sum_f \sum_t V_{ft} P(z|f,t)}, \tag{3.11}$$

$$P(t|z) = \frac{\sum_f V_{ft} P(z|f,t)}{\sum_f \sum_t V_{ft} P(z|f,t)}. \tag{3.12}$$

Figure 3·3 shows an example where PLCA was applied on an audio sample corresponding to three piano notes. The latent variable was allowed to take three values and the extracted *frequency components* correspond to the spectra of the three notes present in the sample.

3.3 Latent Variable Decomposition: Framework

We have seen in the previous section that Latent Class Models and PLCA are equivalent. For acoustic data in the form of a $F \times T$ matrix $\mathbf{V}$, the models decompose the distribution $P(f,t)$ symmetrically by considering both the f and t dimensions as features. Instead of a symmetrical decomposition of PLCA, one can have a different decomposition where the two dimensions treated differently[5]:

$$P(f,t) = P(t) \sum_z P(f|z) P(z|t), \tag{3.13}$$

or

$$\mathbf{P} = \mathbf{WHS} \tag{3.14}$$

in matrix form, where $\mathbf{P}$ represents the two-dimensional distribution $P(f,t)$, $\mathbf{W}$ is an $F \times K$ matrix with the f-th entry of the z-th column representing $P(f|z)$, $\mathbf{H}$ is an $K \times T$ matrix with the z-th entry of the t-th column representing $P(z|t)$, and $\mathbf{S}$ is an $T \times T$ diagonal matrix with the t-th diagonal element equal to $P(t)$. Figure 3·4 shows the graphical model

[5]Instead, we can use $P(f,t) = P(f) \sum_z P(t|z) P(z|f)$ (or in matrix form: $\mathbf{P}_{F\times T} = \mathbf{S}_{F\times F} \mathbf{W}_{F\times K} \mathbf{H}_{K\times T}$, where subscripts denote matrix sizes and $\mathbf{S}$ is a diagonal matrix). This is numerically equivalent to using equation (3.13) or (3.14) with the input dimensions transposed.

for this factorization. Hofmann (2001), motivated by applications in semantic analysis of text corpora, introduced this model as Probabilistic Latent Semantic Analysis.

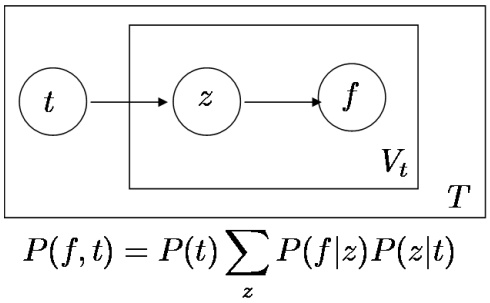


Figure 3·4: Graphical model for alternative decomposition of the two-dimensional latent class model. Circles represent variables, a box surrounding them indicates how many times they should be drawn and arrows indicate statistical dependence. Each column vector of the data matrix $\mathbf{V}$ is considered a separate data vector. z represents the hidden variable, f is the feature drawn in a given draw, V_t is the total number of draws for the t-th data vector, and T is the total number of data vectors.

In this section, we present a specific case of the decomposition of the latent class model as defined in equations (3.13) and (3.14). It was originally proposed by Raj and Smaragdis (2005) in the context of separating talkers from single-channel acoustic recordings. Each data vector is considered independently and we model T one-dimensional distributions $P_t(f)$ instead of the two-dimensional distribution $P(f,t)$. Treating the two dimensions differently allows the resulting decomposition to be interpreted easily as "components" corresponding to underlying structure of the data and their "mixture weights." This model will form the basic computational framework of this thesis. Henceforth, the terms *latent variable model* and *latent variable decomposition* will refer specifically to this model, unless explicitly stated otherwise.

3.3.1 Latent Variable Model

Consider a random process characterized by the probability $P(f)$ of drawing a feature unit f in a given draw. Let the random variable f take values from the set $\{1, 2, \ldots, F\}$. Let us assume that $P(f)$ is unknown and what one can observe instead is the result of

multiple draws from the underlying process. In other words, we observe feature *counts*, or the number of times feature f is observed after repeated draws. We can approximate the *generative distribution* $P(f)$ by using the normalized set of counts.

Now suppose we also know that $P(f)$ is comprised of K *hidden distributions* or *latent factors*. The observation in a given draw might come from any one of the K distributions. The distributions are selected according to their relative probabilities, which remain constant across draws in a given experiment. We are allowed to run multiple experiments and observe feature counts for each experiment. The probabilities of the hidden distributions vary from experiment to experiment. Our task is to characterize these hidden distributions.

Let us define $P(f|z)$ as the probability of observing feature f conditioned on a *latent variable* z, where z represents the index defining which hidden distribution is being considered. The probability of picking the z-th distribution in the t-th experiment can be represented by $P_t(z)$. We can now formally write the model as

$$P_t(f) = \sum_z P(f|z)P_t(z), \tag{3.15}$$

where $P_t(f)$ gives the overall probability of observing feature f in the t-th experiment. Here, the multinomial distributions $\{P(f|z)\}$ can be thought of as *basis components* that are characteristic to all experiments. $P_t(z)$ are mixture weights that signify the contribution of $P(f|z)$ towards $P_t(f)$. The subscript t indicates that mixture weights change from experiment to experiment.

The random process generating counts in the t-th experiment can be summarized as

1. Pick a latent variable z with probability $P_t(z)$.
2. Pick feature f from the multinomial distribution $P(f|z)$.
3. Repeat the above two steps V times,

where V is total number of draws in experiment t. Figure 3·5 shows the graphical model depicting the process. This model is equivalent to using the latent class model (or equivalently PLCA) on the result of every experiment independently.

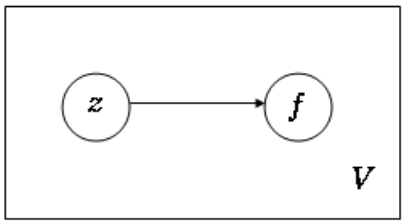


Figure 3·5: Graphical model for the random process underlying the generation of a data vector. Circles represent variables, the surrounding box represents repeated draws and the arrow represents dependence. z is the hidden variable, f is the feature drawn, and V is the total number of draws.

3.3.2 Parameter Estimation

Let V_{ft} represent the feature count of feature f in the t-th experiment. Given feature counts V_{ft}, we wish to estimate parameters $P(f|z)$ and $P_t(z)$. Let Λ represent the set of parameters, i.e. $\Lambda = \{P(f|z), P_t(z)\}$. We use a maximum likelihood formulation of the problem. The log-likelihood of observing the obtained f counts across all T experiments is given by

$$\mathcal{P} = \sum_t \sum_f V_{ft} \log P_t(f). \tag{3.16}$$

The maximum likelihood method estimates parameters such that this log-likelihood is maximized. The standard procedure for maximum likelihood estimation in latent variable models is the Expectation Maximization (EM) algorithm. EM alternates two steps: (1) an expectation (E) step where the *a posteriori* probabilities of the latent variables are computed based on the current estimates of the parameters, and (2) a maximization (M) step, where parameters are updated such that the expected complete data log-likelihood[6] is maximized.

For the E-step, we obtain the *a posteriori* probability for the latent variable as

$$P_t(z|f) = \frac{P_t(z)P(f|z)}{\sum_z P_t(z)P(f|z)}. \tag{3.17}$$

[6]The term "complete data log-likelihood" refers to the log-likelihood calculated by considering the likelihood of the obtained counts of both observed variable f and the latent variable z. It is given by $P(\bar{f}, \bar{z})$, where $\bar{f}$ and $\bar{z}$ represent the sets of all observations of f and z in the draws that generated all data vectors. The expectation is over the distribution $P(\bar{z}|\bar{f}; \Lambda)$.

In the M-step, we maximize the expected complete data log-likelihood. The expected log-likelihood can be written as

$$\mathcal{L} = E_{\bar{z}|\bar{f};\Lambda} \log P(\bar{f}, \bar{z}), \tag{3.18}$$

where $\bar{f}$ and $\bar{z}$ represent the set of all observations of f and z in the draws that generated all data vectors. The complete data likelihood can be written as

$$P(\bar{f}, \bar{z}) = \prod_{j,t} P_t(f_j, z_j) = \prod_{j,t} P_t(z_j) P(f_j|z_j), \tag{3.19}$$

where f_j and z_j are the values of variables f and z in the j-th draw. Hence, we can write the function $\mathcal{L}$ as (ignoring the constant terms)

$$\begin{aligned}
\mathcal{L} &= E_{\bar{z}|\bar{f};\Lambda} \log \prod_{j,t} P_t(f_j, z_j) \\
&= E_{\bar{z}|\bar{f};\Lambda} \sum_{j,t} \log P_t(f_j, z_j) \\
&= \sum_{j,t} E_{z_j|f_j;\Lambda} \log P_t(f_j, z_j) \\
&= \sum_{j,t} E_{z_j|f_j;\Lambda} \log P_t(z_j) + \sum_{j,t} E_{z_j|f_j;\Lambda} \log P(f_j|z_j) \\
&= \sum_{j,t} \sum_{z} P(z|f_j) \log P_t(z) + \sum_{j,t} \sum_{z} P(z|f_j) \log P(f_j|z)
\end{aligned} \tag{3.20}$$

In the above equation, we can change the summation over draws j to a summation over features f by accounting for how many times f was observed, i.e. the f-th entry in the observed data vector[7]. The expected log-likelihood can now be written as

$$\mathcal{L} = \sum_{t} \sum_{f} \gamma V_{ft} \sum_{z} P_t(z|f) \log P_t(z) + \sum_{t} \sum_{f} \gamma V_{ft} \sum_{z} P_t(z|f) \log P(f|z). \tag{3.21}$$

We have additional constraints on the parameters $P_t(z)$ and $P(f|z)$ as they represent probability distributions, given by $\sum_z P_t(z) = 1$ and $\sum_f P(f|z) = 1$. In order to take care of these normalization constraints, the above equation must be augmented by appropriate

[7]Since observed data is modeled as a histogram, entries should be integers. To account for this, we weight the data by an unknown scaling factor γ.

Lagrange multipliers τ_t and ρ_z,

$$Q = \mathcal{L} + \sum_t \tau_t \Big(1 - \sum_z P_t(z)\Big) + \sum_z \rho_z \Big(1 - \sum_f P(f|z)\Big) \tag{3.22}$$

Maximization of Q with respect to $P_t(z)$ and $P(f|z)$ leads to the following sets of equations

$$\sum_f \gamma V_{ft} P_t(z|f) + \tau_t P_t(z) = 0, \tag{3.23}$$

$$\sum_t \gamma V_{ft} P_t(z|f) + \rho_z P(f|z) = 0. \tag{3.24}$$

After eliminating the Lagrange multipliers, we obtain the M-step re-estimation equations

$$P(f|z) = \frac{\sum_t V_{ft} P_t(z|f)}{\sum_f \sum_t V_{ft} P_t(z|f)}, \qquad P_t(z) = \frac{\sum_f V_{ft} P_t(z|f)}{\sum_z \sum_f V_{ft} P_t(z|f)}. \tag{3.25}$$

The E-step update is given by equation (3.17) and the M-step update is given by equations (3.25). The parameters $P(f|z)$ and $P_t(z)$ are randomly initialized and re-estimated using the above equations iteratively until a termination condition is met. The EM algorithm guarantees that the above multiplicative updates converge to a local optimum.

Figure 3·6 shows an example application of the latent variable model. The model was used to analyze handwritten digits from the USPS Handwritten Digits database[8]. Twenty five basis components were extracted by analyzing 1000 different instances for every digit. Each instance of a digit was given by the pixel intensities as a 16×16 matrix. We unwrapped each one as a 256-dimensional vector and represented the set of 1000 vectors as a 256×1000 matrix $\mathbf{V}$. The matrix $\mathbf{V}$ was used as the input to the algorithm. Figure 3·6 shows the extracted components for digit "2."

3.3.3 Latent Variable Model as Matrix Decomposition

We can write the model given by equation (3.15) in matrix form as $\mathbf{p}_t = \mathbf{W}\mathbf{h}_t$, where $\mathbf{p}_t$ is a column vector indicating $P_t(f)$, $\mathbf{h}_t$ is a column vector indicating $P_t(z)$, and $\mathbf{W}$ is

[8] from http://www.cs.toronto.edu/~roweis/data.html.

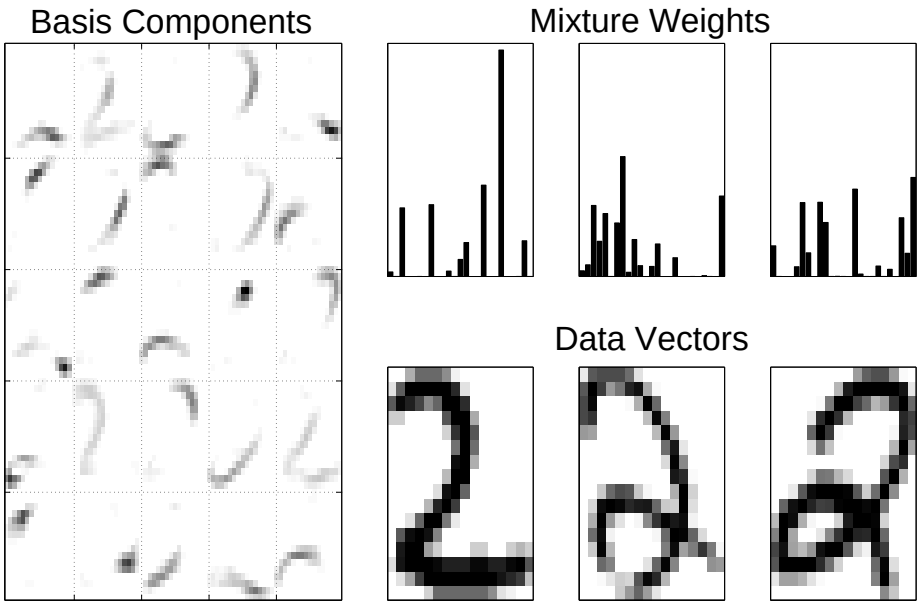


Figure 3·6: Latent Variable Model applied on the USPS Handwritten Digits database. Twenty-five basis components were learned from the data set and basis components extracted for the digit "2" are shown in the left panel. The basis components are shown in a 5×5 tile. They correspond to various hand-strokes (basis vectors) that could be added to obtain the digit "2." The three panels on top-right show the mixture proportions with which the basis components combine to approximate the input vectors (shown in the bottom three panels).

the $F \times K$ matrix with the (f, z)-th element corresponding to $P(f|z)$. Concatenating all column vectors $\mathbf{p}_t$ and $\mathbf{h}_t$ as matrices $\mathbf{P}$ and $\mathbf{H}$ respectively, one can write the model as

$$\mathbf{P} = \mathbf{WH}. \tag{3.26}$$

This formulation is similar to matrix decompositions such as PCA, ICA and NMF. We have additional constraints that the columns of $\mathbf{P}$, $\mathbf{W}$ and $\mathbf{H}$, being probability distributions, should be positive and sum to unity. Thus, the model is equivalent to a matrix decomposition which operates in the probability distribution space and is illustrated in Figure 3·7.

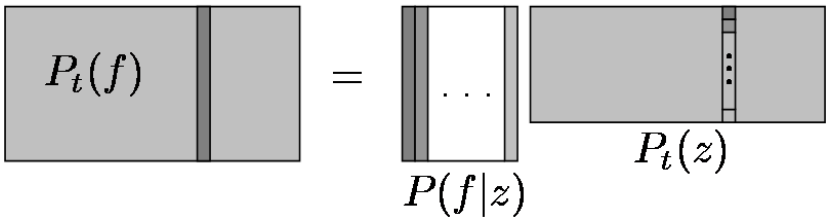


Figure 3·7: Illustration of the Latent Variable Model computation, which is equivalent to a matrix decomposition.

Furthermore, we want to clarify how the decomposition $\mathbf{P} = \mathbf{WH}$ relates to the data matrix $\mathbf{V}$. Let the matrix $\bar{\mathbf{V}}$ refer to the data in $\mathbf{V}$ with all the columns normalized. In other words, the t-th column of $\bar{\mathbf{V}}$, $\bar{\mathbf{v}}_t$, is the normalized version of $\mathbf{v}_t$, the t-th column of $\mathbf{V}$ (the entries of $\bar{\mathbf{v}}_t$ sum to unity). Let us refer to the normalized columns $\bar{\mathbf{v}}_t$ as data distributions. We first show that the maximum likelihood estimator for the parameters $P(f|z)$ and $P_t(z)$ attempts to minimize the Kullback-Leibler (KL) distance between the data distribution $\bar{\mathbf{v}}_t$ and the model approximation $P_t(f)$.

Maximum likelihood method estimates parameters such that the log-likelihood $\mathcal{P}$, given by equation (3.16), is maximized. We can rewrite it as

$$\mathcal{P} = \sum_t (\sum_f V_{ft}) \sum_f \frac{V_{ft}}{\sum_{f'} V_{f't}} \log P_t(f). \tag{3.27}$$

Representing the normalized data $V_{ft}/\sum_f V_{ft}$ by $\bar{V}_{ft}$, we can write the log-likelihood as

$$\begin{aligned}\mathcal{P} &= \sum_t (\sum_f V_{ft}) \sum_f \bar{V}_{ft} \log\Big(\frac{P_t(f)}{\bar{V}_{ft}}\Big) + \sum_t (\sum_f V_{ft}) \sum_f \bar{V}_{ft} \log \bar{V}_{ft} \\ &= \sum_t (\sum_f V_{ft}) \sum_f \bar{V}_{ft} \log\Big(\frac{P_t(f)}{\bar{V}_{ft}}\Big) + \text{constant term} \\ &= -\sum_t (\sum_f V_{ft}) D_{KL}(\bar{\mathbf{v}}_t || \mathbf{p}_t) + \text{constant term}, \end{aligned} \tag{3.28}$$

where

$$D_{KL}(\bar{\mathbf{v}}_t || \mathbf{p}_t) = \sum_f \bar{V}_{ft} \log \frac{\bar{V}_{ft}}{P_t(f)} \tag{3.29}$$

is the Kullback-Leibler distance between the data distribution $\bar{\mathbf{v}}_t$ and the model approximation $\mathbf{p}_t$. The term $\sum_t (\sum_f V_{ft}) \sum_f \bar{V}_{ft} \log \bar{V}_{ft}$ is a constant since it does not depend on $P_t(f)$. From equation (3.28), we can see that maximizing the log-likelihood $\mathcal{P}$ is equivalent to minimizing the sum of the KL distances $D_{KL}(\bar{\mathbf{v}}_t || \mathbf{p}_t)$, scaled by the total number of draws $\sum_f V_{ft}$.

In other words, the model attempts to find a matrix decomposition that approximates the data distributions $\bar{\mathbf{V}}$ as

$$\bar{\mathbf{V}} \approx \mathbf{P} = \mathbf{WH}, \tag{3.30}$$

where the approximation error is measured by the KL distance. Equivalently, the model attempts to approximate the data matrix $\mathbf{V}$ as

$$\mathbf{V} \approx \mathbf{WHG}, \tag{3.31}$$

where $\mathbf{G}$ is a $T \times T$ diagonal matrix with the t-th entry equal to $\sum_f V_{ft}$. Now, we can write the update equations (3.17) and (3.25) in matrix form. Writing the normalization steps separately, we have

$$\begin{aligned} W_{fk}^{new} &= W_{fk}^{old} \sum_t \frac{H_{kt} V_{ft}}{(W^{old} H)_{ft}}, \quad W_{fk}^{new} = \frac{W_{fk}^{new}}{\sum_f W_{fk}^{new}}, \qquad \text{and,} \\ H_{kt}^{new} &= H_{kt}^{old} \sum_f \frac{W_{fk} V_{ft}}{(W H^{old})_{ft}}, \quad H_{kt}^{new} = \frac{H_{kt}^{new}}{\sum_f V_{ft}}, \end{aligned} \tag{3.32}$$

where A_{ij} represents the ij-th entry of matrix $\mathbf{A}$.

3.3.4 Relation to Other Models

The latent variable model we presented is closely related to two techniques - Probabilistic Latent Semantic Analysis, and Non-negative Matrix Factorization. In this subsection, we briefly comment on how the model relates to these techniques.

Probabilistic Latent Semantic Analysis

Hofmann (2001) introduced Probabilistic Latent Semantic Analysis (PLSA), motivated by applications in semantic analysis of text corpora. The aim of the method is to identify contexts of word usage in documents without recourse to a dictionary or a thesaurus. This is not straightforward because of two kinds of words that occur in languages:

Polysems - words with multiple meanings, and

Synonyms - words with identical or similar meaning.

PLSA is largely influenced by *Latent Semantic Analysis* (LSA; Deerwester et al., 1990). The key idea of LSA is to map high-dimensional count vectors to a lower dimensional *latent semantic space*. By applying LSA on vector space representations of text documents, where every document in a corpus is represented by a vector of word-counts (Salton and McGill, 1983), one aims to represent semantic relations between words and/or documents in terms of their proximity in the semantic space. The technique stems from linear algebra and is based on a L_2-optimal approximation of matrices of word counts based on a Singular Value Decomposition (SVD). One starts with the standard SVD given by

$$\mathbf{V} = \mathbf{U}\mathbf{\Sigma}\mathbf{Y}^t, \tag{3.33}$$

where $\mathbf{V}$ is the term-document matrix of word counts, $\mathbf{U}$ and $\mathbf{Y}$ are matrices with orthonormal columns, and the diagonal matrix $\mathbf{\Sigma}$ contains the singular values of $\mathbf{V}$. The LSA approximation of $\mathbf{V}$ is computed by thresholding all but the largest K singular values

in $\mathbf{\Sigma}$ to zero. One might think of the rows of $\mathbf{U\Sigma}$ as defining coordinates for documents in the latent space. The hope is that terms having a common meaning and similar documents (even if they don't have terms in common) are roughly mapped to the same direction in the latent space.

As the name suggests, PLSA provides a probabilistic framework for LSA. Let $P(d)$ denote the probability that a word occurrence will be observed in a particular document d, $P(w|z)$ denote the class-conditional probability of word w conditioned on the unobserved class variable z, and $P(z|d)$ denote a document specific probability distribution over the latent variable space. PLSA defines a generative model for word/document co-occurrences by the following scheme:

1. select a document d with probability $P(d)$,
2. pick a latent class z with probability $P(z|d)$, and
3. generate a word w with probability $P(w|z)$.

One can now describe the joint word-document probability distribution as

$$P(d, w) = P(d) \sum_z P(w|z)P(z|d). \tag{3.34}$$

This equation is identical to equation (3.13) and corresponds to an alternative decomposition of the latent class model

$$P(d, w) = \sum_z P(z)P(d|z)P(w|z).$$

The PLSA model, thus, is a specific case of the general framework. The latent variable model introduced in the previous section corresponds to a simplified version of this model where document probabilities are not explicitly computed. The maximum likelihood estimates of the parameter $P(d)$ is the fraction of all observations that come from the d-th document. The estimates of $P(w|z)$ and $P(z|d)$ can be shown to be identical to the updates of the latent variable parameters derived in Section 3.3.2, where words w correspond

to features f and documents d corresponds to experiments indexed by t.

Non-negative Matrix Factorization

Non-negative Matrix Factorization (Paatero and Tapper, 1994; Lee and Seung, 1999) was introduced as a technique to find non-negative parts-based representation of non-negative data. Given an $F \times T$ matrix $\mathbf{V}$ where each column corresponds to a data vector, NMF approximates it as a product of non-negative matrices $\bar{\mathbf{W}}$ and $\bar{\mathbf{H}}$, i.e. $\mathbf{V} \approx \bar{\mathbf{W}}\bar{\mathbf{H}}$, where $\bar{\mathbf{W}}$ is a $F \times K$ matrix and $\bar{\mathbf{H}}$ is a $K \times T$ matrix. We use $\bar{\mathbf{W}}$ and $\bar{\mathbf{H}}$ to disambiguate the NMF decomposition matrices from the notation used in Section 3.3.3. The columns of $\bar{\mathbf{W}}$ can be thought of as *basis components* that are optimized for the linear approximation of $\mathbf{V}$. The non-negativity constraints make the representation purely additive (allowing no cancellations), in contrast to other linear representations such as Principal Components Analysis (PCA) and Independent Components Analysis (ICA).

The optimal choice of matrices $\bar{\mathbf{W}}$ and $\bar{\mathbf{H}}$ are defined by those non-negative matrices that minimize the reconstruction error between $\mathbf{V}$ and $\bar{\mathbf{W}}\bar{\mathbf{H}}$ using iterative update rules. Different error functions have been proposed which lead to different update rules (Lee and Seung, 1999, 2001). Shown below are multiplicative update rules derived by Lee and Seung (1999) using an error metric similar to the Kullback-Leibler divergence:

$$\begin{aligned} \bar{W}_{fk} &\leftarrow \bar{W}_{fk} \sum_t \frac{\bar{H}_{kt} V_{ft}}{(\bar{W}\bar{H})_{ft}}, \quad \bar{W}_{fk} = \frac{\bar{W}_{fk}}{\sum_f \bar{W}_{fk}}, \qquad \text{and,} \\ \bar{H}_{kt} &\leftarrow \bar{H}_{kt} \sum_f \frac{\bar{W}_{fk} V_{ft}}{(\bar{W}\bar{H})_{ft}}, \end{aligned} \tag{3.35}$$

where A_{ij} represents the i-th row and the j-th column of matrix $\mathbf{A}$. If one compares the above equations to the EM update rules for the latent variable model given by equations (3.32), it is easy to see that the update rules are *identical* if one lets

$$\bar{\mathbf{W}} = \mathbf{W}, \qquad \text{and} \qquad \bar{\mathbf{H}} = \mathbf{H}\mathbf{G}. \tag{3.36}$$

3.4 Latent Variable Decomposition - Geometrical Interpretation

The latent variable model as given by equation (3.15) expresses an F-dimensional distribution $P_t(f)$ as a mixture of K F-dimensional basis distributions $P(f|z)$. The aim of the model is to find $P(f|z)$ such that $P_t(f)$ best approximates the data distributions $\bar{\mathbf{v}}_t$. Being probability distributions, $P(f|z)$, $P_t(z)$ and $\bar{\mathbf{v}}_t$ are points in the $(F-1)$-dimensional simplex. In case of 3-dimensional distributions (a 3-dimensional input space), the generative distributions and basis components lie within the *Standard 2-Simplex* (the plane defined by points on each axis which are unit distance from the origin, see Figure 3·8) and hence are easy to visualize.

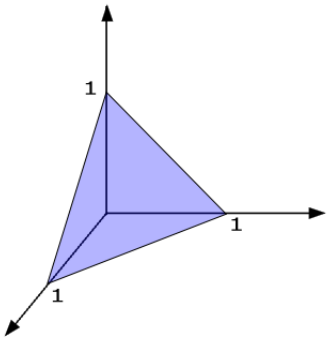


Figure 3·8: The triangle formed by points on each axis which are unit distance from the origin is called the *Standard 2-Simplex*. It is shown in the figure by the blue region. All triples corresponding to 3-dimensional multinomial distributions (so that the three numbers sum to unity) must lie within the Standard 2-Simplex. Similarly, n-tuples corresponding to a n-dimensional multinomial distribution lie within the Standard $(n-1)$ Simplex.

To understand and visualize the workings of the model, we created an artificial data set of 400 3-dimensional distributions and applied the latent variable model. The model expresses the generative distribution $P_t(f)$ as a linear combination of basis components $P(f|z)$ where the mixture weights $P_t(z)$ are positive and sum to unity. Geometrically, this implies that a given generative distribution is expressed as a point within the *convex hull* formed by the basis components. Since $P_t(f)$ is constrained to lie within the simplex defined

by $P(f|z)$, it can only model $\bar{\mathbf{v}}_t$ accurately if the latter also lies within the convex hull. Any $\bar{\mathbf{v}}_t$ that lies outside the convex hull is modeled with error. Thus, the objective of the model is to identify $P(f|z)$ such that they form a convex hull surrounding the data distributions $\bar{\mathbf{v}}_t$. This is illustrated in Figure 3·9 for 2 and 3 basis components.

Both the basis components and mixture weights encode information about the data set. Basis components, being the corners of the convex hull that encloses all the data points, encodes global characteristics about the data. The mixture weights, being associated with individual data points (experiments), encode local characteristics. The intuition is that the basis components correspond to characteristics of the random process that remain invariant during the generation of all the data points (all experiments).

We now consider two special cases of the decomposition that adds insight to its nature. Firstly, consider the case where we extract F basis components, i.e. $K = F$, corresponding to a *complete code*. One of the solutions corresponds to the case where the basis components are such that

$$P(f|z) = \begin{cases} 1 & \text{if } f = z \\ 0 & \text{otherwise,} \end{cases} \tag{3.37}$$

where $f \in \{1, \ldots, F\}$, $z \in \{1, \ldots, F\}$. In terms of the matrix notation used in Section 3.3.3, this implies that the basis component matrix $\mathbf{W}$ is given by the identity matrix $\mathbf{I}$. In this case, $\mathbf{h}_t$, the mixture weight vector corresponding to $P_t(z)$, is equal to the data distribution $\bar{\mathbf{v}}_t$, i.e. $\mathbf{H} = \bar{\mathbf{V}}$. In other words, the basis components correspond to the corners of the Standard $(F-1)$ simplex. Even though this corresponds to a perfect decomposition, it is not of any utility since the basis components do not provide any meaningful characterization of the data. They just represent the dimensions of the space in which the data lie. All the information about the data points is encoded by the mixture weights.

If we try to extract more basis components than the dimensionality of the input space F, we encounter the problem of indeterminacy. In such cases where we aim to extract an *overcomplete* set of basis components, there are multiple ways of expressing the data distri-

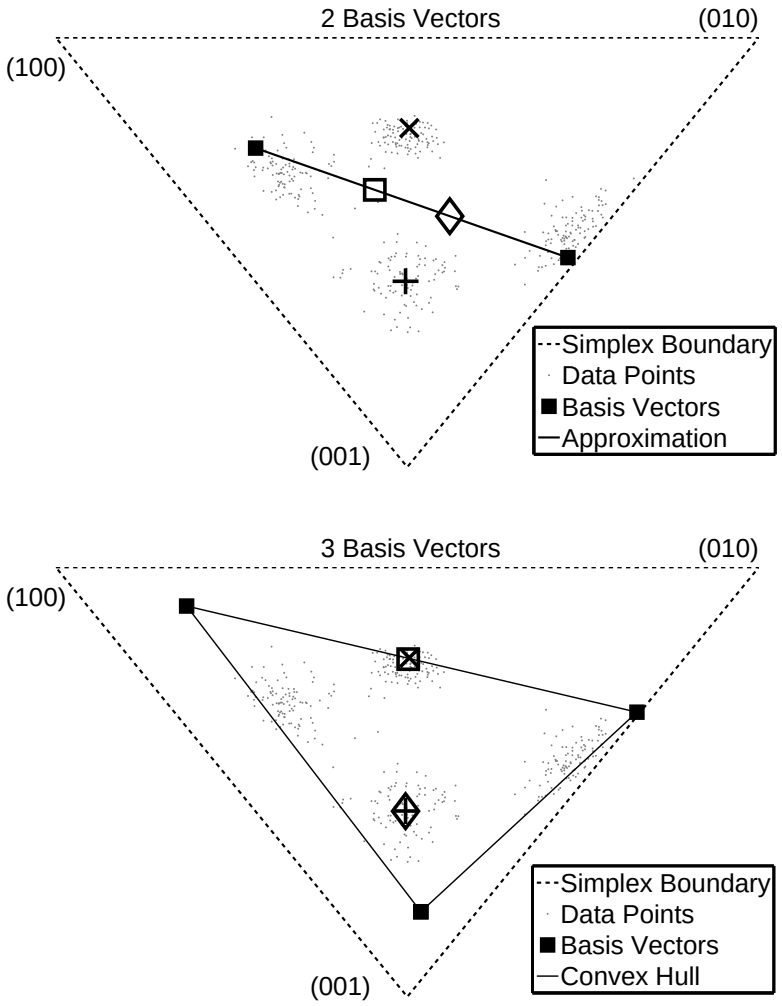


Figure 3·9: Illustration of the latent variable model on 3-dimensional distributions. Both panels show distributions represented within the *Standard 2-Simplex* given by $\{(001), (010), (100)\}$. Two Basis Components (top) and 3 Basis Components (bottom) extracted from 400 data points are shown. The model approximates data vectors as points lying on the line approximation (top) or within the convex hull (bottom) formed by the basis components. Also shown are two data points (marked as $+$ and $\times$) and their approximations by the model (shown by $\diamondsuit$ and $\square$). As one can see, the model gets more accurate as the number of basis components increases from a *compact code* of 2 basis components to a *complete code* of 3 basis components.

butions as linear combinations of the basis components. This implies that there are multiple feasible solutions that perfectly model the data. The second special case corresponds to the decomposition where we extract as many basis components as there are data points, i.e. $K = T$. One trivial solution when $K = T$ occurs when the basis components are the data distributions themselves. In matrix notation, this implies that the basis component matrix $\mathbf{W}$ is equal to $\bar{\mathbf{V}}$. The mixture weight matrix $\mathbf{H}$ is then given by the identity matrix. In this case, all the information about the data set is encoded by the basis components while the mixture weights contribute no information.

3.5 Latent Variable Framework for Source Separation

As mentioned in Chapter 2, the magnitude spectrogram of an acoustic signal can be treated as a histogram. Each time-frequency bin describes how much acoustic energy is present at the particular frequency and the particular time frame. Since the latent variable model is applicable to any data that can be considered as counts or histograms, we can apply it to analyze magnitude spectrograms. In this section, we show how the model can be used to extract frequency structure of all the sounds in the mixture and use the learned information to extract the contributions of each source to the mixture spectrogram.

Let us formally introduce the problem. Let $\mathbf{V}$ represent the magnitude spectrogram of a mixture sound signal. We would like to extract the magnitude spectrograms of each source present in the mixture. Let us assume that we know the number of sources present in the mixture and a set of training recordings is available for each source. Let $\mathbf{L}^s$ represent the magnitude spectrogram of the training data for the s-th source, where L^s_{ft} denotes the energy in frequency bin f at time frame t for source s. There are two stages in the separation algorithm. In the first stage, we learn the component multinomial distributions for each source from the training spectrograms. In the separation stage, these learned basis components are used to extract the contribution of the particular source to the mixture spectrogram.

3.5.1 Training Stage - Learning Parameters for Sources

In the learning stage, the component multinomial distributions denoted by $P_s(f|z)$ are learned for each source. The latent variable model is given by equation (3.15), which is reproduced below:

$$P_t(f) = \sum_z P_t(z)P_s(f|z).$$

$P_t(f)$ represents the normalized counts of the t-th frame of $\mathbf{L}^s$, i.e. it is the underlying generative distribution for the t-th time frame. We would like to characterize it as a mixture of component multinomials $P_s(f|z)$, each one weighted by a corresponding mixture weight $P_t(z)$. The subscript s in $P_s(f|z)$ indicates that these terms are specific to the source; the aim of this stage is to learn these component multinomials for each source.

The parameters $P_t(z)$ and $P_s(f|z)$ are initialized randomly and reestimated through iterations of equations (3.17) and (3.25), reproduced below.

$$\begin{aligned} P_t(z|f) &= \frac{P_t(z)P_s(f|z)}{\sum_z P_t(z)P_s(f|z)}, \\ P_s(f|z) &= \frac{\sum_t P_t(z|f)L^s_{ft}}{\sum_t \sum_f P_t(z|f)L^s_{ft}}, \\ P_t(z) &= \frac{\sum_f P_t(z|f)L^s_{ft}}{\sum_z \sum_f P_t(z|f)L^s_{ft}}. \end{aligned}$$

Only the $P_s(f|z)$ values are used in reconstruction; the rest of the terms are discarded. Figure 3·10 shows a few examples of typical $P_s(f|z)$ distributions learned for a male and a female talker. Figure 3·11 shows more examples of $P_s(f|z)$ distributions characterizing different sources.

3.5.2 Latent Variable Model for Mixture Spectrogram

Before we can describe how to separate the sources, we should have a model for approximating the mixture spectrogram.

In a mixture spectrogram, a fraction of the total spectral content in each frequency is derived from each source. The spectrum is modeled as the outcome of repeated draws from

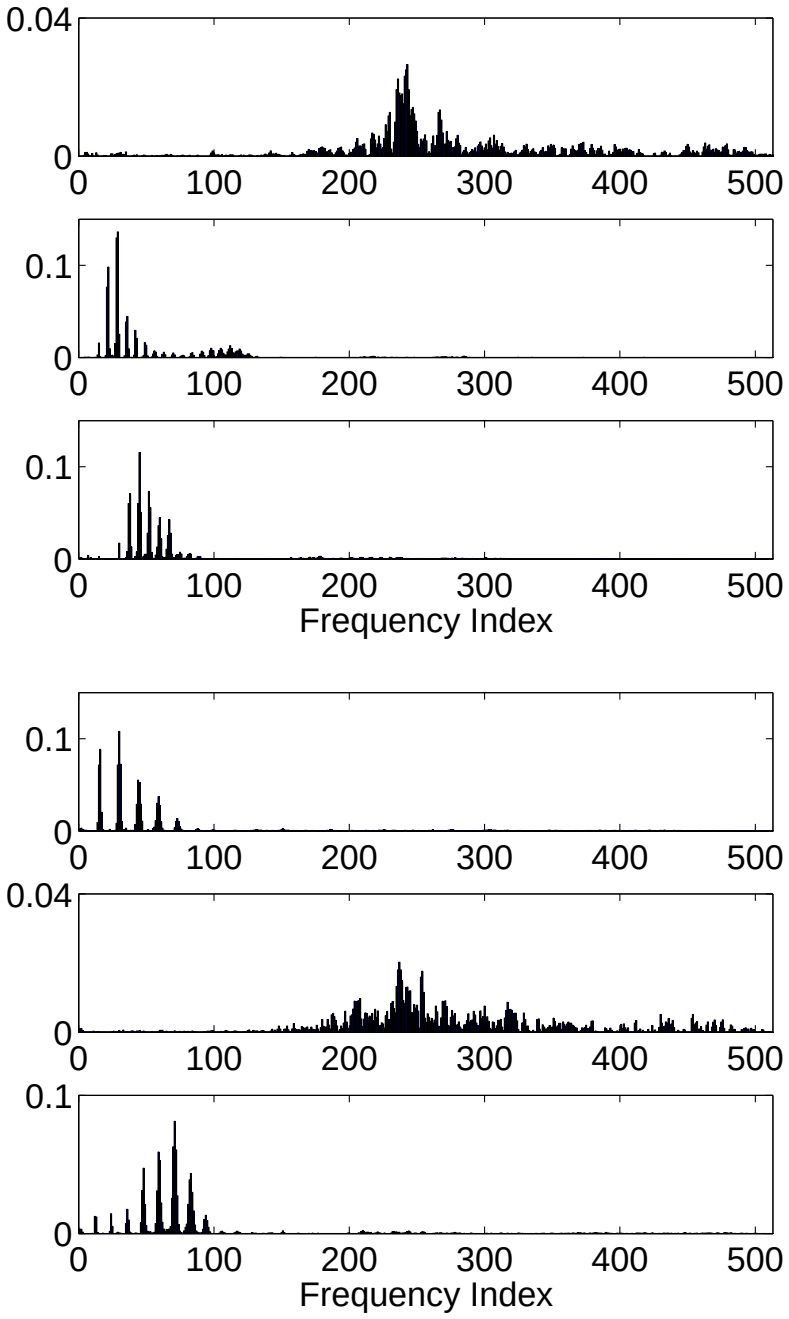
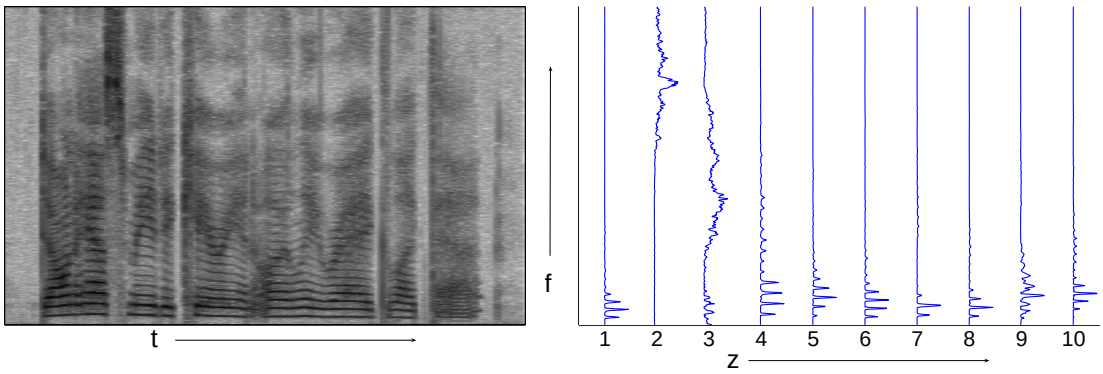
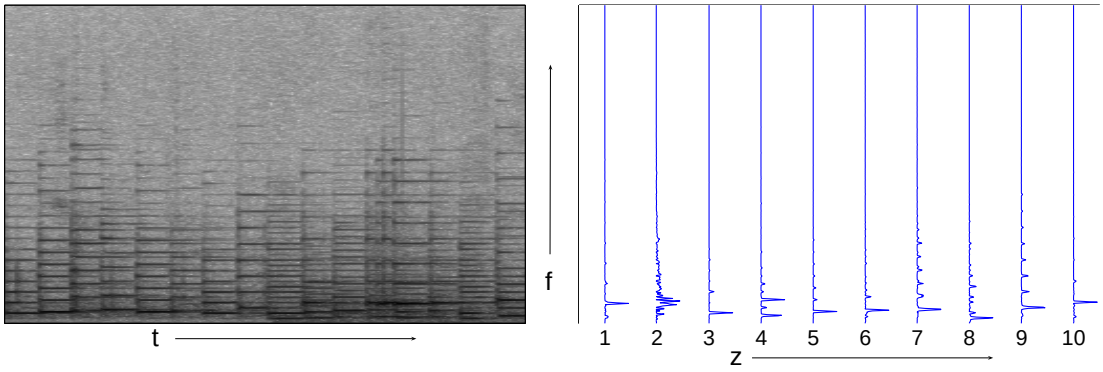


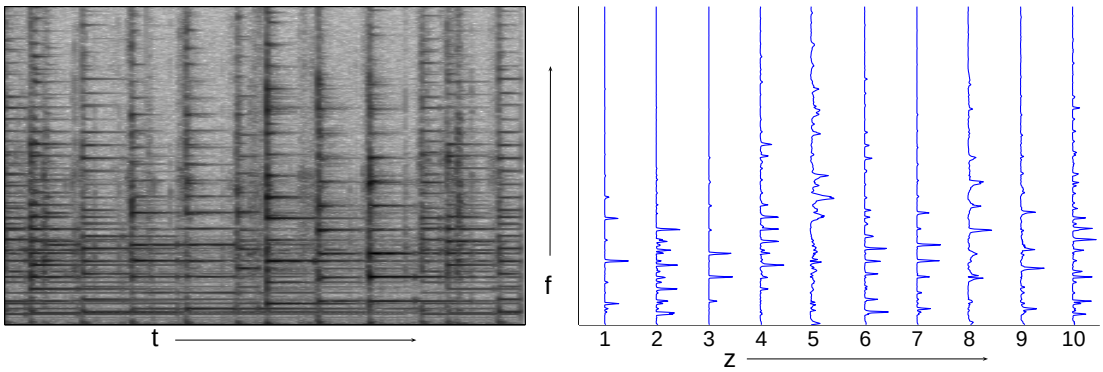
Figure 3·10: The three histograms on the top panel shows typical multinomial distributions obtained for a male talker. The three panels on the bottom show typical multinomials for a female talker.



(a) Speech Spectrogram (left) and Basis Components (right)



(b) Piano Spectrogram (left) and Basis Components (right)



(c) Harp Spectrogram (left) and Basis Components (right)

Figure 3·11: Examples of typical basis components learned from (a) speech, (b) piano, and (c) harp. Notice that the basis components for different signals are qualitatively different and have spectral structure characteristic of the sources they represent.

a two-level random process. Within each draw, the process first draws a source (represented by the latent variable s), then a specific multinomial for the source (latent variable z), and finally a frequency index f from the multinomial. The constraint here is that z takes on a different set of values for each source. The overall distribution underlying the spectral vector for the t-th analysis frame is given by

$$P_t(f) = \sum_s P_t(s) \sum_{z \in \{\mathbf{z}_s\}} P_t(z|s) P_s(f|z), \tag{3.38}$$

where $P_t(s)$ is the *a priori* probability of the s-th source and $\{\mathbf{z}_s\}$ represents the set of values that z can take for that source.

3.5.3 Separating Sources from Mixtures

The process of extracting the contributions of each source to the mixture spectrogram has two stages. In the first stage, the mixture multinomial distribution of each of the sources is estimated in each analysis frame. This implies the estimation of all parameters of equation (3.38) except the $P_s(f|z)$ terms which are obtained from the training stage. In the second stage, the separated spectrum for the source within every frame is obtained as the expected value of the number of draws of each frequency index from the mixture multinomial distribution for the source.

The $P_t(s)$ and $P_t(z|s)$ terms of equation (3.38) can be estimated by iterations of the following equations derived using the EM algorithm:

$$\begin{aligned} P_t(s,z|f) &= \frac{P_t(s)P_t(z|s)P_s(f|z)}{\sum_s P_t(s) \sum_{z \in \{\mathbf{z}_s\}} P_t(z|s)P_s(f|z)} \\ P_t(s) &= \frac{\sum_{z \in \{\mathbf{z}_s\}} \sum_f P_t(s,z|f) V_{ft}}{\sum_s \sum_{z \in \{\mathbf{z}_s\}} \sum_f P_t(s,z|f) V_{ft}} \\ P_t(z|s) &= \frac{\sum_f P_t(s,z|f) V_{ft}}{\sum_{z \in \{\mathbf{z}_s\}} \sum_f P_t(s,z|f) V_{ft}}. \end{aligned} \tag{3.39}$$

Details of the derivation are shown in Appendix A.

Once all the terms have been estimated, the mixture multinomial distribution for the

s-th source in the t-th analysis frame can be obtained as

$$P_t(f|s) = \sum_{z \in \{\mathbf{z}_s\}} P_t(z|s) P_s(f|z). \tag{3.40}$$

According to the model, the total number of draws of any frequency is the sum of the draws for the individual sources, i.e.

$$V_{ft} = \sum_s V_{ft}(s), \tag{3.41}$$

where $V_{ft}(s)$ is the number of draws of f from the s-th source. The expected value of $V_{ft}(s)$, given the total count V_{ft}, is hence given by

$$\hat{V}_{ft}(s) = \frac{P_t(s) P_t(f|s)}{\sum_s P_t(s) P_t(f|s)} V_{ft}. \tag{3.42}$$

$\hat{V}_{ft}(s)$ is the estimated value of frequency f in the spectral vector of the s-th source and the t-th frame. The phase of the short-time Fourier transform of the mixed signal is paired with the estimated magnitude given by $\hat{V}_{ft}$. An inverse Fourier transform is then performed to obtain the time domain reconstruction for the source.

3.5.4 Separation Results

We now present results of experiments that demonstrate the applicability of the latent variable framework for separation of talkers. Experiments were conducted on synthetic mixtures of talkers taken from the Wall Street Journal (WSJ) database. We evaluated the results on six pairs of talker combinations – two were female/male pairs, two were male/male, and two were female/female. Three female and three male talkers were randomly chosen from the database to obtain the six talker pairs. For every talker, the WSJ corpus consists of about 140 utterances comprising between 16 to 18 minutes of speech. Of these, 134 utterances were randomly chosen to serve as the training set. The remaining 6 utterances were labeled as the test set. The sampling rate for all the signals was set to 16 kHz.

We used short-term Fourier transforms to obtain spectrograms from the time signals. We incremented our analysis frame by one-fourth of the FFT length. No zero padding was

used, and data was shaped by a Hanning window before the FFT. We used various values for the FFT size, taken from the set $\{128, 256, 512, 1024, 2048, 4096\}$, corresponding to an analysis window length ranging from 8 ms to 256 ms. All computations were performed using MATLAB software.

Consider a given experiment where the task is to separate the two talkers present in the mixture signal. In the training stage, we learned K basis components, $K \in \{10, 20, 40, 80, 120, 160, 200\}$, from the training data for each talker. Following the procedure outlined in Section 3.5.1, 15 randomly chosen utterances from the training set, comprising about 100 to 120 seconds of speech, were used to determine the basis components.

We created the mixture signal by digitally adding test signals for both talkers. The length of the mixed signal was set to the shorter of the two signals. Prior to addition, the signals were normalized to have 0 mean and unit variance, resulting in a 0 dB target-to-interference ratio for each talker. The mixture spectrogram was analyzed using the procedure outlined in Section 3.5.3 to obtain reconstructions of both talkers.

The quality of speech separation is hard to evaluate reliably. We provide two measures that have been used in the literature. Let ${}^{i}\mathbf{O}$ and ${}^{i}\mathbf{R}$ represent the magnitude spectrograms of the original test signal and the reconstructed signal of the i-th talker in the mixture. Let $\mathbf{N}$ and $\mathbf{\Phi}$ represent the magnitude and phase of the mixture spectrogram. Define a function

$$g_i(\mathbf{X}) = 10 \log_{10} \left(\frac{\sum_{f,t} {}^{i}O_{ft}^2}{\sum_{f,t} |{}^{i}O_{ft} e^{j\Phi_{ft}} - X_{ft} e^{j\Phi_{ft}}|^2} \right). \tag{3.43}$$

Following Raj and Smaragdis (2005), we define the *SNR improvement* for the i-th talker as

$$SNR_i = g_i({}^{i}\mathbf{R}) - g_i(\mathbf{N}) \tag{3.44}$$

The second metric, *Speaker Energy Ratio (SER)*, was used by Smaragdis (2007) and is based on correlations between reconstructed and original signals. The SER for talker i is given by

$$SER_i = 10 \log_{10} \left(\frac{c_{ii}}{\sum_{\forall j \neq i} c_{ij}} \right) \tag{3.45}$$

where c_{ij} is the correlation between the reconstructed time signal for the i-th talker and the original signal for the j-th talker.

Figures 3·12 and 3·13 summarize results, plotting SNR and SER improvements, respectively, for various cases of FFT sizes and number of basis components. The SNR and SER values were averaged over six experiments where each experiment had a different mixture of test signals. The separation results for Male/Female talker combinations are much better than the same sex talker combinations. We obtain average SNR improvements of up to 6 dB in the Male/Female case and up to 3 dB in the case of same sex talker pairs, primarily because the basis components of talkers of the same sex have more similar characteristics than basis components of different-sex talkers. There needs to be some difference in the spectral quality of the sources present in the mixture for obtaining good performance with the algorithm. The more similar the spectral characteristics are, the poorer performance will be. The degree of separation achieved depends on the specific talker pair present in the mixture; not all talker pairs of the same sex will result in poor separation. Performance varies depending on the choice of FFT size and the number of basis components. In most cases, FFT sizes between 512 and 2048, in conjunction with 40 to 80 basis components, provides good performance. Figures 3·14 and 3·15 show how the FFT size and the choice of number of basis components affect performance on average. Small values for FFT size (128 points and less) will result in the omission of low frequencies in the representation, while FFT sizes longer than phoneme widths fail to model formant variations present in speech, thus resulting in poor performance. In theory, a larger number of basis components better approximates a talker. However, if too many basis components are used, they begin to model the other talker in the mixture, reducing performance. A choice of 40 or 80 basis components and an FFT size of 1024 points provides reasonable performance across all talker combinations. Figure 3·16 shows spectrograms from a particular example for the Male/Female talker pair where the spectrograms used an FFT point size of 1024 and the training stage extracted 80 basis components.

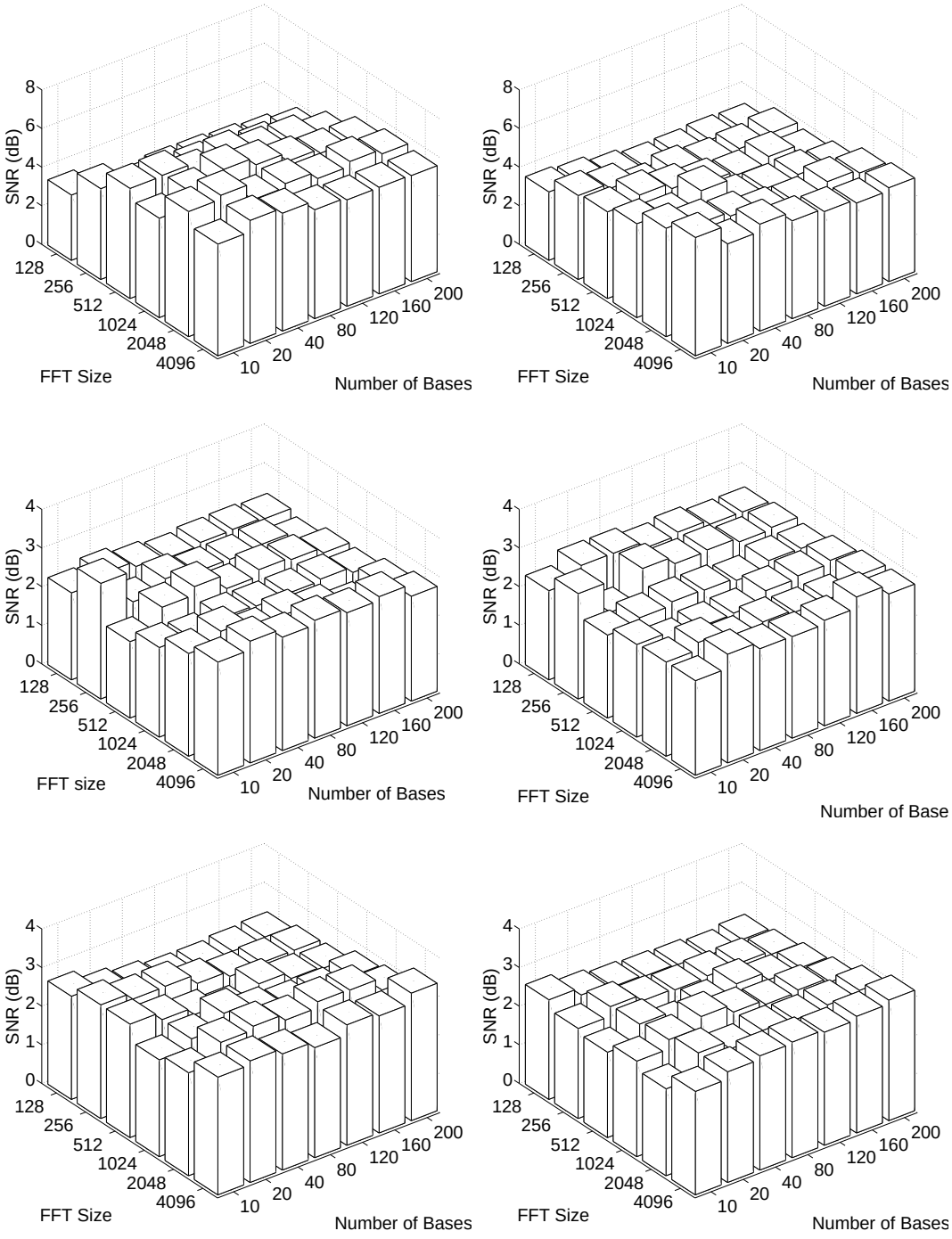


Figure 3·12: Average SNR improvements for various combinations of number of basis components and FFT sizes. The two top-panels correspond to the two Male/Female talker pairs, middle-panels correspond to the Female/Female pairs and bottom-panels correspond to Male/Male pairs.

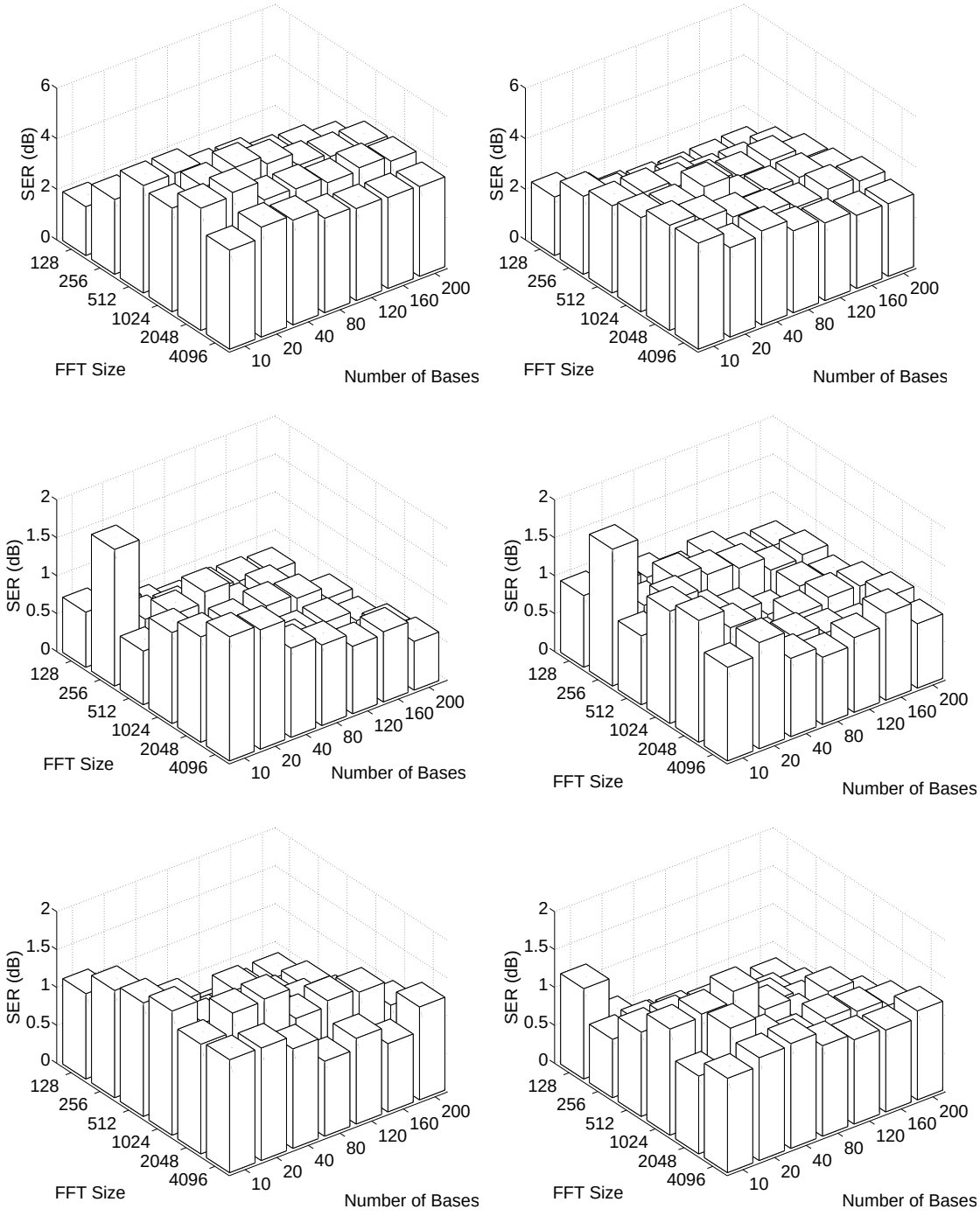


Figure 3·13: Average SER improvements for various combinations of number of basis components and FFT sizes. The two top-panels correspond to the two Male/Female talker pairs, middle-panels correspond to the Female/Female pairs and bottom-panels correspond to Male/Male pairs.

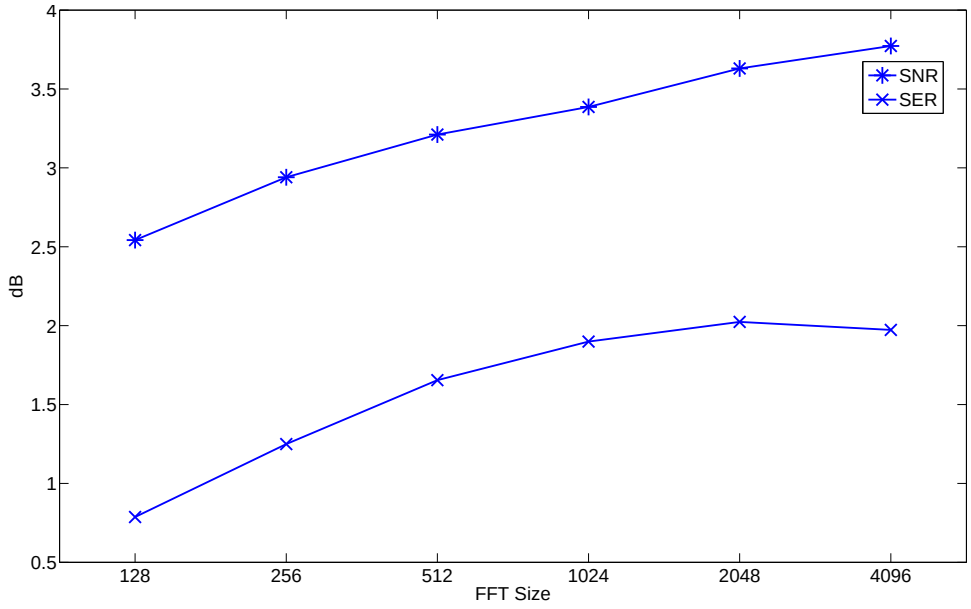


Figure 3·14: Average SNR and SER improvements for different FFT sizes. Results are averaged over all talker pairs and number of basis components.

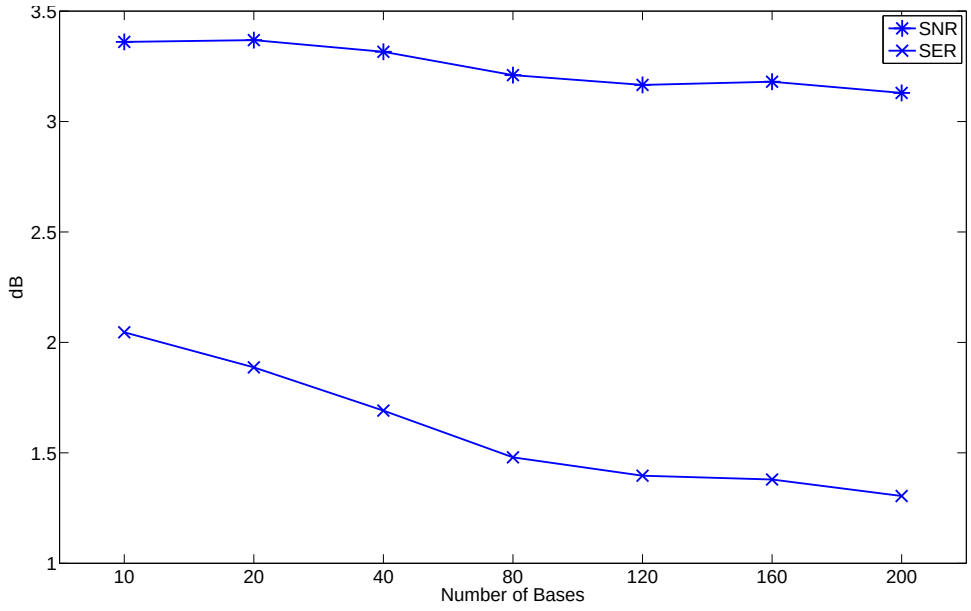


Figure 3·15: Average SNR and SER improvements for different choices of the number of basis components learned. Results are averaged over all talker pairs and FFT lengths.

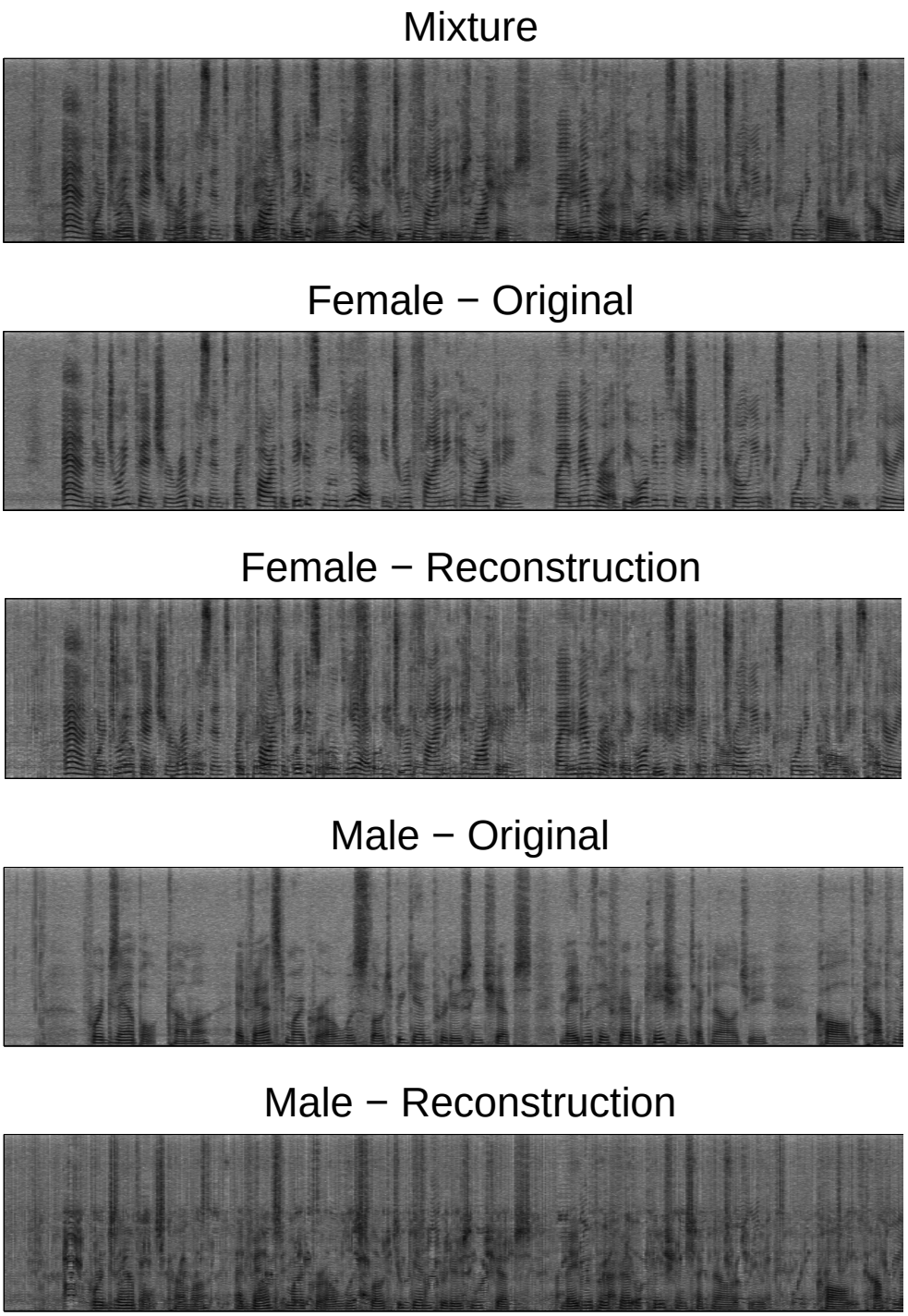


Figure 3·16: Result of a Separation Experiment for a Male/Female Talker Pair, with 1024 point FFT size and 80 basis components. The SNR and SER improvements for the female were 6.6194 dB and 4.4414 dB respectively. For the male, the improvements were 6.4959 dB and 5.1683 dB.

3.5.5 Other Applications

The latent variable framework can be used for other applications in addition to source separation. Below, we briefly present an example of semi-supervised denoising application.

Semi-supervised Denoising

Now, consider a situation where the available speech signal is noisy. The talker characteristics are known a priori (i.e., clean training data from the talker is available), but the noise present in the mixture is unknown. We can use the latent variable framework to remove noise from the mixture.

As in the case of talker separation, we first learn a set of basis components for the talker from the training data. The separation stage follows the processing steps discussed in the talker separation application. In addition, we also update (learn) the basis components representing the noise. In other words, for the known talker, we estimate the mixture weights while keeping the basis vectors fixed; however we estimate both the basis vectors and mixture weights for the noise component. Figure 3·17 shows an example in which interfering chime noise from cymbals was removed from a noisy signal of female speech.

This approach of semi-supervised separation can also be used to extract foreground singers or lead instruments from the background music in a song. Examples can be found at http://cns.bu.edu/∼mvss/courses/speechseg/.

Another example application of the latent variable framework is bandwidth expansion. The idea is to estimate high-frequency components of narrow-band signals, such as signals carried over over a telephone channel. In the training stage, basis components are learned from full-band signals. The estimation stage has two steps. We first estimate mixture weights for the test signals by approximating them as linear combinations of the narrow-band portion of the learned basis vectors. The full-band basis components are then combined with the newly estimated mixture weights, from which counts for the unobserved frequencies are estimated. Details and example results are reported in (Raj et al., 2007).

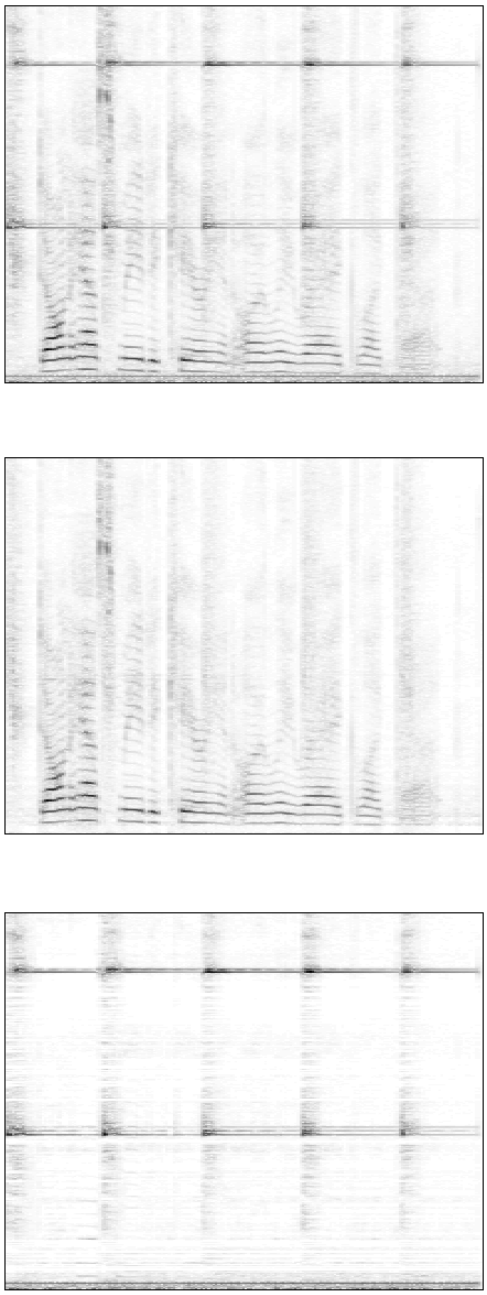


Figure 3·17: Results of a denoising experiment using the latent variable framework. The top plot shows the spectrogram of a speech utterance by a female talker mixed with chimes from cymbals. Twenty bases were learned from training data for the talker. During separation, five extra bases were learned to account for the unknown source (cymbals). The separated speech spectrogram (mid-panel) and noise spectrogram (bottom-panel) are shown.

3.6 Discussion and Conclusions

This chapter introduced a probabilistic latent variable framework for single-channel acoustic processing. The central idea of the framework is to model the random process that generates a given spectral vector as a mixture of hidden multinomial distributions. These latent distributions or basis components are assumed to be characteristic of the source that generates the entire set of spectral vectors comprising the signal. We presented the theory and illustrated the workings of the model with a geometric interpretation. We derived inference algorithms and showed how the framework can be used for source separation and other applications, including denoising. We demonstrated the utility of the framework by presenting results of separation experiments.

A framework using latent components or basis components is very powerful. Standard models such as Gaussian Mixture Models or Hidden Markov Models that are typically employed to model spectrograms work well for monophonic sounds. However, these models grow in complexity for polyphonic sound examples, and are not designed to model the property of additivity, describing how energy from multiple sounds combines in each frequency-time bin of the mixed signal. The latent variable framework provides an explicit way to represent such mixture sounds as being composed of a linear combination of underlying components. This allows the model to be simple and at the same time endows it with the flexibility to model various types of mixtures. The second important advantage of the latent variable framework is its probabilistic formulation, which allows us to employ statistical methods for estimating model parameters. This approach also enables one to model known or hypothesized structure in the data in the form of prior distributions. One such prior, imposing sparsity, will be the focus of the next chapter.

An important limitation of the proposed framework is related to the number of components that can be extracted. The number of components that are required to characterize a particular source potentially could be very large. However, the proposed framework cannot learn more components than the dimensionality of the spectral vectors, i.e. the number of

frequency bins. As discussed in Section 3.4, two problems arise if we attempt to extract an overcomplete set of basis components where there are more components than the dimensionality of the spectral vector. First of all, there will be multiple feasible solutions, because the problem will be under-determined. Secondly, the feasible solutions do not necessarily characterize the data very well, a problem considered further in the next chapter. Thus, the number of components that can be extracted is limited by the number of frequency bins, which in turn depends on the representation chosen to describe the input. The dimensionality of the spectral vectors is not a true characteristic of the signal being analyzed, but is instead just a characteristic of the representation selected for the signal. It is not reasonable to expect the number of true underlying components of a signal to be limited by the representation, and this problem reveals a logical flaw in the approach where an arbitrary choice in the initial representation directly impacts the quality of the solution that will be found. The next chapter extends and improves the model to overcome this limitation, using the concept of sparsity.

Chapter 4

Sparse Overcomplete Latent Variable Decomposition

Representation of the world, like the world itself, is the work of men; they describe it from their own point of view, which they confuse with the absolute truth.

Simone de Beauvoir

4.1 Introduction

This chapter presents an extension to the latent variable framework that allows it to overcome the limitations discussed in the previous chapter. One of the main weaknesses of latent variable decomposition is related to the number of basis components one can extract. When modeling spectrograms, this limitation means that the number of basis components that can be found is limited to be equal to or less than the dimensionality of the spectral vectors. Thus, the model is limited by the representation that is chosen to describe the signal being analyzed. In this chapter, we present a learning formulation that enables one to extract an *overcomplete* set of meaningful components, with more components than the dimensionality of the spectral vectors. We employ the notion of sparsity for this purpose. Sparse coding refers to a representational scheme where, of a set of components that may be combined to approximate or represent data, only a small number are necessary to represent any particular input. In the framework of latent variable decomposition, one way to obtain such a sparse overcomplete code of basis components is to constrain the mixture weights associated with the basis components to have low entropy. A mixture weight set with low entropy guarantees that only a few mixture weight terms are significant. We show that this

approach eliminates the problem of indeterminacy, permitting us to learn an unrestricted number of basis components. Mathematically, the general approach provides a way to explicitly control the entropy of any of the parameters of the model. Since entropy is an information theoretic measure of "information," the approach provides a way to control the information content or the "expressiveness" of the basis components.

The chapter is organized as follows. Section 4.2 presents the theory and mathematical concepts behind this extension. We motivate the need for sparsity, present entropy as a sparsity metric, and show how it can be incorporated into the latent variable framework. We use a maximum a posteriori formulation and derive inference algorithms. In Section 4.3, we provide a geometrical interpretation of the model to give some intuition into sparse overcomplete codes. Section 4.4 shows how the new formulation can be effective for source separation and presents results of experiments. We briefly review other approaches that have been used for sparsity in Section 4.5 and end the chapter with conclusions in Section 4.6.

4.2 Sparsity in the Latent Variable Framework

This section introduces the concept of sparsity. We first motivate the need for sparsity in the latent variable framework. We then show how it can be imposed in the framework and derive inference algorithms.

4.2.1 The Need for Sparsity

Consider a real signal, such as a speech utterance. It is reasonable to expect that such real signals exhibit complex spectral structure. The number of components required to model the structure could potentially be very large. However, the latent variable framework as introduced in the last chapter has an upper bound on the number of basis components that one can extract. This limit is given by the dimensionality of the input vectors, which in the case of spectrograms, is provided by the number of frequency bins. This is a clear conceptual limitation, since the model is restricted by the representation used (which is

an arbitrary choice, not grounded in any theoretical considerations) for the signal being analyzed.

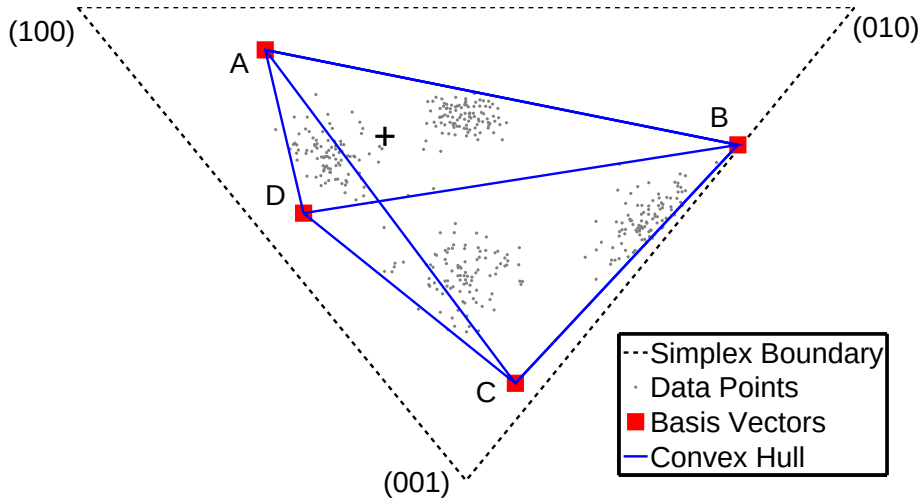


Figure 4·1: Illustration of multiple solutions in an overcomplete code of four basis components. The basis components are marked A, B, C and D. A typical data point '+' can be expressed as a linear combination of the basis components in several ways as evidenced by the multiple different convex hulls in which it lies. The convex hulls are given by ABCD, ABC and ABD.

The main problem in extracting an overcomplete set of basis components is indeterminacy. There are multiple ways in which one can combine an overcomplete set of basis components to approximate any particular data distribution. This is best illustrated by utilizing the artificial dataset first presented in Section 3.4. An overcomplete set of four basis components are shown in Figure 4·1, corresponding to the points A, B, C, and D. The figure illustrates the various ways in which the basis vectors can be combined to represent a typical data point. These basis points are capable of representing the data distributions as the data falls within the convex hull defined by these points. However, there are many different ways in which any given data point can be represented using these four basis components.

To overcome this problem of indeterminacy, additional constraints have to be imposed during parameter estimation, to lead to a unique solution. The concept of sparsity is one such constraint that has been widely used. The goal is to find a set of components such that the mixture weights by which the basis components are multiplied prior to being added to produce a datum are "sparse;" i.e., few of the weights are large. Adding a penalty to solutions that require few non-zero weights will favor a sparse solution over another solution that also can represent a particular datum, but requires more non-zero weights. Figure 4·2 illustrates a a sparse overcomplete code and compares the sparse code with a "dense" or compact code.

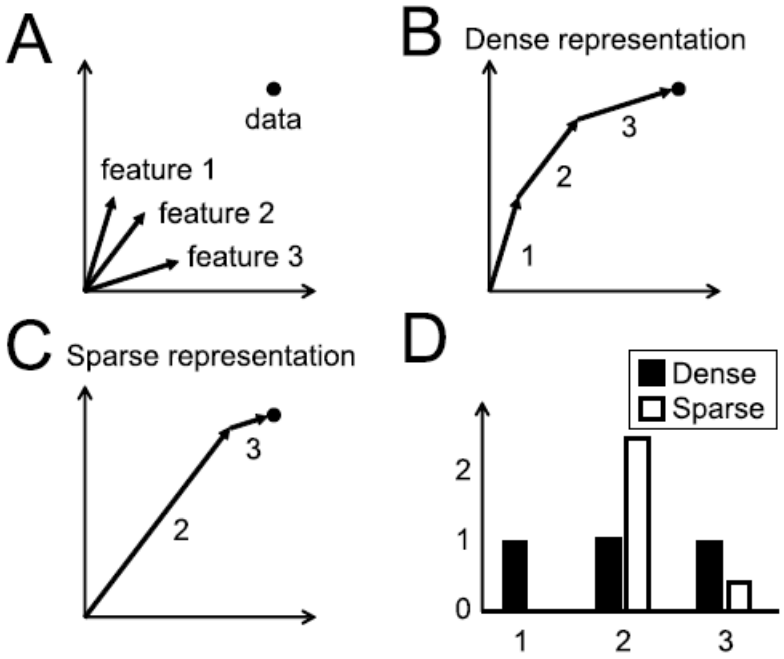


Figure 4·2: Reproduced from Asari et al. (2006), Figure 1, with the permission of authors. (A) Three non-orthogonal feature vectors in a 2D space constitute an overcomplete representation, offering many possible ways to represent a data point with no error. (B) A dense representation that weights all features roughly evenly. (C) A sparse representation that invokes only two features. (D) The sparse and dense representations compared.

4.2.2 Entropy as a Sparsity Metric

Different metrics have been proposed to measure sparsity. These metrics are used as constraints during model parameter estimation to favor sparse coding. These constraints correspond to different cost functions that, during estimation, penalize the objective function corresponding to the solution requiring more non-zero weights, thereby favoring equally "good" solutions for reconstructing a datum that requires fewer non-zero weights. Consider a distribution $\boldsymbol{\theta}$ for which a sparse code is desired. Some approaches use variants of the L_p norm of $\boldsymbol{\theta}$ as the cost function to favor sparse coding (Hoyer, 2004) while other approaches use various approximations of entropy of $\boldsymbol{\theta}$ as the cost function (Field, 1994). Instead of using approximations for entropy, we directly calculate the entropy itself as a sparsity metric and seek to reduce this metric at the same time that we find a solution that models the data. Figure 4·3 illustrates how adding a constraint that favors reducing the entropy of the mixture weights leads to a unique solution.

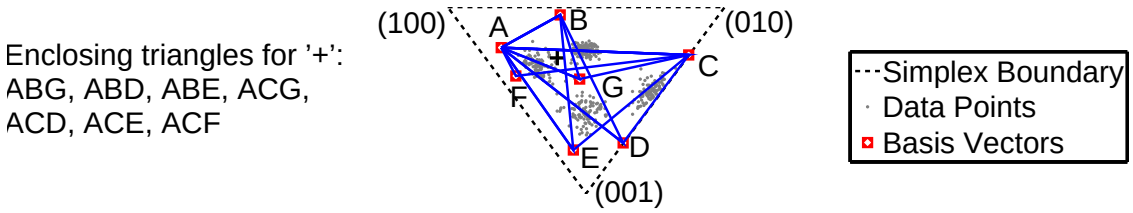


Figure 4·3: Illustration of sparsifying mixture weights in an overcomplete code. A-G represent 7 basis components. The '+' represents a typical data point. This datum can be accurately represented by any set of three or more bases that form an enclosing polygon; moreover, there are many such polygons consistent with the underconstrained nature of the problem. However, if the goodness of a solution weights the number of bases used to enclose '+' to be minimal, favoring solutions that use fewer non-zero weights, only the 7 enclosing triangles listed may be optimal solutions. By further imposing the restriction that the entropy of the mixture weights is to be minimized, only one triangle is obtained as the unique, optimal enclosure.

There is another advantage of using entropy as a sparsity metric. In information theory, entropy is a measure of the information encoded by a distribution. Reducing the entropy of the mixture weights results in increased entropy of the basis vectors (increasing the in-

formation they convey). Sparse-coding, where entropy of the mixture weights is reduced, forces more information to be encoded by the basis components, making them more "expressive." Thus, using entropy as a metric provides an explicit way to control the amount of information present in the basis components versus in the mixture weights.

4.2.3 Parameter Estimation

The concept of *entropic prior* has been used in the maximum entropy literature (Jaynes, 1982; Skilling, 1989) to enforce sparsity. Given a probability distribution $\boldsymbol{\theta}$, the entropic prior is defined as

$$P_e(\boldsymbol{\theta}) \propto e^{-\alpha \mathcal{H}(\boldsymbol{\theta})}, \tag{4.1}$$

where $\mathcal{H}(\boldsymbol{\theta}) = -\sum_i \theta_i \log \theta_i$ is the entropy of the distribution and α is a weighting factor. Positive values of α favor distributions with lower entropies while negative values of α favor distributions with higher entropies. Imposing this prior with positive α during *maximum a posteriori* estimation is a way to minimize entropy, which will result in a sparse $\boldsymbol{\theta}$ distribution. The distribution $\boldsymbol{\theta}$ could correspond to the basis components $P(f|z)$, the mixture weights $P_t(z)$, or both.

We use the EM algorithm to derive update equations for the parameters of the model. Let us examine the case in which both $P(f|z)$ and $P_t(z)$ employ the entropic prior[9]. The model is given by the equation

$$P_t(f) = \sum_z P(f|z) P_t(z).$$

The set of parameters to be estimated are $P(f|z)$ and $P_t(z)$ i.e. $\Lambda = \{P(f|z), P_t(z)\}$. We impose an *a priori* probability on the parameters given by

$$P(\Lambda) \propto \prod_z e^{\bar{\alpha} \sum_f P(f|z) \log P(f|z)} \prod_t e^{\bar{\beta} \sum_z P_t(z) \log P_t(z)},$$

[9] In this thesis, we only consider the case in which we impose a sparsity constraint on the mixture weights $P_t(z)$. However, we present the case where both the basis components and mixture weights have the entropic prior to keep the exposition general.

where $\bar{\alpha}$ and $\bar{\beta}$ are parameters indicating the relative importance and sign of the sparsity desired on $P(f|z)$ and $P_t(z)$, respectively. Ignoring constant terms, the log-prior (logarithm of the above *a priori* probability) can be written as

$$\log P(\Lambda) = \bar{\alpha} \sum_z \sum_f P(f|z) \log P(f|z) + \bar{\beta} \sum_t \sum_z P_t(z) \log P_t(z), \tag{4.2}$$

We use *maximum a posteriori* estimation and use the EM algorithm.

For the E-step, we compute the *a posteriori* probability of the latent variable as before:

$$P_t(z|f) = \frac{P_t(z)P(f|z)}{\sum_z P_t(z)P(f|z)}. \tag{4.3}$$

In the M-step, instead of maximizing the log-likelihood, we maximize the log-posterior (the logarithm of the *a posteriori* probability of the model parameters). The log-posterior to be maximized is given by

$$\begin{aligned} \mathcal{L} &= \mathcal{D} + \mathcal{R} \\ &= E_{\bar{z}|\bar{f};\Lambda} \log P(\bar{f}, \bar{z}) + \log P(\Lambda), \end{aligned} \tag{4.4}$$

where $\mathcal{D} = E_{\bar{z}|\bar{f};\Lambda} \log P(\bar{f}, \bar{z})$ is the expected log-likelihood, $\mathcal{R} = \log P(\Lambda)$ is the log-prior, and $\bar{f}$ and $\bar{z}$ represent the set of all observations of f and z in the draws that generated all data vectors.

Let us consider the log-likelihood term $\mathcal{D}$ first. The complete data likelihood can be written as

$$P(\bar{f}, \bar{z}) \propto \prod_{j,t} P_t(f_j, z_j) = \prod_{j,t} P_t(z_j) P(f_j|z_j), \tag{4.5}$$

where f_j and z_j are the observed values of variables f and z in the j-th draw. Hence, we

can write the function $\mathcal{D}$ as (ignoring the constant terms)

$$\begin{aligned}
\mathcal{D} &= E_{\bar{z}|\bar{f};\Lambda} \sum_{j,t} \log P_t(f_j, z_j) \\
&= \sum_{j,t} E_{z_j|f_j;\Lambda} \log P_t(f_j, z_j) \\
&= \sum_{j,t} E_{z_j|f_j;\Lambda} \log P_t(z_j) + \sum_{j,t} E_{z_j|f_j;\Lambda} \log P(f_j|z_j) \\
&= \sum_{j,t} \sum_{z} P(z|f_j) \log P_t(z) + \sum_{j,t} \sum_{z} P(z|f_j) \log P(f_j|z). \qquad (4.6)
\end{aligned}$$

In the above equation, we can change the summation over draws j to a summation over features f by accounting for how many times f was observed, i.e. the f-th entry in the observed data vector[10]. The expected log-likelihood can now be written as

$$\mathcal{D} = \sum_{t} \sum_{f} \gamma V_{ft} \sum_{z} P_t(z|f) \log P_t(z) + \sum_{t} \sum_{f} \gamma V_{ft} \sum_{z} P_t(z|f) \log P(f|z). \qquad (4.7)$$

The second term $\mathcal{R}$ in equation (4.4) corresponding to the log-prior is given by equation (4.2). Hence, we can write the function $\mathcal{L}$ as (ignoring the constant terms)

$$\begin{aligned}
\mathcal{L} &= \mathcal{D} + \mathcal{R} \\
&= \sum_{t} \sum_{f} \gamma V_{ft} \sum_{z} P_t(z|f) \log P_t(z) + \sum_{t} \sum_{f} \gamma V_{ft} \sum_{z} P_t(z|f) \log P(f|z) \\
&\quad + \bar{\alpha} \sum_{z} \sum_{f} P(f|z) \log P(f|z) + \bar{\beta} \sum_{t} \sum_{z} P_t(z) \log P_t(z). \qquad (4.8)
\end{aligned}$$

Here, γ is a parameter that weights the data while $\bar{\alpha}$ and $\bar{\beta}$ are parameters weighting the priors.

In order to take care of the normalization constraints, the above equation must be augmented by appropriate Lagrange multipliers τ_t and ρ_z,

$$Q = \mathcal{L} + \sum_{t} \tau_t \Big(1 - \sum_{z} P_t(z)\Big) + \sum_{z} \rho_z \Big(1 - \sum_{f} P(f|z)\Big). \qquad (4.9)$$

[10]Since observed data is modeled as a histogram, entries should be integers. To account for this, we weight the data by an unknown scaling factor γ, without loss of generality.

Maximization of Q with respect to $P_t(z)$ and $P(f|z)$ leads to the following sets of equations

$$\frac{\sum_t V_{ft} P_t(z|f)}{P(f|z)} + \alpha + \alpha \log P(f|z) + \rho_z = 0, \tag{4.10}$$

$$\frac{\sum_f V_{ft} P_t(z|f)}{P_t(z)} + \beta + \beta \log P_t(z) + \tau_t = 0, \tag{4.11}$$

where $\alpha = \bar{\alpha}/\gamma$ and $\beta = \bar{\beta}/\gamma$. We have replaced two parameters weighting the data and prior separately (γ and $\bar{\alpha}$ for equation (4.10), γ and $\bar{\beta}$ for equation (4.11)) by a single parameter that weights the prior with respect to the data (α and β in equations (4.10) and (4.11) respectively).

Now, consider solving for $P_t(z)$. Equation (4.11) can be written as

$$\frac{\omega_z}{P_t(z)} + \beta + \beta \log P_t(z) + \tau_t = 0, \tag{4.12}$$

where ω_z represents $\sum_f V_{ft} P_t(z|f)$. The above set of simultaneous transcendental equations for $P_t(z)$ can be solved using the Lambert's $\mathcal{W}$ function (Corless et al., 1996) as proposed by Brand (1999a).

Lambert's $\mathcal{W}$ function is an inverse mapping satisfying

$$\mathcal{W}(y) e^{\mathcal{W}(y)} = y \qquad \Longrightarrow \qquad \log \mathcal{W}(y) + \mathcal{W}(y) = \log y.$$

As shown by Brand (1999a), we can set $y = e^x$ and work backwards towards equation (4.12) as follows,

$$\begin{aligned} 0 &= -\mathcal{W}(e^x) - \log \mathcal{W}(e^x) + x \\ &= \frac{-1}{1/\mathcal{W}(e^x)} - \log \mathcal{W}(e^x) + x + \log q - \log q \\ &= \frac{-q}{q/\mathcal{W}(e^x)} + \log q/\mathcal{W}(e^x) + x - \log q. \end{aligned}$$

Setting $x = 1 + \tau_t/\beta + \log q$ and $q = -\omega_z/P_t(z)$, the above equation simplifies to equa-

tion (4.12):

$$\begin{aligned} 0 &= \frac{\omega_z/\beta}{-(\omega_z/\beta)/\mathcal{W}(-\omega_z e^{1+\tau_t/\beta}/\beta)} + \log \frac{-\omega_z/\beta}{\mathcal{W}(-\omega_z e^{1+\tau_t/\beta}/\beta)} \\ &\quad +1 + \frac{\tau_t}{\beta} \\ &= \frac{\omega_z/\beta}{P_t(z)} + \log P_t(z) + 1 + \frac{\tau_t}{\beta}, \end{aligned}$$

which implies that

$$\hat{P}_t(z) = \frac{-\omega_z/\beta}{\mathcal{W}(-\omega_z e^{1+\tau_t/\beta}/\beta)}, \tag{4.13}$$

where equations (4.12) and (4.13) form a set of fixed-point iterations for τ_t, and thus the M-step for finding $P_t(z)$. Brand (1999a) points out that these equations typically converge in 2-5 iterations. Brand (1999b) provides details of computation of the Lambert's $\mathcal{W}$ function.

$P(f|z)$ can be found by solving the set of transcendental equations given by equation (4.10) using Lambert's $\mathcal{W}$ function. It can be estimated as

$$\hat{P}(f|z) = \frac{-\xi/\alpha}{\mathcal{W}(-\xi e^{1+\rho_z/\alpha}/\alpha)}, \tag{4.14}$$

where ξ is $\sum_t V_{ft} P_t(z|f)$. Equations (4.10) and (4.14) form a set of fixed-point iterations and correspond to the M-step updates for $P(f|z)$.

4.2.4 Examples

Consider a simple music clip shown by the magnitude spectrogram in Figure 4·4. This example can be used to show how sparsity can be useful in analyzing sounds. The music clip consists of three notes played successively followed by a chord which is composed of all the three notes. Learning three basis components by performing latent variable decomposition on the spectrogram provides results as shown in Figure 4·5(a). The three components correspond to the three notes present in the clip. The mixture weight corresponding to any particular basis component is high in all those time frames where that note is "on." In the last segment of the clip corresponding to the chord, mixture weights of all the three components have roughly equal values indicating that all the three notes are present.

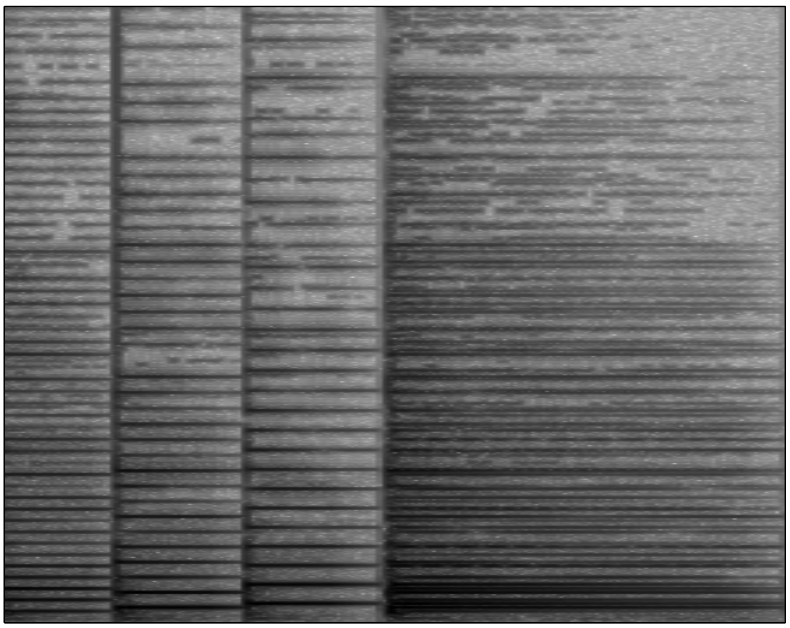
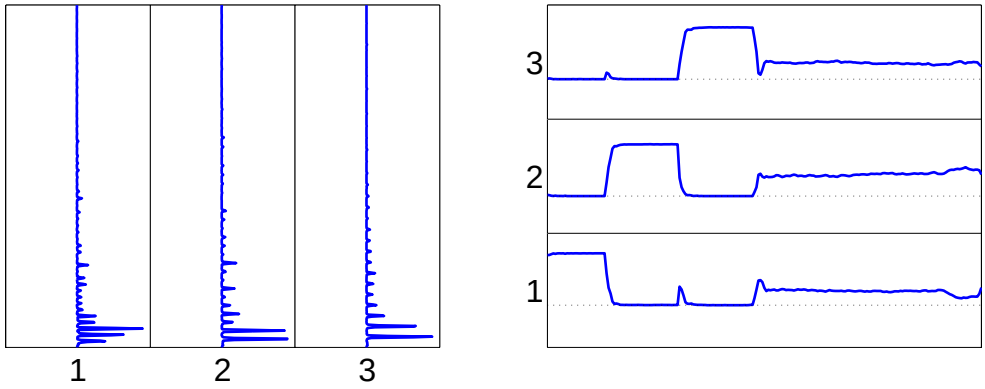
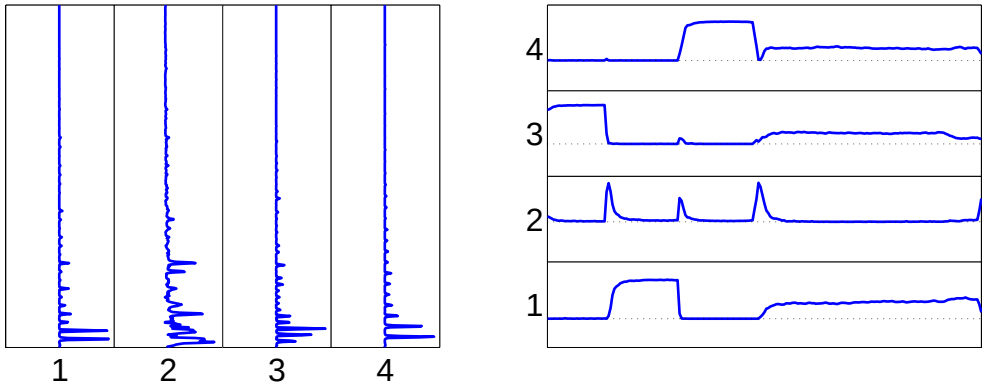


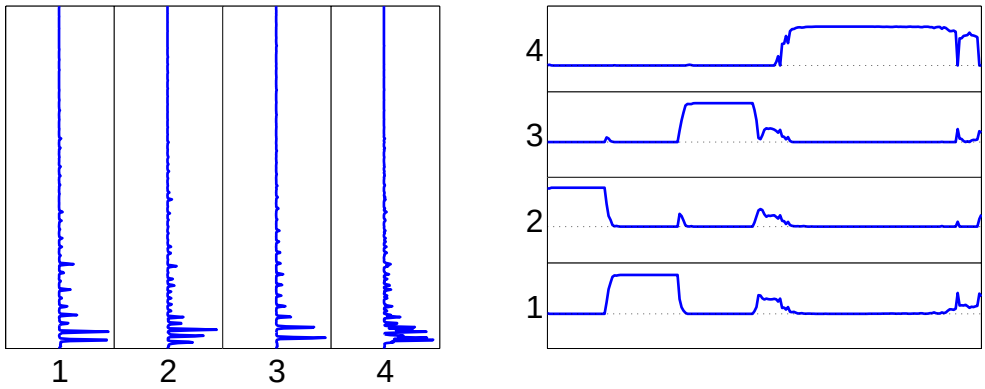
Figure 4·4: Spectrogram of a piano music clip. It represents three notes played successively followed by a chord which is composed of the three notes. The abscissa represents time and the ordinate represents frequency.



(a) Basis Components $P(f|z)$ (left) and Mixture Weights $P_t(z)$ (right) learned with no sparsity.



(b) Basis Components $P(f|z)$ (left) and Mixture Weights $P_t(z)$ (right) learned with no sparsity.



(c) Basis Components $P(f|z)$ (left) and Mixture Weights $P_t(z)$ (right), learned with sparsity imposed on $P_t(z)$.

Figure 4·5: (a) and (b) show 3 and 4 basis components learned from the spectrogram of Figure 4·4. Also shown are the corresponding mixture weights. (c) shows 4 basis components and corresponding mixture weights that were learned by imposing sparsity on the mixture weights.

Now, suppose one would like to have a decomposition in which, in addition to the individual notes, the chord is also extracted as a separate basis component. This is intuitively appealing, since a combination of notes that are harmonic (as is the case in this example) is perceptually recognized as a distinct entity rather than as a combination of distinct sounds. Figure 4·5(b) shows the result of latent variable decomposition where four basis components were extracted. Notice that the additional component, instead of modeling the chord, represents the transitions between the notes. The solution in which the additional component corresponds to the chord is a feasible solution but is no more likely than the solution shown in the figure.

Addition of the sparsity (entropic) prior on the mixture weights $P_t(z)$ provides a way to extract the chord as a separate component. The sparsity constraint implies that mixture weights corresponding to few basis components have values that are significantly above zero. If the value of the sparsity parameter chosen is appropriately high, this constraint forces only one of the basis components to be "active" in any particular time frame. Thus, each of the basis components that are learned end up representing the spectral structure in the time frames in which corresponding mixture weight values are high. This is illustrated by the results shown in Figure 4·5(c) where the four components correspond to the three notes and the chord. The sparsity constraint makes such a solution more likely when compared to all the other feasible solutions.

Another example that illustrates the effect of sparsity is shown in Figure 4·6. The model was used to analyze handwritten digits from the USPS Handwritten Digits database[11]. Twenty-five basis components were extracted by analyzing 1000 different instances for every digit, with the additional constraint that the mixture weights be sparse. Each instance of a digit was given by the pixel intensities as a 16×16 matrix. We unwrapped each matrix and treated it as a 256-dimensional vector, representing the set of 1000 vectors as a 256×1000 matrix $\mathbf{V}$. The matrix $\mathbf{V}$ was used as the input to the algorithm. Figure 4·6 shows the extracted components for digits "2" and "3." Notice the qualitative difference in

[11] from http://www.cs.toronto.edu/~roweis/data.html.

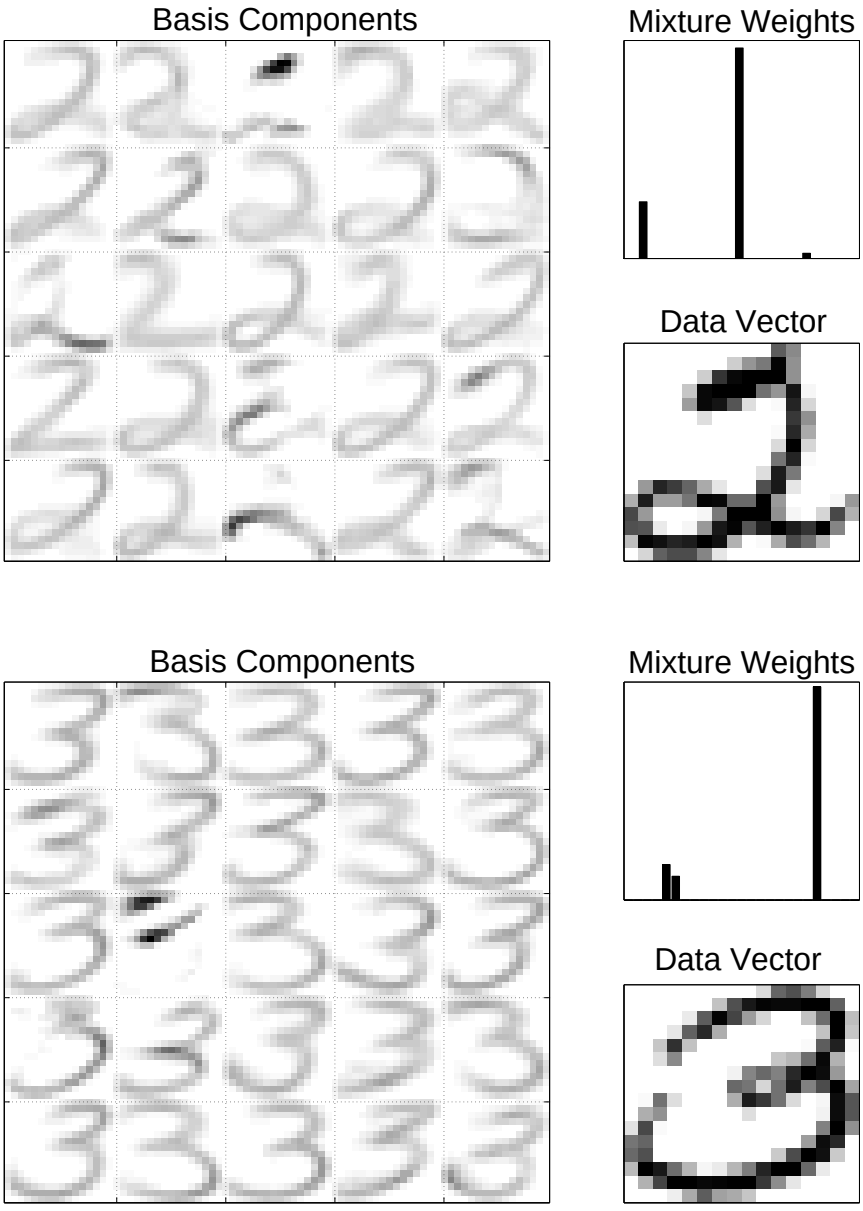


Figure 4·6: Latent Variable Model applied on the USPS Handwritten Digits database, with the additional constraint that mixture weights be sparse. Twenty-five basis components were learned from the data set. Basis components wereextracted for the digits "2" (top) and "3" (bottom), shown in the left panels as 5×5 tiles. The smaller panels on the right show the mixture proportions with which the basis components combine to approximate the input vectors. In this example, we constrained mixture weights to be sparse by imposing a sparsity parameter of 0.2 ($\beta = 0.2$).

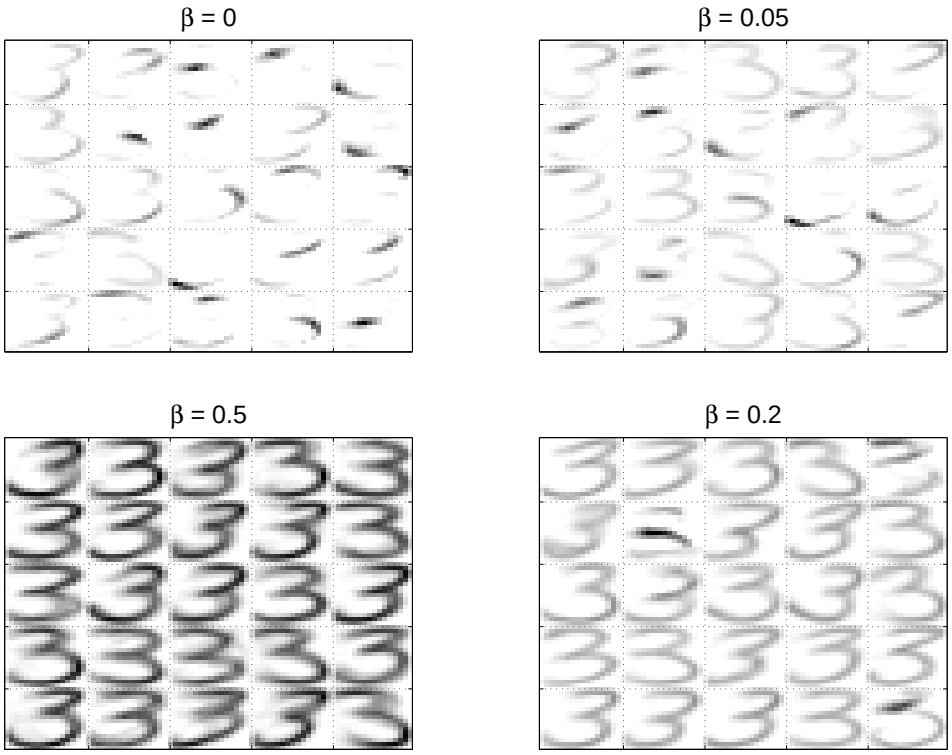


Figure 4·7: Twenty-five basis components learned from training data for class "3" with increasing sparsity parameters on the mixture weights. The sparsity parameter was set to (from top-left in clockwise direction) 0, 0.05, 0.2, and 0.5, respectively. Increasing the sparsity parameter for the mixture weights produces basis components that are more representative of instances of the input set rather than part-like features of the inputs.

the extracted components compared to those on Figure 3·6, demonstrating the qualitative change in the basis components when sparsity constrains the solution.

4.3 Sparse Overcomplete Coding: Geometry

In Section 3.4, we used a data set of 400 3-dimensional multinomials to understand and visualize the geometry of the latent variable model. And in the previous section, we have derived a method to impose sparsity in the framework. We use the same dataset to understand how sparsity makes a difference in the model. Figure 4·8 reproduces the dataset, where each multinomial distribution is represented as a point in the Standard 2-Simplex.

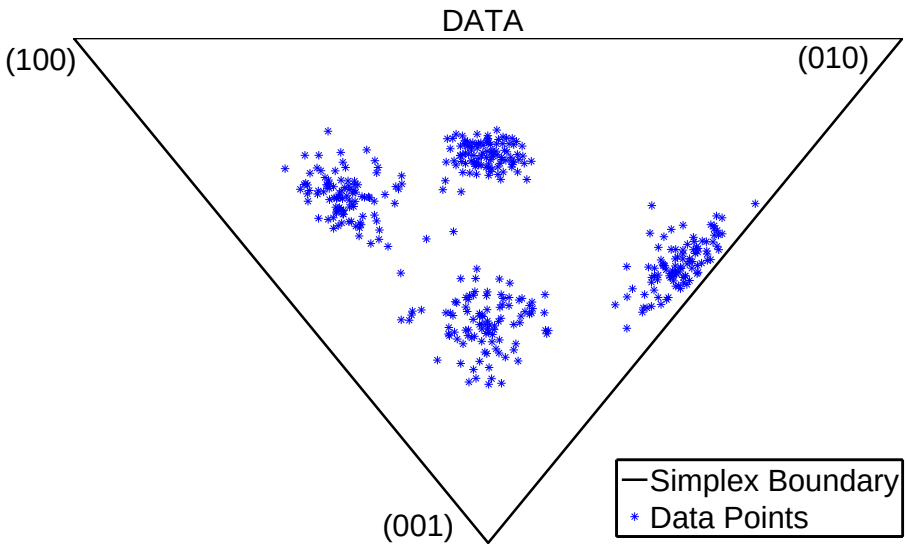


Figure 4·8: 400 3-dimensional multinomial distributions represented in the Standard 2-Simplex.

As mentioned earlier, the problem of overcomplete codes is indeterminacy if sparsity is not imposed. One can still arrive at one of the many feasible solutions. Figure 4·9 shows the effect of increasing the number of basis components in an overcomplete code without imposing sparsity. As the number of basis components increases, the convex hulls formed by the bases "expand" around the data. This larger set of basis components can accurately

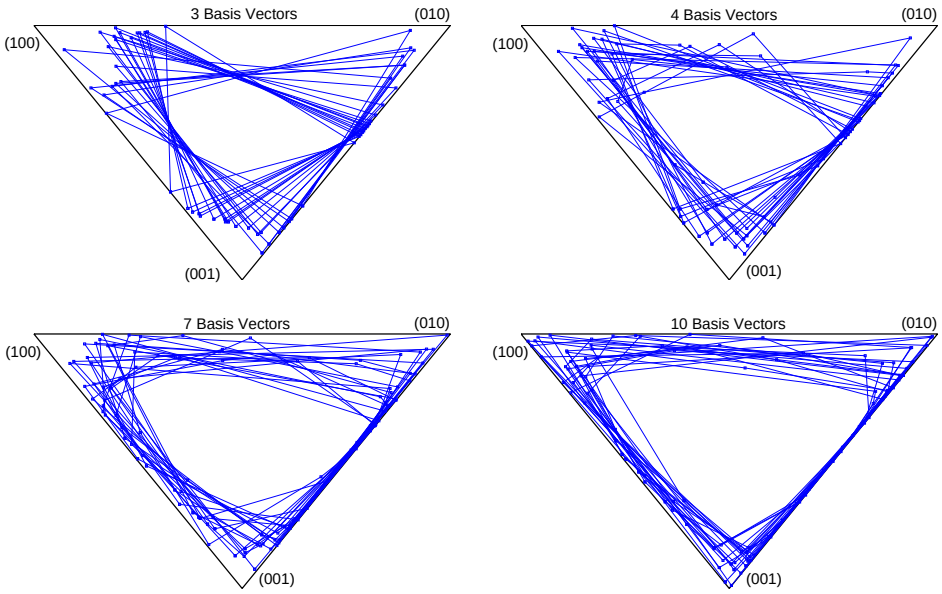


Figure 4·9: Illustration of the effect of number of basis vectors on the latent variable model applied on 3-dimensional distributions. Points are represented within the *Standard 2-Simplex* given by $\{(001), (010), (100)\}$. The model was applied on the data set of 400 points shown in Figure 4·8 to extract 3, 4, 7, and 10 basis components. Each case consisted of 20 repeated runs and the resulting convex hulls formed by the basis components were plotted as shown in the panels from left to right. Notice that increasing the number of basis vectors enlarges the sizes of convex hulls.

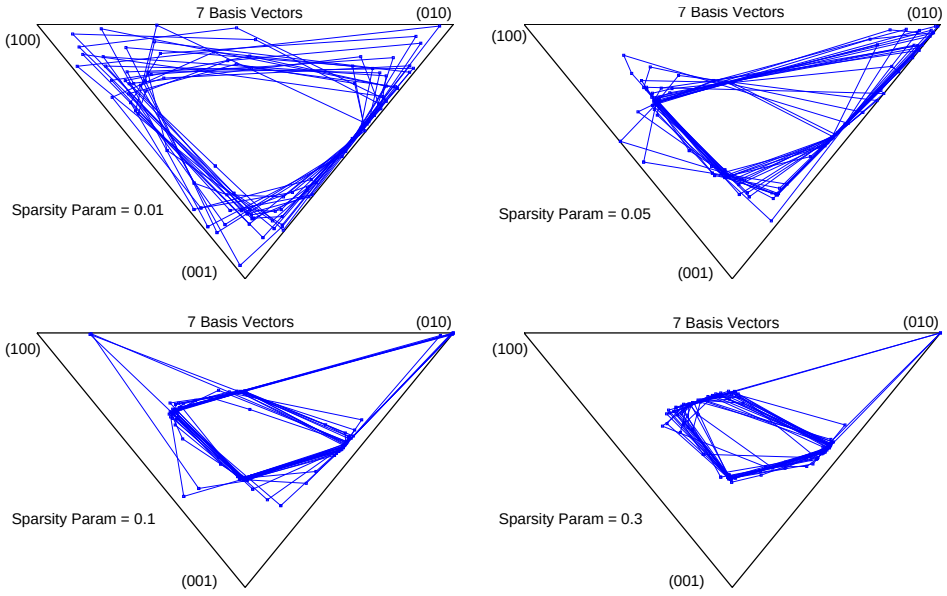


Figure 4·10: Illustration of the effect of sparsity on the latent variable model applied on 3-dimensional distributions. Points are represented within the *Standard 2-Simplex* given by $\{(001), (010), (100)\}$. The latent variable model was applied on data shown in Figure 4·8 to extract 7 basis components with different values of the sparsity parameter on the mixture weights. There were 20 repeated runs for a given value of the sparsity parameter and the resulting convex hulls are plotted as shown. Increasing the sparsity of mixture weights makes the resulting convex hulls more compact. The case when no sparsity was imposed was shown in Figure 4·9.

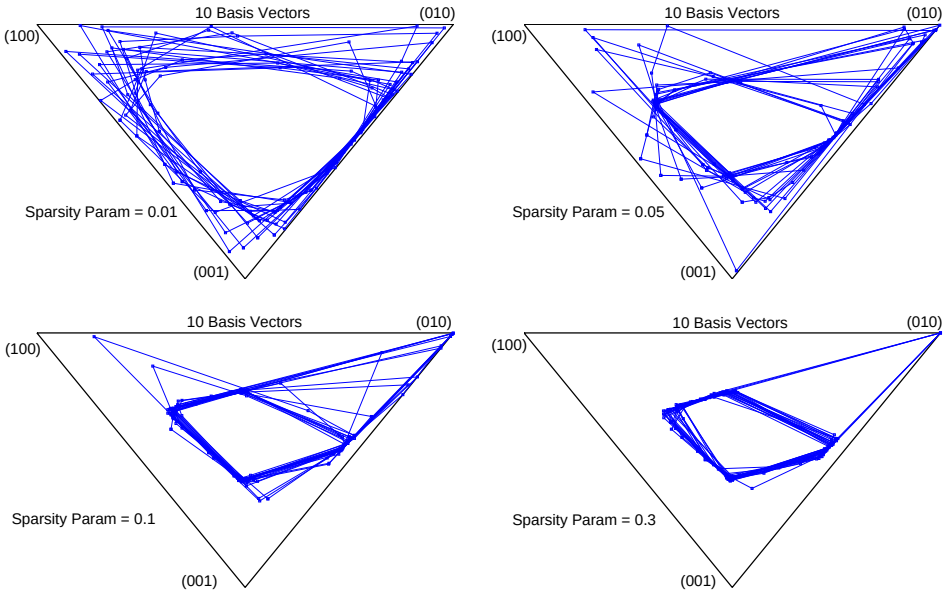


Figure 4·11: Illustration of the effect of sparsity on the latent variable model applied on 3-dimensional distributions. Points are represented within the *Standard 2-Simplex* given by $\{(001), (010), (100)\}$. The latent variable model was applied on data shown in Figure 4·8 to extract 10 basis components with different values of the sparsity parameter on the mixture weights. There were 20 repeated runs for a given value of the sparsity parameter and the resulting convex hulls are plotted as shown. Increasing the sparsity of mixture weights makes the resulting convex hulls more compact. The case when no sparsity was imposed was shown in Figure 4·9.

represent the data but are less characteristic of the distribution of data points. In other words, the new set of basis components is less informative about the data set. Consider the extreme case where we have the set of corners of the 2-simplex as basis vectors. They accurately represent the data set but do not provide any information. This is because they can represent not just this dataset but *any* other data set with perfect accuracy.

However, imposing sparsity on the mixing proportions gives us desirable properties. Figures 4·10 and 4·11 show that as the sparsity parameter is increased, convex hulls formed by the basis components get more "compact" around the data. Since few basis components contribute to any particular data instance, they are more data-like, or in other words, provide more holistic representations of the input space. Figure 4·7 shows 25 basis components extracted from hand-written examples for digit "3." The components become more representative of "3" as the mixture weights become more sparse. In terms of how information is encoded about the distribution of the data, increasing sparsity of the mixture weights *pushes* information from the mixture weights to the basis components. This occurs because reducing the entropy of the mixture weights increases the entropy of the basis components in the model (see Figure 4·13). This pushes the basis components from the corners of the standard simplex towards the center. In the extreme case in which the set of basis components is given by the entire data set itself, all the information is encoded by the basis components, with the mixture weights providing no information.

4.4 Sparse Decomposition for Source Separation

In this section, we explore how sparse latent variable decomposition can be used for source separation with a procedure similar to that presented in Section 3.5. The main difference is in the training stage, where we learn overcomplete sets of basis components by imposing sparsity on the mixture weights. The separation stage remains the same, where using the learned basis components, the remaining parameters of the mixture spectrogram model are estimated using a maximum likelihood formulation.

As before, let $\mathbf{V}$ represent the magnitude spectrogram of a mixture signal. Let $\mathbf{L}^s$

represent the magnitude spectrogram of the training recording for the s-th source, where L^s_{ft} denotes the energy in frequency bin f at time frame t.

4.4.1 Training Stage

In the training stage, we learn basis components, denoted by $P_s(f|z)$, for each source. The model is given by equation (3.15).

For a given source s, the parameters $P_t(z)$ and $P_s(f|z)$ are initialized randomly and reestimated through iterations of the equations derived in Section 4.2.3. Mixture weights are estimated with a positive entropic prior imposed on them. To summarize, the update equations can be written as

$$P_t(z|f) = \frac{P_t(z)P_s(f|z)}{\sum_z P_t(z)P_s(f|z)}, \tag{4.15}$$

$$P_s(f|z) = \frac{\sum_t P_t(z|f)L^s_{ft}}{\sum_t \sum_f P_t(z|f)L^s_{ft}}, \tag{4.16}$$

$$0 = \frac{\omega_z}{P_t(z)} + \beta + \beta \log P_t(z) + \tau_t, \tag{4.17}$$

$$P_t(z) = \frac{-\omega_z/\beta}{\mathcal{W}(-\omega_z e^{1+\tau_t/\beta}/\beta)}, \tag{4.18}$$

where β is the sparsity parameter and ω_z represents $\sum_f L^s_{ft} P_t(z|f)$.

A given training iteration includes one update each of equations (4.15) and (4.16), and 2-5 iterations of the fixed-point equation pair for τ_t, given by equations (4.17) and (4.18).

Only the $P_s(f|z)$ values are used in reconstruction; the rest of the terms are discarded. Figure 4·12 shows examples of a sparse overcomplete set of basis components learned for a female talker. Examples of a compact code, found when sparsity is not imposed on the mixture weights, are also shown for comparison. The sparse overcomplete component solutions exhibit harmonic structure similar to what is observed in speech signals. Figure 4·13 illustrates how the average entropies of basis components and mixture weights vary with different values of the sparsity parameter β. Reducing the entropies of mixture weights increases the entropy of basis vectors. Thus, empirical evidence shows that a set of sparse overcomplete basis components can also be obtained by having a negative entropic prior

on the basis components. This observation, while interesting, is beyond the scope of this thesis, and is left for future work.

4.4.2 Separation Stage

To separate the mixture spectrogram, we use the model presented in Section 3.5.2. The overall distribution underlying the spectral vector for the t-th analysis frame of the mixture spectrogram is given by

$$P_t(f) = \sum_s P_t(s) \sum_{z \in \{\mathbf{z}_s\}} P_t(z|s) P_s(f|z), \tag{4.19}$$

where $P_t(s)$ is the *a priori* probability of the s-th source and $\{\mathbf{z}_s\}$ represents the set of values that z can take for that source.

We do not impose sparsity during separation. Sparsity is used in the training stage to ensure that a large set of basis components is found that can characterize the sources in the training set. In the separation stage, we utilize these learned basis components to approximate the mixture spectrogram. As derived in Section 3.5.3, we estimate the parameters of the model by iterations of equations (3.39), which are reproduced below:

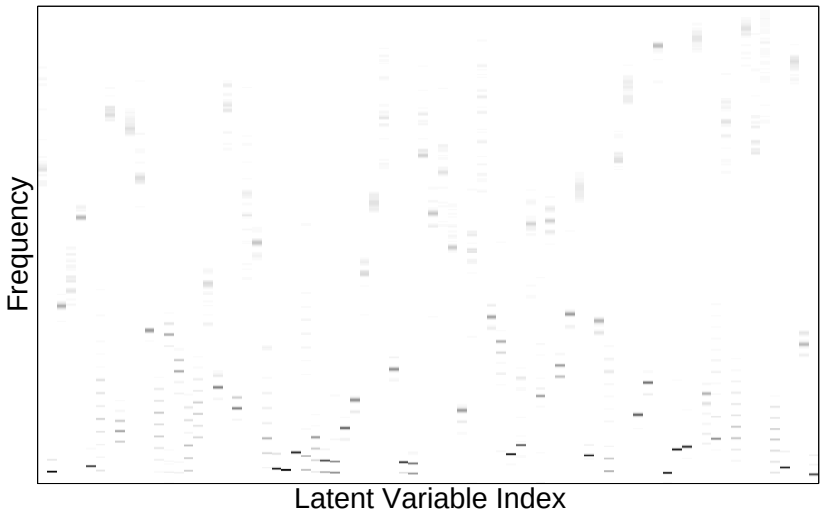
$$\begin{aligned} P_t(s,z|f) &= \frac{P_t(s)P_t(z|s)P_s(f|z)}{\sum_s P_t(s) \sum_{z \in \{\mathbf{z}_s\}} P_t(z|s)P_s(f|z)} \\ P_t(s) &= \frac{\sum_{z \in \{\mathbf{z}_s\}} \sum_f P_t(s,z|f) V_{ft}}{\sum_s \sum_{z \in \{\mathbf{z}_s\}} \sum_f P_t(s,z|f) V_{ft}} \\ P_t(z|s) &= \frac{\sum_f P_t(s,z|f) V_{ft}}{\sum_{z \in \{\mathbf{z}_s\}} \sum_f P_t(s,z|f) V_{ft}}. \end{aligned}$$

The spectrogram of the s-th source can be estimated as

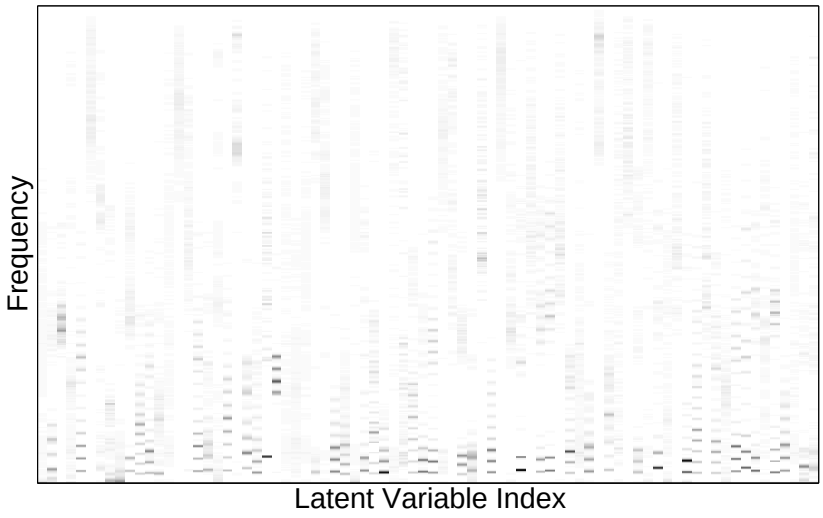
$$\hat{V}_{ft}(s) = \frac{P_t(s)P(f|s)}{\sum_s P_t(s)P(f|s)} V_{ft}, \tag{4.20}$$

where $P_t(f|s)$ is given by

$$P_t(f|s) = \sum_{z \in \{\mathbf{z}_s\}} P_t(z|s) P_s(f|z). \tag{4.21}$$



(a) Compact code



(b) Sparse Overcomplete Code

Figure 4·12: (a) A subset of 80 basis vectors from a total of 160 basis components learned for a female talker. Sparsity was not imposed on the mixture weights during estimation. (b) A subset of 80 basis components out of a total of 1000 learned basis components for the same talker. Sparsity was imposed ($\beta = 0.3$) on the mixture weights during estimation. Darker values of the grayscale correspond to higher probabilities.

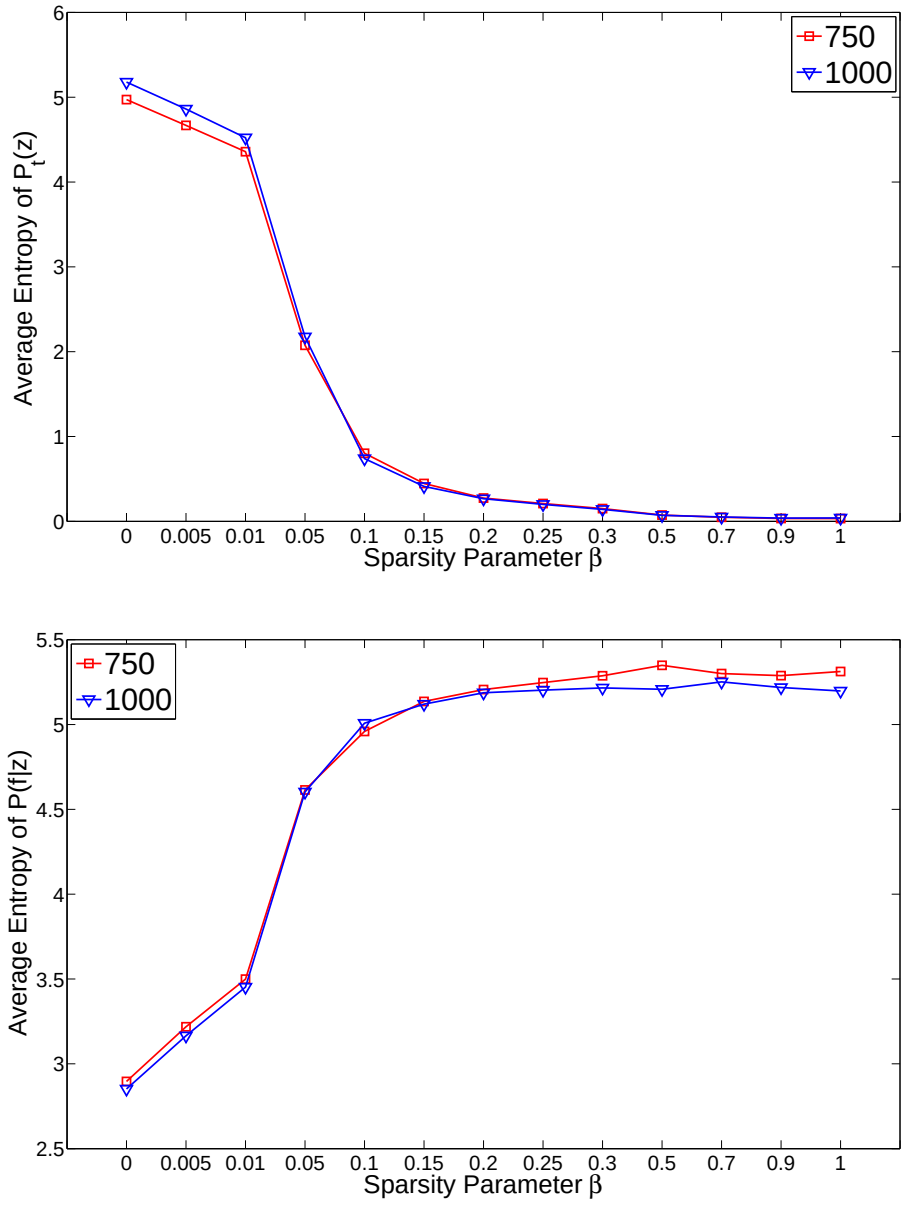


Figure 4·13: Illustration of how enforcing sparsity on the mixture weights changes the average entropy of the mixture weights (top panel) and the basis components (bottom panel). Average entropy was calculated during the training stage when 750 (red squares) and 1000 (blue triangles) basis components were learned. Notice that decreasing entropy of the mixture weights is equivalent to increasing the entropy of the basis vectors. Using a set of overcomplete basis components can be considered more "expressive" in the information-theoretic sense.

Once the values are estimated for all f and t, the phase of the short-time Fourier transform of the mixed signal is combined with the estimated magnitude spectrogram. An inverse Fourier transform is performed to obtain the time domain reconstruction of the source.

4.4.3 Separation Results

Experiments evaluated the separation performance for the proposed method on synthetic mixtures. Experiments were done on the same set of talkers used in Section 3.5.4. We used six pairs of talkers chosen from the Wall Street Journal (WSJ) database: two pairs were female/male, two were male/male and two were female/female.

A set of 134 utterances comprising approximately 16 minutes of speech was separated as training data for each talker. Signals were sampled at 16 kHz and short-term Fourier transforms were generated with an FFT point size of 1024, hop size of 256 between frames, and a Hanning window. The dimensionality of each spectral vector was 513 ($F = 513$). For the overcomplete case, 750 or 1000 basis components were learned for different values of the sparsity parameter β. The set of values used for β is given by the set $\{0.005, 0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.5, 0.7, 0.9\}$.

For a given pair of talkers, mixed signals were obtained by digitally adding different pairs of test signals. The length of the mixed signal was set to the shorter of the two signals. The component signals were all normalized to 0 mean and unit variance prior to addition, resulting in 0 dB SNR for each talker. A set of five mixed recordings were obtained as test cases for every talker pair considered. Figure 4·14 shows example spectrograms of reconstructions from a mixture with male and female talkers. In this case, a sparsity parameter of $\beta = 0.3$ was used in the training to estimate an overcomplete set of 1000 basis components. For evaluating the quality of separation, SNR (equation 3.44) and SER (equation 3.45) introduced in Section 3.5.4, were computed.

Figures 4·15 and 4·16 illustrate the effect of changing the sparsity parameter on separation. Experiments were conducted on two test mixture signals belonging to a Male/Female talker pair. Different values of the sparsity parameter was used during the training phase.

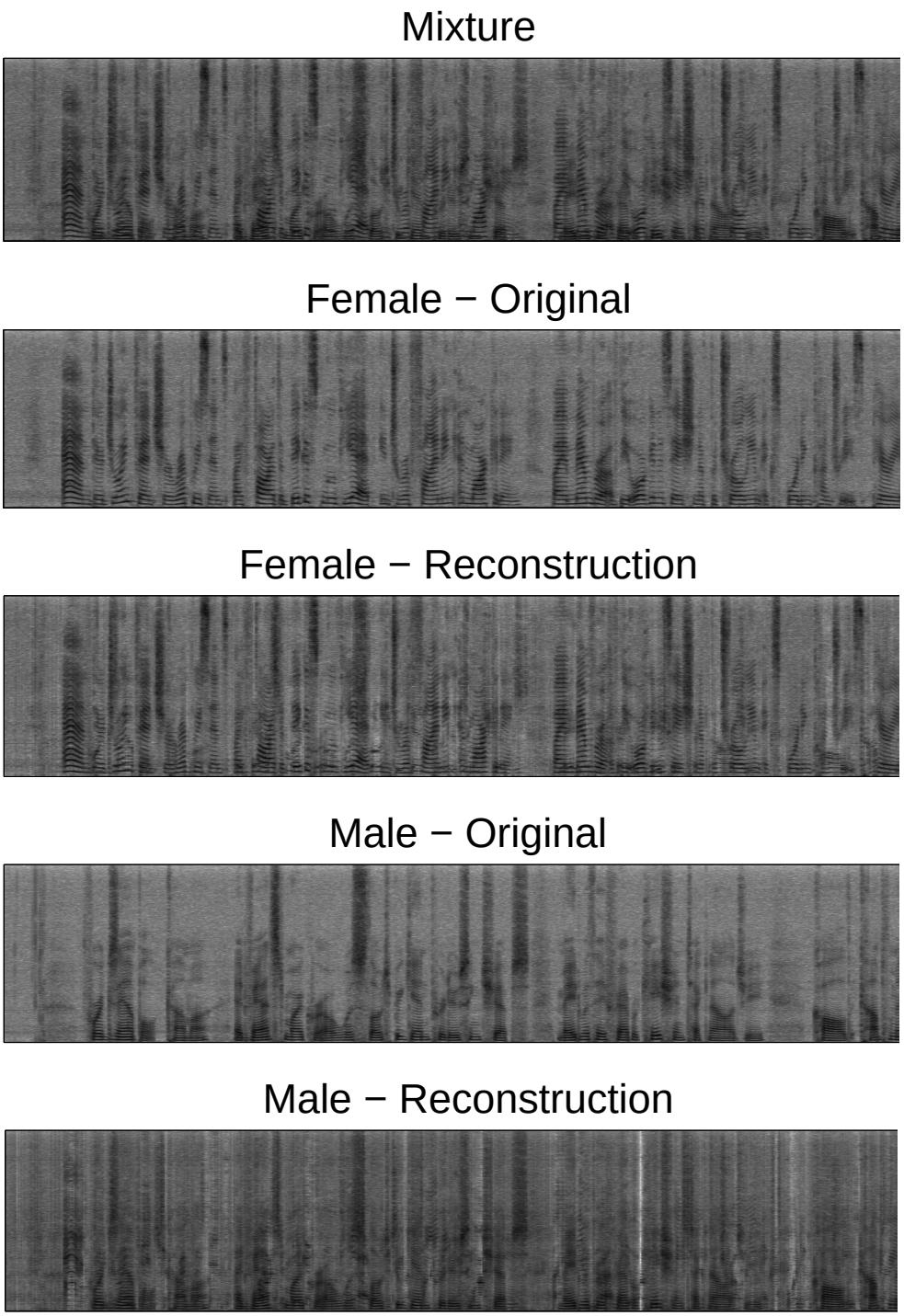


Figure 4·14: Result of a separation experiment for a male/female talker pair, with 1024 point FFT size and 1000 basis components ($\beta = 0.3$). The SNR and SER improvements for the female were 8.1208 dB and 5.7684 dB respectively. For the male, improvements were 8.1320 dB and 5.7681 dB.

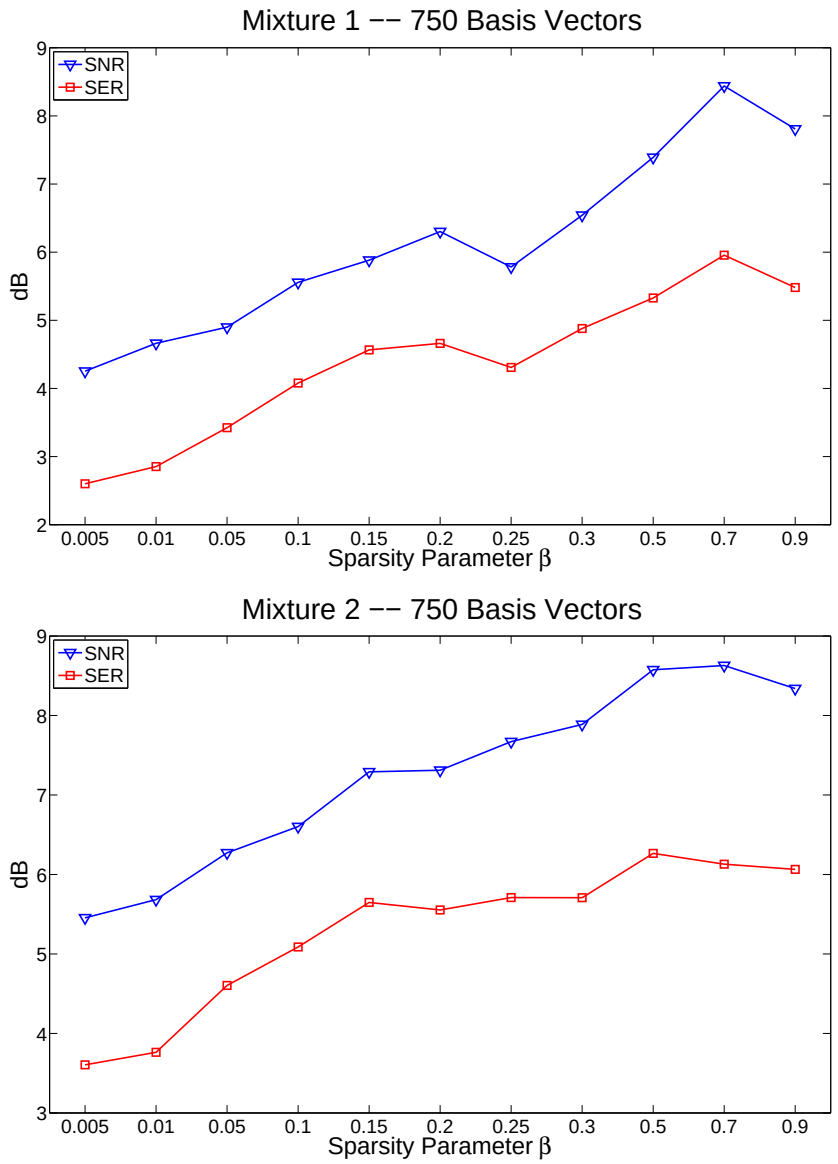


Figure 4·15: Results of separation experiments that illustrate the effect of the value of the mixture weight sparsity parameter β on the quality of separation. Overcomplete sets of 750 basis components were extracted with the sparsity (entropic) prior and separation (Male/Female talker pair) was performed. The panels display the average SNR and SER values of the reconstructed signals. The top and bottom panels correspond to two different test mixtures.

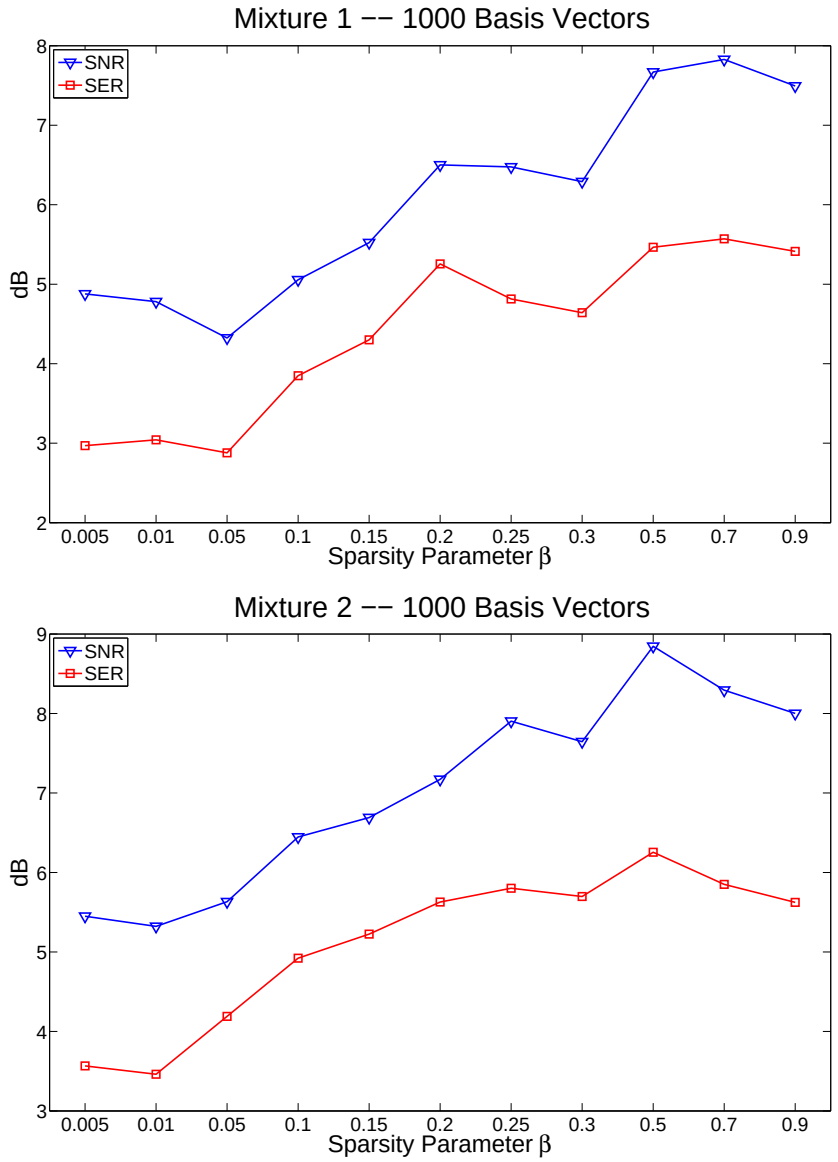


Figure 4·16: Results of separation experiments that illustrate the effect of the value of the mixture weight sparsity parameter β on the quality of separation. Overcomplete sets of 1000 basis components were extracted with the sparsity (entropic) prior and separation (Male/Female talker pair) was performed. The panels display the average SNR and SER values of the reconstructed signals. The top and bottom panels correspond to two different test mixtures.

The SNR/SER improvements of the separated signals are plotted in the figures. There is a general trend for separation to improve with increasing values of the sparsity parameter.

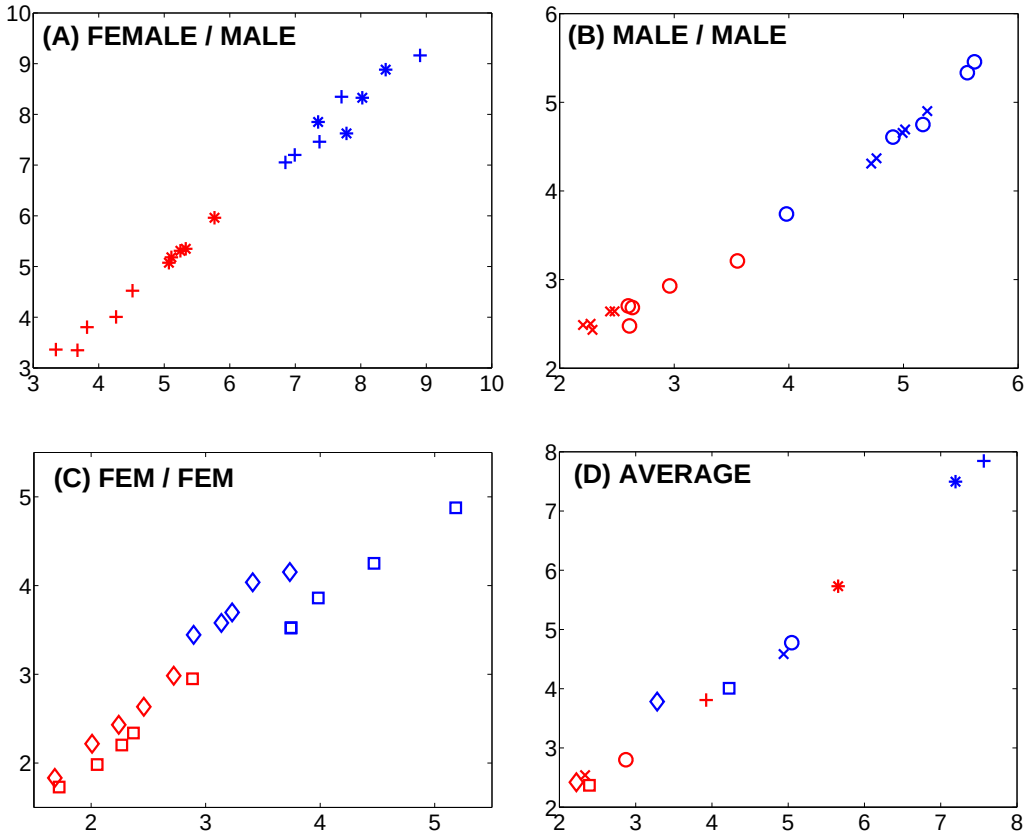


Figure 4·17: Talker separation evaluation results in terms of SNR (Signal/Noise Ratio) improvements (in dB) for the sparse overcomplete code (in Blue) and for the compact code (in Red). The Y-axis corresponds to the SNR improvement for the first talker while the X-axis represents the improvement for the second talker. Each point corresponds to a particular experiment. Different symbols represent different talker pairs in the mixtures. Each point in Panel (D) is the average of the corresponding points in the first three panels. Notice that the sparse code consistently performs better than the compact code.

Figures 4·17 and 4·18 summarize results of experiments comparing performance for the sparse overcomplete code and the compact code. The compact code corresponded to a set of 100 basis components estimated without the imposition of sparsity, while the sparse

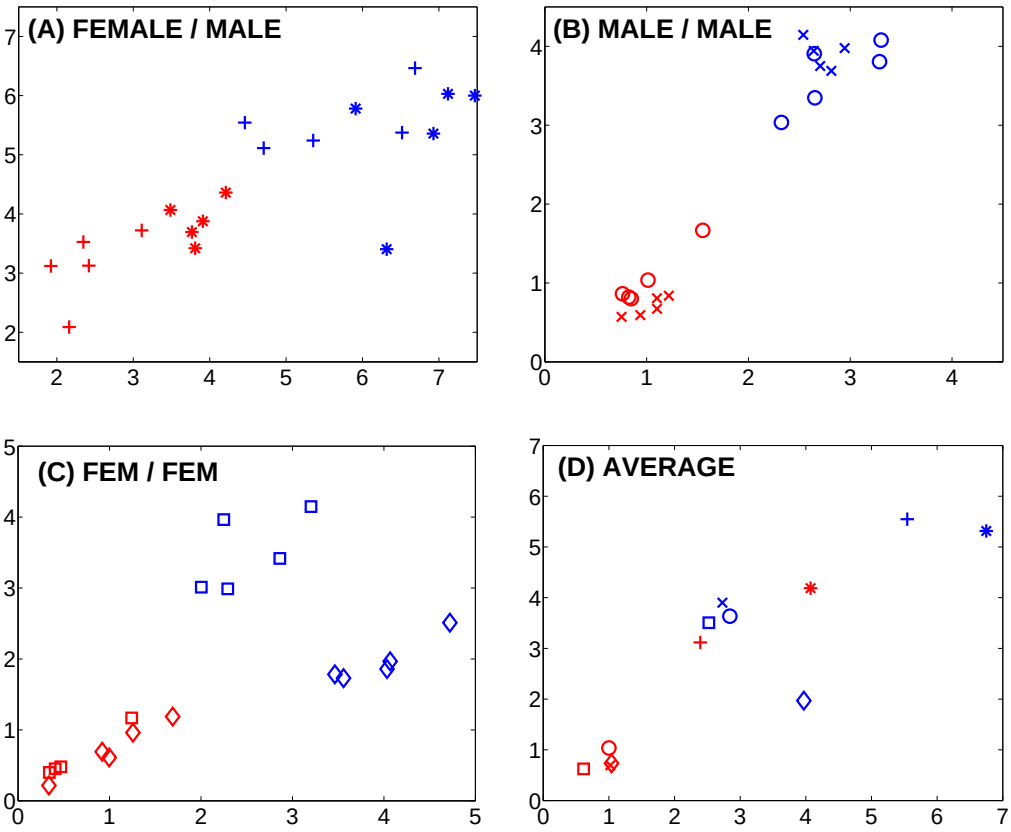


Figure 4·18: Talker separation evaluation results in terms of SER (Speaker Energy Ratio) improvements (in dB) for the sparse overcomplete code (in Blue) and for the compact code (in Red). The Y-axis corresponds to the SER improvement for the first talker while the X-axis represents the improvement for the second talker. Each point corresponds to a particular experiment. Different symbols represent different talker pairs in the mixtures. Each point in Panel (D) is the average of the corresponding points in the first three panels. Notice that the sparse code consistently performs better than the compact code.

overcomplete code used a set of 1000 basis vectors estimated with a sparsity parameter of $\beta = 0.7$. For every talker pair, separation was evaluated using both the sparse overcomplete code and the compact code for five different mixtures. Figure 4·17 plots the SNR improvements of the two reconstructed signals against each other, while Figure 4·18 plots the speaker energy ratios. Every point in the figures corresponds to the result of one experiment. Points in blue correspond to results with the sparse code while points in red correspond to results from the compact code. All the results for a given talker pair are represented by the same symbol; different symbols represent different talker pairs. Results show that the sparse code performs significantly and consistently better than the compact code, for both metrics. A few examples of the separated signals can be obtained at `http://cns.bu.edu/~mvss/courses/speechseg/`.

4.5 Other approaches to Sparsity

Finally, this section presents a brief overview of other approaches to sparsity that have been used in the literature. Sparsity has been used in techniques that model inputs as a linear combination of bases (kernels) and mixture weights. Two groups of techniques are presented: approaches motivated by neural coding theory and approaches motivated by machine learning applications.

4.5.1 Neural Coding Theory

Sparse coding is thought to be a fundamental principle driving biological sensory and neural systems to encode and process sensory information (Kanerva, 1988; Field, 1994; Olshausen and Field, 1996). Several theoretical, computational and experimental studies suggest that neurons encode sensory information using a small number of active neurons at any given point in time (Olshausen and Field, 2004) as a way to minimize the metabolic energy or cost of encoding information. Furthermore, sensory systems are thought to transform the input into a code that reduces the redundancy among the elements of the input stream, following the ideas of Attneave (1954) and Barlow (1959, 1961), who argue that the principles

information theory can be used to understand perceptual processes.

Consider basis decomposition models that have been proposed to understand sensory coding (Olshausen, 2001; Lewicki, 2002; Smith and Lewicki, 2006). The data vector $\mathbf{v}$ (or the underlying generative distribution in the case of a latent variable model) is approximated as $\mathbf{Wh}$, where the columns of $\mathbf{W}$ are basis components and the elements h_i of the vector $\mathbf{h}$ are the mixture weights. In this context, this goal of *efficient coding* is equivalent to finding a set of basis vectors that forms a complete code (i.e., spans the input space) and results in mixture weights that are as statistically independent as possible, given an ensemble of inputs. One way of achieving this, as suggested by Field (1994), is to have a representational scheme in which only a few (out of a large population) of the basis components are required to explain any particular data vector. Such a representational scheme is referred to as a *sparse code*. As Olshausen and Field (1996) explain, the existence of any statistical dependencies among a set of variables h_i may be discerned whenever the the joint entropy is less than the sum of the individual entropies (i.e., $\mathcal{H}(h_1, h_2, \ldots, h_r) < \sum_i \mathcal{H}(h_i)$, where $\mathcal{H}$ is the entropy). A possible strategy for reducing statistical dependencies is to lower the individual entropies $\mathcal{H}(h_i)$. Thus, reducing entropies of mixture weights is equivalent to having a sparse code of basis components. In the approach presented in this chapter, entropy is reduced directly by making use of the entropic prior on the mixture weights.

4.5.2 Machine Learning

Component-wise decompositions have played an important role in machine learning applications. Popular techniques include Principal Components Analysis (PCA), Independent Components Analysis (ICA), Non-negative Matrix Factorization (NMF), and others. These techniques express inputs as mixtures of data-dependent components that are learned during estimation. We focus on the latter technique, NMF, and briefly review research work that has attempted to extend NMF by incorporating sparsity[12].

[12]Sparse extensions to PCA and ICA have been proposed: see (Zou et al., 2004) and (Zibulevsky and Pearlmutter, 2001), respectively, for examples. Also, there is a body of literature on sparse representation of signals using a known dictionary such as Fourier Bases or wavelets. These approaches use algorithms like basis pursuit (Chen et al., 2001) and linear/quadratic programming (Donoho and Elad, 2003; Fuchs,

One of the important properties of NMF is that it usually produces a sparse representation of the data. Lee and Seung (2001) point out that basis vectors of NMF used in distributed, yet sparse combinations generate expressiveness in the reconstructions. However, as Hoyer (2004) points out, sparsity given by NMF is a side-effect rather than a goal of the algorithm. This idea is supported by recent research work aimed at developing sparse versions of NMF (Eggert and Korner, 2004; Heiler and Schnorr, 2006; Hoyer, 2004; Morup and Schmidt, 2006; Pascaul-Montano et al., 2006). Hoyer (2004) introduced an algorithm that used a sparsity measure based on L_1 and L_2 norms and used this sparsity measure to derive a projected gradient descent algorithm. Update equations were derived by minimizing the Euclidean distance measure between data and the reconstruction. Instead of the L_1 norm, Morup and Schmidt (2006) and Eggert and Korner (2004) use a general function of the mixture weight matrix $\bar{\mathbf{H}}$ as a penalization term during estimation. They suggest that any function with a positive derivative can be used as a penalty term. While Eggert and Korner (2004) use an objective function based on the Euclidean distance to derive the updates, Morup and Schmidt (2006) use a KL-distance measure. However, neither paper proves that the update equations converge to a solution. Pascaul-Montano et al. (2006) take a different approach by imposing a multiplicative smoothing matrix during estimation that enforces sparsity.

While several other extensions exist, the studies mentioned above are representative of the various approaches. The review, however, does not constitute an exhaustive survey of sparse extensions to NMF.

4.6 Conclusions

This chapter introduced an important extension to the latent variable framework. The framework from the previous chapter is limited by a restriction on the number of components that can be learned. Here, a learning formulation that addresses this limitation is derived that utilizes the notion of sparsity. An entropic prior in a maximum a posteriori formulation

2004) for L_1 norm minimization to obtain sparse representations. This work is beyond the scope of work reviewed in this thesis.

enforces sparsity. A geometric interpretation of the model was presented using an artificial dataset. Enforcing sparsity in the framework enables one to learn an overcomplete set of latent components which can better characterize the data. Inference algorithms were derived, and the framework was applied to the problem of source separation. Experimental evidence of the utility of such sparse overcomplete representations was presented. The sparse decomposition framework presented in this chapter and in the previous chapter is general in its scope and applicable to data other than acoustic spectrograms. For instance, Appendix B presents three applications of the framework for analyzing image data.

Chapter 5

Conclusions

5.1 Thesis Overview

The cocktail party effect is a challenging problem from a computational perspective. One fundamental question is whether it is possible to build a machine capable of solving the cocktail party problem in a satisfactory manner. A simpler formulation of the question would be whether machines can identify/separate sources from the acoustic signal of an auditory scene. Many researchers have attempted to answer this question and build such an automatic system. This thesis represents a step towards that goal.

The focus of this thesis is single-channel audio. There has been relatively little work on modeling single-channel sounds compared to the large body of work on blind-source separation with multiple input channels. Dealing with just one signal of an acoustic event instead of many makes the problem formulation simpler, albeit harder since there is much less information with which to work. Instead of building *a* system that solves a particular problem, a framework for modeling single-channel audio was developed, based on probability theory, which is the natural language to express uncertainty.

Specifically, a latent variable model was proposed to model time-frequency representations (eg. spectrograms), where the energy in a time-frequency bin is treated as a histogram count of multiple draws. A variant of a specific kind of latent variable model called "latent class models" were used, built on the principle of local independence (or the common cause criterion). This a powerful formulation and has been used to extract latent structure from data in a variety of fields such as social sciences (latent structure analysis), analysis of text corpora (latent semantic analysis), and machine learning (non-negative matrix

factorization), among others. The formulation allowed the underlying distribution of each spectral vector to be modeled as a mixture of multinomial distributions. The component multinomial distributions were assumed to be the same for all spectral vectors of a given source, while the proportions with which they combined to generate a particular vector differed from frame to frame. The intuition was that these latent components are learned so that they characterize the source, and not individual spectral vectors. Chapter 3 presented the theory and derived inference algorithms to realize the approach. Experimental evidence of the applicability of the proposed framework to single-channel audio processing, by demonstrating source separation and denoising experiments, was also presented.

An important limitation of the proposed framework is that the number of latent components that can be extracted is limited by the dimensionality of the input space, which, in the context of time-frequency representations, is given by the number of frequency bins. The number of components required to model a complex sound signal potentially could be large and should not be limited by the arbitrary choice of representation. To overcome the limitation, an extension employing the concept of sparsity was presented (Chapter 4). An entropic prior in a maximum a posteriori formulation was used to enforce sparsity. Lowering entropy of the mixture weights to extract an overcomplete set of basis components results in components that are more "expressive" and better characterize the data. The theory and inference algorithms were presented in Chapter 4 along with experiments that provided evidence of the utility of such sparse-overcomplete representations for single-channel audio processing applications.

To summarize, a probabilistic latent variable framework to model single-channel auditory signals was developed. The statistical framework makes the proposed models amenable to principled extensions and improvements. One such extension, incorporating sparsity by employing the entropic prior, demonstrated the advantages of the extension. More generally, the extension demonstrates how the method can be extended to impose known or hypothesized structure about the data by utilizing prior distributions on the parameters, thus pointing to other possible extensions of the general framework proposed.

5.2 Future Work

The work presented in this thesis points to several extensions, some of which are mentioned below.

Representation

Here, magnitude STFTs were used as inputs for evaluating the performance of the framework. As mentioned previously, one can utilize other representations and compare performance, including TF representations with a log-frequency spacing (Brown, 1991) and TF representations that are physiologically motivated (Patterson et al., 1995; Irino and Patterson, 1997). The possibility of utilizing multidimensional generalizations of the framework (PLCA) to analyze more sophisticated representations of sound such as correlograms (Slaney and Lyon, 1990) and higher-order spectral representations (Nikias and Petropulu, 1993) can also be explored.

The current work ignores the phase information of the sounds and the mixture during analysis. However, studies have shown that phase-spectra carry rich information that can be utilized. For example, experiments by Alsteris and Paliwal (2005) suggest that magnitude spectra can be uniquely reconstructed from phase spectra, although recovering phase from magnitude spectra is not feasible. Future work should explore how to utilize phase information in the present framework. Other avenues include extending the framework to handle multimodal signals such as audio-visual signals.

Model and Theory

The probabilistic foundation of the proposed framework allows it to be easily extended. Specifically, the framework allows one to impose structure on the data by employing prior distributions. The methods proposed so far do not explicitly model the structure present in the mixture weights in a way that captures correlations. In other words, the approach does not model how the basis components co-occur to generate a given spectral vector. One could impose various priors to model this explicitly. The most straightforward choice for

modeling multinomials is the Dirichlet distribution (Minka, 2003) - a conjugate prior for the multinomial distribution (Blei et al., 2003). However, our experiments with the Dirichlet prior (Raj et al., 2006) did not result in significant improvements in source separation. Other choices for the prior include mixture Dirichlet distributions (Bouguila et al., 2004) and the logistic normal distribution (Blei and Lafferty, 2006a). The next step would be to explicitly model the time structure by using hidden Markov models, or other dynamic models (e.g., Blei and Lafferty, 2006b). Secondly, the latent components of the framework can be modeled further in a hierarchical way. One can use existing approaches such as Gaussian mixtures to model each component separately.

In terms of the learning paradigm, the proposed framework is not discriminative in nature. For the source separation problem that we have formulated as a supervised learning problem, it would be more beneficial if the source-dependent components can be learned in a discriminative fashion. This would be especially helpful in cases where the sources present in the mixture exhibit similar spectral structure. Preliminary experiments were conducted that explicitly modeled structure common to both sources of the mixture by learning a separate set of "common basis components" from training data of both sources. This approach yielded marginally better separation and is worthy of further research. Another approach is to enforce a prior during learning that increases the "distance" (in latent variable space) between the sets of components of the different sources. One possibility is to use the concept of independence between sets of vectors, as has been done with Independent Subspace Analysis (Hyvarinen and Hoyer, 2000). If this approach is successful, it opens the possibility of utilizing the approach in an unsupervised framework to learn and separate sources from the mixed signal, obviating the necessity of a training stage.

Related to the above approach is the question of how sparse decomposition relates to ICA. Experiments and empirical results suggest that entropy manipulation of the parameters in the proposed framework produces results similar to non-negative ICA algorithms (Plumbley, 2003). More theoretical analysis is required to fully understand the relationship between these approaches. Another approach that is related to the work presented here

is the emerging field of "compressed sensing" (Candes, 2006; Candes and Tao, 2006). The idea is that it is possible to reconstruct signals accurately from a number of samples which is far smaller than the signal resolution (e.g., reconstructing an image from fewer number of samples than the number of pixels in the image). Research in this field utilizes sparsity, L1 norm minimization, and related concepts. Methods of obtaining sparse codes presented in this work might find applicability in compressed sensing and should be explored further.

And finally, there is room for improvements and analyses of the inference algorithms used to find solutions. Specifically, one can consider alternatives and improvements of the EM algorithm, such as tempered EM, to improve the rate of convergence and the quality of the found solutions.

Applications

This thesis focused on the application of the framework to audio source separation problems. However, it can also be used for other applications, including music transcription, auditory scene analysis, denoising, bandwidth expansion, speaker recognition, audio classification, and more. We have also mentioned that the framework is more general and demonstrated its utility for three image processing applications. Applications of the framework to analyze data in other domains, such as data-mining, brain imaging, text semantic analysis, radiology, chemical spectral analysis, etc., should be explored.

5.3 Concluding Comments

A general probabilistic framework for analyzing multi-dimensional non-negative data was developed. Future researchers should utilize this framework and extend it further to applications in other fields and domains. Specifically, this work may spawn research efforts to build a machine with "auditory awareness" of its surroundings.

Appendix A

Latent Variable Model: Inference for a Mixture Spectrogram

Section 3.5.2 presented the latent variable model for a mixture spectrogram. The model is given by equation (3.38), as reproduced below:

$$P_t(f) = \sum_s P_t(s) \sum_{z \in \{\mathbf{z}_s\}} P_t(z|s) P_s(f|z), \tag{A.1}$$

where $P_t(f)$ is the overall distribution underlying the t-th analysis frame of the mixture spectrogram, $P_t(s)$ is the *a priori* probability of the s-th source, $P_s(f|z)$ is the z-th basis component for the s-th source, $P_t(z|s)$ is the corresponding mixture weight, and $\{\mathbf{z}_s\}$ represents the set of values that z can take for that source. Let $\mathbf{V}$ represent the observed mixture spectrogram where V_{ft} represents the energy in the f-th frequency bin and the t-th analysis frame.

In this appendix, we derive update equations for the parameters of the above model. There are two latent variables in the model – z reflects the index of the latent basis vector and s reflects the source being considered. Following the approach presented in Section 3.3.2, we use a maximum likelihood formulation and derive Expectation Maximization update rules for the parameters.

For the E-step, we obtain *a posteriori* probability for the latent variables as

$$P_t(s, z|f) = \frac{P_t(s) P_t(z|s) P_s(f|z)}{\sum_s P_t(s) \sum_{z \in \{\mathbf{z}_s\}} P_t(z|s) P_s(f|z)}. \tag{A.2}$$

It should be understood that the variable z, when it occurs in the terms $P_t(z|s)$ and $P_s(f|z)$, belongs to the set $\{\mathbf{z}_s\}$ of latent variables that corresponds to the particular source s.

In the M-step, we maximize the expected complete data log-likelihood. Let Λ represent the set of parameters of the model, i.e. $\Lambda = \{P_t(s), P_s(f|z), P_t(z|s)\}$. The expected log-likelihood can be written as

$$\mathcal{L} = E_{\{\bar{s},\bar{z}\}|\bar{f};\Lambda} \log P(\bar{f}, \bar{z}, \bar{s}), \tag{A.3}$$

where $\bar{f}$, $\bar{z}$ and $\bar{s}$ represent the set of all observations of f, z and s in the draws that generated the observed spectral vectors. The complete data likelihood can be written as

$$P(\bar{f}, \bar{z}, \bar{s}) \propto \prod_{j,t} P_t(f_j, z_j, s_j) = \prod_{j,t} P_t(s_j) P_t(z_j|s_j) P_{s_j}(f_j|z_j), \tag{A.4}$$

where f_j, z_j and s_j are the observed values of f, z and s respectively in the j-th draw. The function $\mathcal{L}$ can be written as (ignoring constant terms)

$$\begin{aligned}
\mathcal{L} &= E_{\{\bar{s},\bar{z}\}|\bar{f};\Lambda} \sum_{j,t} \log P_t(f_j, z_j, s_j) \\
&= \sum_{j,t} E_{\{s_j,z_j\}|f_j;\Lambda} \log P_t(f_j, z_j, s_j) \\
&= \sum_{j,t} E_{\{s_j,z_j\}|f_j;\Lambda} \log P_t(s_j) + \sum_{j,t} E_{\{s_j,z_j\}|f_j;\Lambda} \log P_t(z_j|s_j) \\
&\quad + \sum_{j,t} E_{\{s_j,z_j\}|f_j;\Lambda} \log P_{s_j}(f_j|z_j) \\
&= \sum_{j,t} \sum_{z,s} P_t(s, z|f_j) \log P_t(s) + \sum_{j,t} \sum_{z,s} P_t(s, z|f_j) \log P_t(z|s) \\
&\quad + \sum_{j,t} \sum_{z,s} P_t(s, z|f_j) \log P_s(f_j|z).
\end{aligned}$$

In the above equation, the summation over draws j can be changed to a summation over frequencies f by accounting for how many times f was observed, i.e. the f-th entry

of the observed spectral vector V_{ft}[13]. The expected log-likelihood can now be written as

$$\begin{aligned}\mathcal{L} &= \sum_t \sum_f \gamma V_{ft} \sum_s \sum_{z\in\{\mathbf{z}_s\}} P_t(s,z|f)\log P_t(s) \\ &+ \sum_t \sum_f \gamma V_{ft} \sum_s \sum_{z\in\{\mathbf{z}_s\}} P_t(s,z|f)\log P_t(z|s) \\ &+ \sum_t \sum_f \gamma V_{ft} \sum_s \sum_{z\in\{\mathbf{z}_s\}} P_t(s,z|f)\log P_s(f|z). \end{aligned} \tag{A.5}$$

In order to take care of the normalization constraints, the above equation must be augmented by appropriate Lagrange multipliers ϕ_t, τ_t^s and ρ_z^s, yielding

$$\begin{aligned}\mathcal{Q} &= \mathcal{L} + \sum_t \phi_t\Big(1 - \sum_s P_t(s)\Big) + \sum_s \sum_t \tau_t^s\Big(1 - \sum_{z\in\{\mathbf{z}_s\}} P_t(z|s)\Big) \\ &+ \sum_s \sum_z \rho_z^s\Big(1 - \sum_f P_s(f|z)\Big). \end{aligned} \tag{A.6}$$

Maximization of $\mathcal{Q}$ with respect to $P_t(s)$, $P_t(z|s)$ and $P_s(f|z)$ leads to the following set of equations:

$$\sum_f \gamma V_{ft} \sum_{z\in\{\mathbf{z}_s\}} P_t(s,z|f) + \phi_t P_t(s) = 0 \tag{A.7}$$

$$\sum_f \gamma V_{ft} P_t(s,z|f) + \tau_t^s P_t(z|s) = 0 \tag{A.8}$$

$$\sum_t \gamma V_{ft} P_t(s,z|f) + \rho_z^s P_s(f|z) = 0. \tag{A.9}$$

After eliminating the Lagrange multipliers, the M-step equations are obtained as

$$\begin{aligned} P_t(s) &= \frac{\sum_{z\in\{\mathbf{z}_s\}} \sum_f P_t(s,z|f) V_{ft}}{\sum_s \sum_{z\in\{\mathbf{z}_s\}} \sum_f P_t(s,z|f) V_{ft}} \\ P_t(z|s) &= \frac{\sum_f P_t(s,z|f) V_{ft}}{\sum_{z\in\{\mathbf{z}_s\}} \sum_f P_t(s,z|f) V_{ft}}. \end{aligned} \tag{A.10}$$

The above two equations, along with the E-step update of equation (A.2), form the update equations for supervised separation. Notice that the basis vectors $P_s(f|z)$, which are learned

[13]Since observed data is modeled as a histogram, entries should be integers. To account for this, the data is weighted by an unknown scaling factor γ.

in a separate training stage, are kept fixed and not updated.

In a semi-supervised framework where one also wants to estimate the basis vectors for a subset of the sources, the update equation is obtained by solving equation (A.9) as

$$P_s(f|z) = \frac{\sum_t V_{ft} P_t(s, z|f)}{\sum_f \sum_t V_{ft} P_t(s, z|f)}. \tag{A.11}$$

Appendix B

Sparse Overcomplete Decomposition: Application to Image Data

This appendix shows the applicability of the sparse latent variable decomposition framework to analyze image data. Three applications – unsupervised feature extraction, supervised image reconstruction and supervised classification – are presented.

The CBCL database[14] is used to demonstrate the first two applications; the USPS handwritten digits database[15] demonstrates the third application. These datasets are described here briefly.

The CBCL database consists of 2429 frontal-view image faces, each image hand-aligned in a 19×19 grid. Lee and Seung (1999) have used this dataset to demonstrate the utility of NMF for extracting parts-based representations of data. Following their approach for preprocessing, the grayscale intensities were linearly scaled so that the pixel mean and standard deviation were equal to 0.25. The intensities were then clipped to the range [0, 1]. The USPS handwritten digits database consists of 8-bit grayscale 16×16 images of digits "0" through "9." There are 1100 examples of each class.

Feature Extraction

Lee and Seung (1999) applied NMF on the CBCL database and showed that the extracted basis components had localized features that fit well with intuitive notions of parts of faces. The latent variable model was applied to the database and Figure B·1(c) shows the results. The components are qualitatively similar to those extracted from NMF.

[14] available from http://cbcl.mit.edu/software-datasets/FaceData2.html
[15] available from http;//www.cs.toronto.edu/~roweis/data.html

However, the extracted bases are not entirely parts-based representations, as seen in the figure: compared to holistic representations, parts-based representations should have lower entropy. We ran experiments on the *CBCL Database* by applying sparsity on the basis vectors. Results are shown in Figure B·1(a). Decreasing the entropy of basis vectors leads to parts-like representations. Qualitatively similar results can be obtained by *increasing* the entropy of mixture weights as shown in Figure B·1(d).

Instead of parts-like representations, one can obtain holistic representations by imposing sparsity on the mixture weights, as shown by Figure B·1(e). Qualitatively similar results can be obtained by increasing the entropy of basis vectors as shown in Figure B·1(b).

Image Reconstruction

The ability of the overcomplete bases to explain new data and predict the values of unobserved components of the data was evaluated. Specifically, the approach was used to reconstruct occluded portions of images. The *CBCL database*, consisting of 2429 frontal view face images hand-aligned in 19×19 grids, was used for the experiment. Two thousand images were randomly chosen as the training set. One hundred images from the remaining 429 were randomly chosen as the test set. To create occluded test images, 6×6 grids were removed in ten random configurations for 10 test faces each, resulting in 100 occluded images. Four sets of test images, where each set had one, two, three or four 6×6 patches removed, were created. Figure B·2A illustrates the case where 4 patches were removed from each face.

In a training stage, sets of $K \in \{50, 200, 500, 750, 1000\}$ basis distributions were learned from the training data. Sparsity was not used in the compact ($K < F$) case (50 and 200 bases), while sparsity was imposed (parameter = 0.1) on the mixture weights in the overcomplete cases (500, 750 and 1000 basis vectors).

The procedure for estimating the occluded regions of a test image has two steps. In the first step, the distribution underlying the image is estimated as a linear combination of the basis distributions. This is obtained by iterations of equations (3.17) and (3.25)

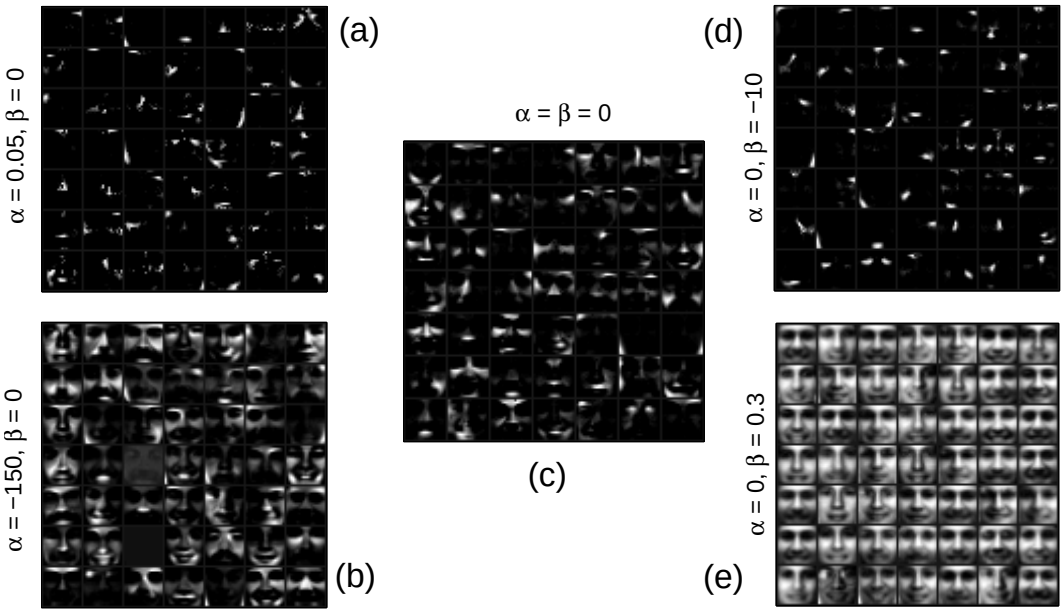


Figure B·1: Basis images extracted from the *CBCL Database* using the latent variable model. Panel (c) shows 49 basis images extracted without using sparsity. These are qualitatively similar to the basis vectors obtained by NMF (not shown). Notice that they are not entirely parts-like representations. Panels (a) and (b) show results of varying α - the sparsity parameter on the basis vectors. Panels (d) and (e) show the effects of varying β - the sparsity parameter on mixture weights. Parts-like representations are obtained when one imposes sparsity on the basis vectors (a) or increases entropy of the mixture weights (d). Increasing entropy of basis vectors (b) and decreasing entropy of the mixture weights (e) leads to holistic face-like representations.

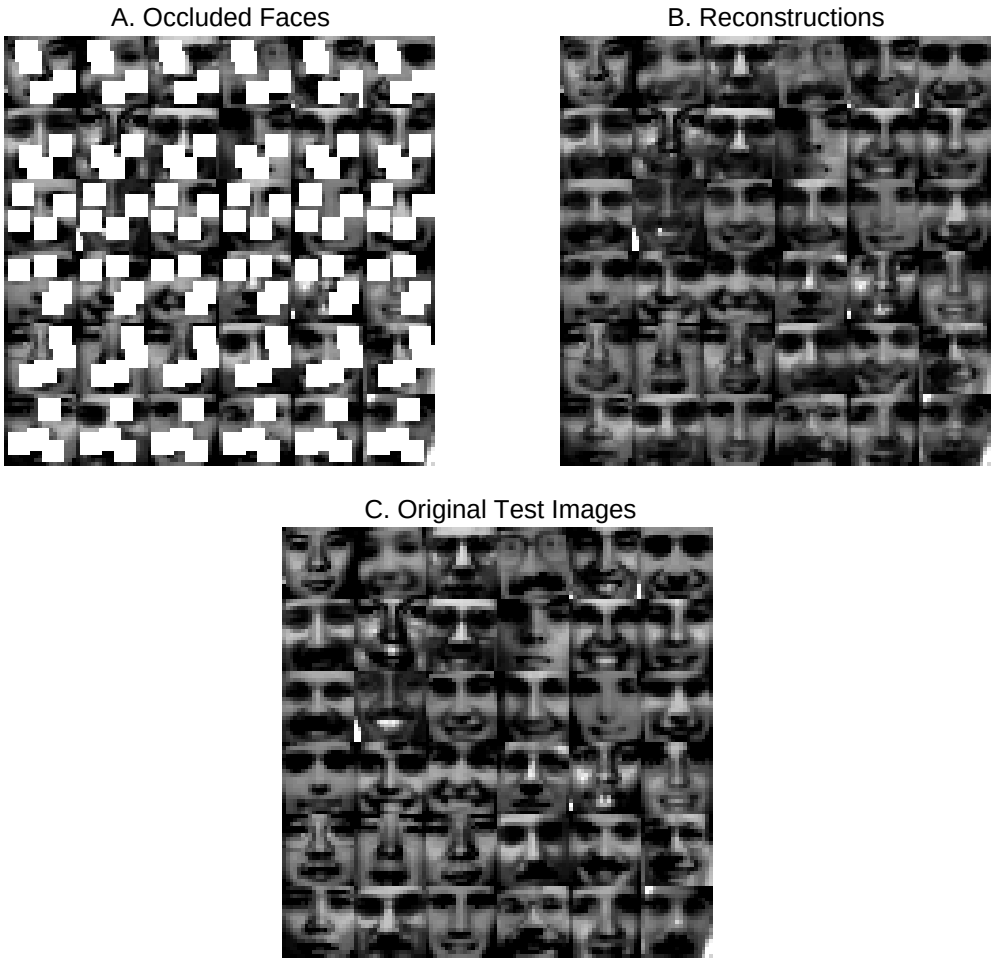


Figure B·2: Application of latent variable decomposition for reconstructing faces from occluded images (*CBCL Database*). (A). Example of a random subset of 36 occluded test images. Four 6×6 patches were removed from the images in several randomly chosen configurations (corresponding to the rows). (B). Reconstructed faces from a sparse-overcomplete basis set of 1000 learned components (sparsity parameter = 0.1). (C). Original test images are shown for comparison.

to estimate $P_t(z)$ (the bases $P(f|z)$, being already known, stay fixed), based only on the pixels that are observed (*i.e.* we marginalize out the occluded pixels). The combination of the bases $P(f|z)$ and the estimated $P_t(z)$ give the overall distribution $P_t(f)$ for the image. The occluded pixel values at any pixel f is estimated as the *expected* number of counts at the pixels, given by $P_t(f)(\sum_{f' \in \{F_o\}} V_{f'})/(\sum_{f' \in \{F_o\}} P_t(f'))$ where V_f represents the value of the image at the f^{th} pixel and $\{F_o\}$ is the set of observed pixels. Figure B·2B shows the reconstructed faces for the sparse-overcomplete case of 1000 basis vectors. Figure B·3 summarizes the results for all cases. Performance is measured by mean Signal-to-Noise-Ratio (SNR), where SNR for an image was computed as the ratio of the sum of squared pixel intensities of the original image to the sum of squared error between the original image pixels and the reconstruction.

Handwritten Digit Classification

This experiment evaluates the specificity of the bases to the process represented by the training data set for a simple example of handwritten digit classification. The USPS Handwritten Digits database which has 1100 examples for each digit class, was used. One hundred randomly chosen examples from each class were used as the test set. The remaining examples were used for training. During training, separate sets of basis distributions $P^k(f|z)$ were learned for each class, where k represents the index of the class. To classify any test image v, the distribution underlying the image was estimated using the bases for each class (by estimating the mixture weights $P_v^k(z)$, keeping the bases fixed, as before). The "match" of the bases to the test instance was indicated by the likelihood $\mathcal{L}^k$ of the image computed using $P^k(f) = \sum_z P^k(f|z)P_v^k(z)$ as $\mathcal{L}^k = \sum_f v_f \log P^k(f)$. Since the bases for the true class of a given image are expected to best compose the image, the likelihood for the correct class should be greatest. Hence, the image $\mathbf{v}$ was assigned to the class for which likelihood was the highest.

Results are shown in Figure B·4. As shown in the figure, imposing sparsity improves classification performance in almost all cases. Figure 4·7 shows four sets of basis distri-

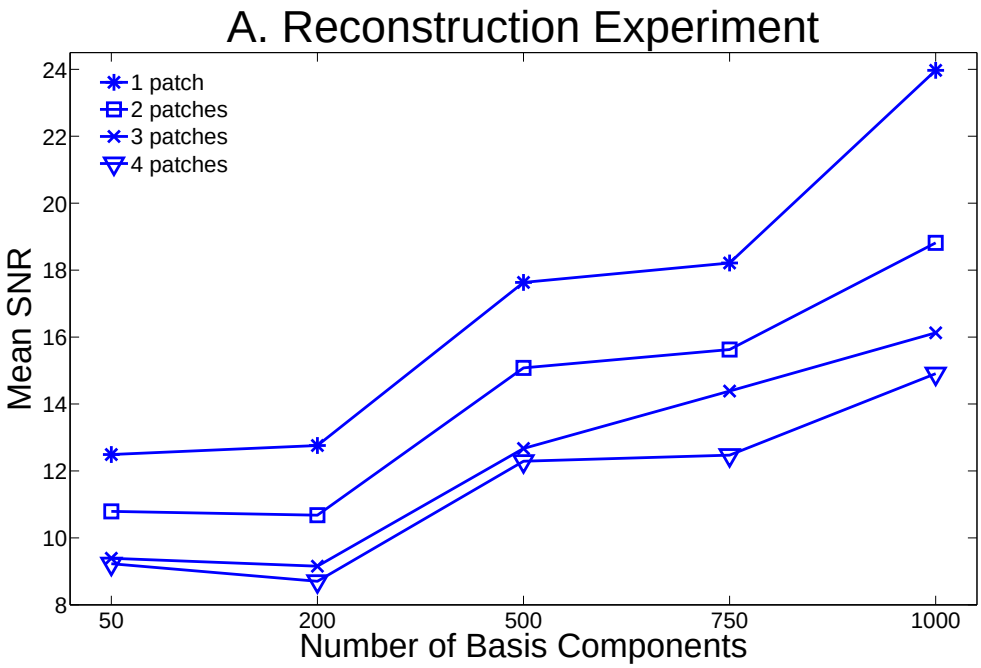


Figure B·3: Results of the face reconstruction experiment. Mean SNR of the reconstructions is shown as a function of the number of basis vectors and the test case (number of deleted patches, shown in the legend). Sparsity was not used in the compact ($K < F$) case (50 and 200 bases), while sparsity was imposed (parameter = 0.1) on the mixture weights in the overcomplete cases (500, 750 and 1000 basis vectors). Notice that the sparse-overcomplete codes consistently perform better than the compact codes.

butions learned for the handwritten digit class "3" with different sparsity values on the mixture weights. As the sparsity parameter is increased, bases tend to be holistic representations of the input histograms, consistent with improved classification performance. As the representation of basis distributions get more holistic, the more *unlike* they become when compared to bases of other classes. Thus, there is a smaller chance that the bases of one class can compose an image in another class, thereby improving performance. Only when the number of bases used is too small does performance decrease as sparsity increases (see results for 25 basis components).

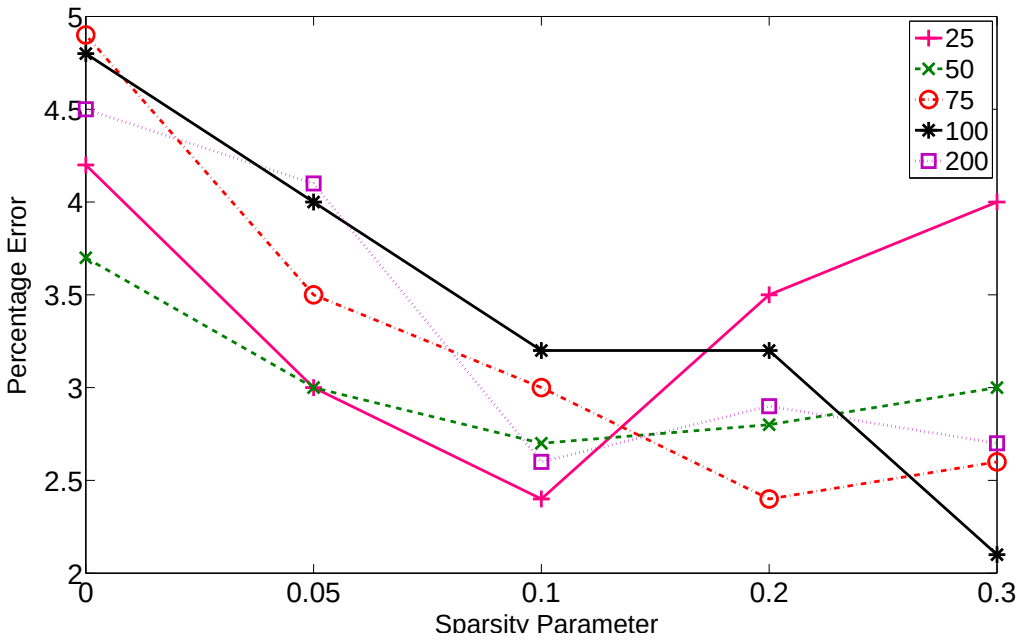


Figure B·4: Results of the classification experiment. The legend shows number of basis distributions used. Notice that imposing sparsity almost always leads to better classification performance. In the case of 100 bases, error rate comes down by almost 50% when a sparsity parameter of 0.3 is imposed.

References

Alsteris, L. and Paliwal, K. (2005). Some Experiments on Iterative Reconstruction of Speech from STFT Phase and Magnitude Spectra. In *Proceedings of Interspeech - 9th European Conference on Speech Communication and Technology*.

Asari, H., Pearlmutter, B., and Zador, A. (2006). Sparse Representations for the Cocktail Party Problem. *The Journal of Neuroscience*, 26(28):7477–7490.

Attneave, F. (1954). Informational Aspects of Visual Perception. *Psychological Review*, 61:183–193.

Barlow, H. (1959). Sensory Mechanisms, the Reduction of Redundancy, and Intelligence. In *The Mechanization of Thought Process*, National Physical Laboratory Symposium No. 10.

Barlow, H. (1961). Possible Principles Underlying the Transformation of Sensory Messages. In Rosenblith, W., editor, *Sensory Communication*, pages 217–234. MIT Press, Cambridge, MA.

Blei, D. and Lafferty, J. (2006a). Correlated Topic Models. In *Proceedings of Neural Information Processing Systems Conference*.

Blei, D. and Lafferty, J. (2006b). Dynamic Topic Models. In *Proceedings of the International Conference on Machine Learning*.

Blei, D., Ng, A., and Jordan, M. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

Borsboom, D., Mellenbergh, G., and van Heerden, J. (2003). The Theoretical Status of Latent Variables. *Psychological Review*, 110(2):203–219.

Bouguila, N., Ziou, D., and Vaillancourt, J. (2004). Unsupervised Learning of a Finite Mixture Model Based on the Dirichlet Distribution and its Application . *IEEE Transactions on Image Processing*, 13(11):1533–1543.

Brand, M. (1999a). Pattern Discovery via Entropy Minimization. In *Proceedings of Uncertainty 99: The Seventh International Workshop on Artificial Intelligence and Statistics*.

Brand, M. (1999b). Structure Learning in Conditional Probability Models via an Entropic Prior and Parameter Extinction. *Neural Computation*.

Brandstein, M. and Ward, D., editors (2001). *Microphone Arrays: Signal Processing Techniques and Applications*. Springer, Berlin.

Bregman, A. (1990). *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press, Cambridge, MA.

Bronkhorst, A. (2000). The Cocktail Party Phenomenon: A Review of Research on Speech Intelligibility in Multiple-Talker Conditions . *Acoustica*, 86:117–128.

Brown, G. (1992). *Computational Auditory Scene Analysis: A Representational Approach*. PhD thesis, University of Sheffield.

Brown, G. and Cooke, M. (1994). Computational Auditory Scene Analysis. *Computer Speech and Language*, 8(4):297–336.

Brown, G. and Wang, D. (2005). Separation of Speech by Computational Auditory Scene Analysis. In Benesty, L., Makino, S., and Chen, J., editors, *Speech Enhancement*. Springer, New York.

Brown, J. (1991). Calculation of a Constant Q Spectral Transform. *Journal of the Acoustical Society of America*, 89(1):425–434.

Candes, E. (2006). Compressive Sampling. In *Proceedings of the International Congress of Mathematics*, volume 3, pages 1433–1452.

Candes, E. and Tao, T. (2006). Near Optimal Signal Recovery from Random Projections: Universal Encoding Strategies? . *IEEE Transactions on Information Theory*, 52(12):5406–5425.

Casey, M. and Westner, A. (2000). Separation of Mixed Audio Sources by Independent Subspace Analysis. In *Proceedings of the International Computer Music Conference*, Berlin, Germany.

Chen, S., Donoho, D., and Saunders, M. (2001). Atomic Decomposition by Basis Pursuit. *SIAM Review*, 43(1):129–159.

Cherry, C. (1953). Some Experiments on the Recognition of Speech, with One and with Two Ears. *Journal of the Acoustical Society of America*, 25(5):975–979.

Choi, S., Cichoki, A., Park, H.-M., and Lee, S.-Y. (2005). Blind Source Separation and Independent Component Analysis: A Review. *Neural Information Processing: Letters and Reviews*, 6(1).

Cooke, M. (1991). *Modeling Auditory Processing and Organization*. PhD thesis, University of Sheffield.

Corless, R., Gonnet, G., Hare, D., Jeffrey, D., and Knuth, D. (1996). On the Lambert $\mathcal{W}$ Function. *Advances in Computational Mathematics*.

Deerwester, S., Dumais, G., Furnas, S., Landauer, T., and Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41:391–407.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, B*, 39:1–38.

Divenyi, P., editor (2005). *Speech Separation by Humans and Machines*. Kluwer Academic.

Donoho, D. and Elad, M. (2003). Maximal Sparsity Representation via L_1 Minimization. *Proceedings of the National Academy of Sciences*, 100:2197–2202.

Duda, R., Lyon, R., and Slaney, M. (1990). Correlograms and the Separation of Sounds. In *Proceedings of the Asilomar Conference on Signals, Systems and Computers*.

Eggert, J. and Korner, E. (2004). Sparse Coding and NMF. *Neural Networks*.

Ellis, D. (1991). A Wavelet-Based Sinusoid Model of Sound for Auditory Signal Separation. In *Proceedings of International Computer Music Conference*, pages 86–89.

Ellis, D. (1992). A Perceptual Representation of Sound. Master's thesis, MIT.

Ellis, D. (1996). *Prediction Driven Computational Auditory Scene Analysis*. PhD thesis, MIT.

Field, D. (1994). What is the Goal of Sensory Coding? *Neural Computation*.

Fuchs, J.-J. (2004). On Sparse Representations in Arbitrary Redundant Bases. *IEEE Transactions on Information Theory*, 50(6):1341–1344.

Gabor, D. (1946). Theory of Communications. *Journal of the IEE*, 93:429–457.

Godsmark, D. and Brown, G. (1999). A Blackboard Architecture for Computational Auditory Scene Analysis. *Speech Communication*, 27:351–366.

Goodman, L. (1974). Exploratory Latent Structure Analysis using both Identifiable and Unidentifiable Models. *Biometrika*, 61:215–231.

Green Jr., B. (1952). Latent Structure Analysis and its Relation to Factor Analysis. *Journal of the American Statistical Association*, 47:71–76.

Grossberg, S., Govindarajan, K., Wyse, L., and Cohen, M. (2004). ARTSTREAM: A Neural Network Model of Auditory Scene Analysis and Source Segregation. *Neural Networks*, 17:511–536.

Haykin, S. and Chen, Z. (2005). The Cocktail Party Problem. *Neural Computation*, 17:1875–1902.

Heiler, M. and Schnorr, C. (2006). Learning Sparse Representations by Non-negative Matrix Factorization and Sequential Cone Programming. *Journal of Machine Learning Research*, pages 1385–1407.

Helmholtz, H. (1863). *On the Sensation of Tone*. Dover Publishers. English Translation, 1954.

Hofmann, T. (2001). Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Machine Learning*, 42:177–196.

Hoyer, P. (2004). Non-negative Matrix Factorization with Sparseness Constraints. *Journal of Machine Learning Research*, 5.

Hyvarinen, A. (1999). Survey on Independent Component Analysis. *Neural Computing Surveys*, 2:94–128.

Hyvarinen, A. and Hoyer, P. (2000). Emergence of Phase and Shift Invariant Features by Decomposition of Natural Images into Independent Feature Subspaces. *Neural Computation*, 12(7):1705–1720.

Irino, T. and Patterson, R. (1997). A Time-Domain, Level-Dependent Auditory Filter: The Gammachirp. *Journal of the Acoustical Society of America*, 101:412–419.

Jang, G.-J. and Lee, T.-W. (2003). A Maximum Likelihood Approach to Single-Channel Source Separation. *Journal of Machine Learning Research*, 4:1365–1393.

Jaynes, E. (1982). *Papers on Probability, Statistics and Statistical Mechanics* . Kluwer Academic.

Kanerva, P. (1988). *Sparse Distributed Memory*. The MIT Press.

Lazarsfeld, P. and Henry, N. (1968). *Latent Structure Analysis*. Boston: Houghton Mifflin.

Lee, D. and Seung, H. (1999). Learning the Parts of Objects by Non-negative Matrix Factorization. *Nature*, 401.

Lee, D. and Seung, H. (2001). Algorithms for Non-negative Matrix Factorization. *Advances in Neural Information Processing Systems*, 13.

Lewicki, M. (2002). Efficient Coding of Natural Sounds. *Nature Neuroscience*, 5(4):356–363.

Loughlin, P., Pitton, J., and Atlas, L. (1994). Construction of Positive Time-Frequency Distributions. *IEEE Transactions on Signal Processing*, 42(10):2697–2705.

McLachlan, G. and Krishnan, T. (1997). *The EM Algorithm and Extensions*. Wiley.

Mellinger, D. (1991). *Event Formation and Separation in Musical Sound*. PhD thesis, Stanford University.

Metzger, W. (1953). *Gesetze des Sehens*, pages 59–60. Waldemer Kramer, Frankfurt am Main, Germany.

Minka, T. (2003). Estimating a Dirichlet Distribution. Technical report, Microsoft Research.

Morup, M. and Schmidt, M. (2006). Sparse Non-negative Matrix Factor 2-D Deconvolution. Technical report, Technical University of Denmark.

Neal, R. and Hinton, G. (1998). A View of the EM Algorithm that Justifies Incremental, Sparse and Other Variants. In Jordan, M., editor, *Learning in Graphical Models*, pages 355–368. Kluwer Academic, Dordrecht.

Nikias, C. and Petropulu, A. (1993). *Higher-Order Spectra Analysis*. PTR Prentice Hall.

O'Grady, P. (2007). *Sparse Separation of Under-determined Speech Mixtures*. PhD thesis, National University of Ireland, Maynooth.

Olshausen, B. (2001). Sparse Codes and Spikes. In Rao, R., Olshausen, B., and Lewicki, M., editors, *Probabilistic Models of Perception and Brain Function*, pages 245–260. MIT Press, Cambridge, MA.

Olshausen, B. and Field, D. (1996). Emergence of Simple-Cell Properties by Learning a Sparse Code for Natural Images. *Nature*, 381.

Olshausen, B. and Field, D. (2004). Sparse Coding of Sensory Inputs. *Current Opinion in Neurobiology*, 14:481–487.

Paatero, P. and Tapper, U. (1994). Positive Matrix Factorization: A Non-Negative Factor Model with Optimal Utilization of Error Estimates of Data Values. *Environmetrics*, 5:111–126.

Pascaul-Montano, A., Carazo, J., Kochi, K., Lehmann, D., and Pascaul-Marqui, R. (2006). Nonsmooth Nonnegative Matrix Factorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3).

Patterson, R., Allerhand, M., and Giguere, C. (1995). Time-domain Modeling of Peripheral Auditory Processing: A Modular Architecture and Software Platform. *Journal of the Acoustical Society of America*, 98:1890–1894.

Plomp, R. (2002). *The Intelligent Ear: On the Nature of Sound Perception*. Lawrence Earlbaum.

Plumbley, M. (2003). Algorithms for Nonnegative Independent Component Analysis. *IEEE Transactions on Neural Networks*, 14(3):534–543.

Raj, B., Shashanka, M., and Smaragdis, P. (2006). Latent Dirichlet Decomposition for Single Channel Speaker Separation. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*.

Raj, B., Singh, R., Shashanka, M., and Smaragdis, P. (2007). Bandwidth Expansion with a Polya Urn Model. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*.

Raj, B. and Smaragdis, P. (2005). Latent Variable Decomposition of Spectrograms for Single Channel Speaker Separation. In *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*.

Roman, N. (2005). *Auditory Based Algorithms for Sound Segregation in Multisource and Reverberant Environments.* PhD thesis, The Ohio State University.

Rosenthal, D. and Okuno, H., editors (1998). *Computational Auditory Scene Analysis.* Lawrence Erlbaum Associates.

Roweis, S. (2001). One Microphone Source Separation. In *Advances in Neural Information Processing Systems*, volume 13, pages 793–799.

Salton, G. and McGill, M. (1983). *Introduction to Modern Information Retrieval.* McGraw-Hill, New York.

Skilling, J. (1989). Classic Maximum Entropy. In Skilling, J., editor, *Maximum Entropy and Bayesian Methods.* Kluwer Academic.

Slaney, M. and Lyon, R. (1990). A Perceptual Pitch Detector. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 357–360.

Smaragdis, P. (2001). *Redundancy Reduction for Computational Audition, a Unifying Approach.* PhD thesis, Massachusetts Institute of Technology.

Smaragdis, P. (2004). Discovering Auditory Objects through Non-negativity Constraints . In *Proceedings of the Workshop on Perceptual and Statistical Audio Processing.*

Smaragdis, P. (2007). Convolutive Speech Bases and their Application to Supervised Speech Separation. *IEEE Transactions on Audio, Speech and Language Processing*, 15(1):1–12.

Smaragdis, P. and Brown, J. (2003). Non-negative Matrix Factorization for Polyphonic Music Transcription . In *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics.*

Smaragdis, P. and Raj, B. (2007). Shift-Invariant Probabilistic Latent Component Analysis. *Journal of Machine Learning Research (submitted).*

Smaragdis, P., Raj, B., and Shashanka, M. (2006). A Probabilistic Latent Variable Model for Acoustic Modeling. In *NIPS Workshop on Advances in Modeling for Acoustic Processing.*

Smith, E. and Lewicki, M. (2006). Efficient Auditory Coding. *Nature*, 439:978–982.

Spearman, C. (1904). General Intelligence, Objectively Determined and Measured. *American Journal on Psychology*, 15:201–293.

Suppes, P. and Zanotti, M. (1981). When are Probabilistic Explanations Possible? *Synthese*, 48:191–199.

van der Kouwe, A., Wang, D., and Brown, G. (2001). A Comparison of Auditory and Blind Separation Techniques for Speech Segregation. *IEEE Transactions on Speech and Audio Processing*, 9(3):189–195.

Vercoe, B. and Cumming, D. (1988). Connection Machine Tracking of Polyphonic Audio. In *Proceedings of the International Computer Music Conference*, pages 211–218.

Virtanen, T. (2006). *Sound Source Separation in Monaural Music Signals*. PhD thesis, Tampere University of Technology.

Virtanen, T. (2007). Monaural Sound Source Separation by Nonnegative Matrix Factorization with Temporal Continuity and Sparseness Criteria. *IEEE Transactions on Audio, Speech and Language Processing*, 15(3):1066–1074.

Wang, D. (2005). On Ideal Binary Mask as the Computational Goal of Auditory Scene Analysis. In Divenyi, P., editor, *Speech Separation by Humans and Machines*, pages 181–197. Kluwer Academic.

Weintraub, M. (1985). *A Theory and Computational Model of Monaural Auditory Sound Separation*. PhD thesis, Stanford University.

Yilmaz, O. and Rickard, S. (2004). Blind Separation of Speech Mixtures via Time-Frequency Masking. *IEEE Transactions on Signal Processing*, 52(7):1830–1847.

Zibulevsky, M. and Pearlmutter, B. (2001). Blind Source Separation by Sparse Decomposition. *Neural Computation*, 13(4):863–882.

Zou, H., Hastie, T., and Tibshirani, R. (2004). Sparse Principal Component Analysis. Technical report, Stanford University.

Curriculum Vitae

Madhusudana Shashanka

Department of Cognitive and Neural Systems
Boston University
677 Beacon St.
Boston, MA 02215

Tel: (617) 817-7235
Fax: (617) 353-7755
shashanka@cns.bu.edu
http://cns.bu.edu/∼mvss/

Research Areas

Machine Learning, Semantic Analysis, Auditory Scene Analysis

Education

2003 - present **Boston University** **Boston, MA, USA**
PhD Candidate, Cognitive and Neural Systems (CNS) 3.96/4.00
Advisor: Prof. Barbara Shinn-Cunningham

1999 - 2003 **Birla Institute of Technology & Science (BITS)** **Pilani, India**
BE (Honors) in Computer Science 9.39/10.00

Publications

Journals

MVS Shashanka, B Raj, P Smaragdis. Probabilistic Latent Variable Model for Sparse Decompositions of Non-negative Data. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, submitted.

P Smaragdis, MVS Shashanka. A Framework for Secure Speech Recognition. *IEEE Trans. on Audio, Speech and Language Processing*, to appear.

E Ardizzoni, AA Bertossi, MC Pinotti, S Ramaprasad, R Rizzi, MVS Shashanka. Optimal Skewed Data Allocation on Multiple Channels with Flat Broadcast per Channel. *IEEE Trans. on Computers*, Vol. 54, No. 5, May 2005.

MVS Shashanka, A Pati, AM Shende. A Characterisation of Optimal Channel Assignments for Wireless Networks Modelled as Cellular and Square Grids. *Mobile Networks and Applications*, Vol. 10, Issue 1-2, Feb-Apr 2005.

Conferences

MVS Shashanka, B Raj, P Smaragdis. Sparse Overcomplete Latent Variable Decomposition of Counts Data. submitted to *Neural Information Processing Systems Conference* (NIPS), 2007.

MVS Shashanka, P Smaragdis. Privacy-Preserving Musical Database Matching. *IEEE WASPAA*, New Paltz, New York, Oct 2007.

P Smaragdis, B Raj, MVS Shashanka. Supervised and Semi-Supervised Separation of Sounds from Single-Channel Mixtures. *Intl. Conf. on ICA and Signal Separation*, London, UK, Sep 2007.

MVS Shashanka, B Raj, P Smaragdis. Sparse Overcomplete Decomposition for Single Channel Speaker Separation. *IEEE ICASSP*, Honolulu, Hawaii, Apr 2007.

P Smaragdis, MVS Shashanka. A Framework for Secure Speech Recognition. *IEEE ICASSP*, Honolulu, Hawaii, Apr 2007.

B Raj, R Singh, MVS Shashanka, P Smaragdis. Bandwidth Expansion with a Polya Urn Model. *IEEE ICASSP*, Honolulu, Hawaii, Apr 2007.

B Raj, P Smaragdis, MVS Shashanka, R Singh. Separating a Foreground Singer from Background Music. *Intl Symposium on Frontiers of Research on Speech and Music*, Mysore, India, Jan 2007.

P Smaragdis, B Raj, MVS Shashanka. A Probabilistic Latent Variable Model for Acoustic Modeling. *NIPS Workshop on Advances in Models for Acoustic Processing*, Dec 2006.

MVS Shashanka, P Smaragdis. Secure Sound Classification: Gaussian Mixture Models. *IEEE ICASSP*, Toulouse, France, May 2006. (**Finalist for the Best Student Paper Award**).

B Raj, MVS Shashanka, P Smaragdis. Latent Dirichlet Decomposition for Single Channel Speaker Separation. *IEEE ICASSP*, France, May 2006.

AA Bertossi, MC Pinotti, S Ramaprasad, R Rizzi, MVS Shashanka. Optimal Multi-Channel Data Allocation with Flat Broadcast per Channel. *Proceedings of IPDPS*, Santa Fe, USA, Apr 2004.

MVS Shashanka, A Pati, AM Shende. A Characterisation of Optimal Channel Assignments for Wireless Networks Modelled as Cellular and Square Grids. *Proceedings of IPDPS*, Nice, France, Apr 2003.

A Dubhashi, MVS Shashanka, A Pati, S Ramaprasad, AM Shende. Channel Assignment for Wireless Networks Modelled as d-Dimensional Square Grids. *Proc. of Intl. Workshop on Distributed Computing*, India, Dec 2002.

Abstracts

BG Shinn-Cunningham, S Bressler, MVS Shashanka. Separating and Understanding a Talker from a Mixture in Reverberant Spaces. *152nd Meeting of the Acoust. Soc. of America*, Honolulu, Hawaii, Dec 2006.

MVS Shashanka, BG Shinn-Cunningham, S Nasser. The Role of Fundamental Frequency in Segregating and Understanding a Talker Competing with Another Talker in a Reverberant Setting. *29th Midwinter Meeting of the ARO*, Baltimore, Feb 2006.

Poster

MVS Shashanka, BG Shinn-Cunningham, M Cooke. Effects of Reverberant Energy on Statistics of Speech. *Workshop on Speech Separation and Comprehension in Complex Acoustic Environments*, Montreal, Nov 2004.

Patents

P Smaragdis, MVS Shashanka. System and Method for Recognizing Speech Securely. *USA Patent Application filed.*

MVS Shashanka, P Smaragdis. Secure Classification of Data with Gaussian Distributions. *USA Patent Application filed.*

Teaching Experience

Spring 2005 **Teaching Assistant for CN550** **Boston University**
Computational Models of Recognition, Memory and Attention

Research Experience

2006 - Present **Mitsubishi Electric Research Labs** **Cambridge, MA, USA**
Research Intern. Working on probabilistic models and algorithms for separating sources from single-channel audio recordings.

2004 - Present **Boston University** **Boston, MA, USA**
Research Assistant in the *Auditory Neuroscience Lab.* Designed and conducted experiments to understand human auditory processing in natural environments.

Summer 2005 **Mitsubishi Electric Research Labs** **Cambridge, MA, USA**
Research Intern. Developed algorithms for secure classification between parties such that private data is not exchanged; worked on sound separation algorithms.

Spring 2003 **University of Trento** **Trento, Italy**
Research Fellow in the *Dept. of Informatics and Telecommunication.* De-

veloped efficient algorithms for optimally allocating data over multiple channels for broadcast such that average client waiting time is minimized.

Summer 2002 **Indian Institute of Science (IISc)** **Bangalore, India**
Summer Research Fellow (program administered by Dept. of Science and Technology, Govt. of India) in the *Dept. of Electrical Engineering*. Worked on applying learning automata to the problem of channel assignment in wireless networks.

Summer 2001 **Bhabha Atomic Research Centre (BARC)** **Mumbai, India**
Summer Intern in the *Ultrasonics Instrumentation Section, Electronics Division*. Developed a software module using Visual Basic for better visualization of scan data for an ultrasonic imaging system.

Entrepreneurial Experience

Fall 2005 Completed technology and venture assessments for the Boston University startup Biomimetic Systems (`http://www.biomimetic-systems.com`).

Skill Set

Programming	MATLAB, C, Java
OS	Windows, UNIX, GNU/Linux
Publishing	MS Office, TEX, LATEX, HTML/CSS

Scholarships & Awards

2003 - 2004	**Arts and Science Dean's Fellowship**, Boston University
Summer 2002	**Research Fellowship (top 3%)**, JN Center for Advanced Scientific Rsearch and Govt. of India
1999 - 2002	**Merit Scholarship**, Birla Institute of Technology & Science (awarded to top ten students across majors in order of merit)
1997 - 2003	**National Talent Search Scholarship**, Govt. of India
1999	**National Top 1%** of 17846 candidates, Natl. Standard Exam in Physics. **Rank 7** in Pre University Examinations, State of Karnataka.
1997	**Rank 1** in the state of Karnataka, National Talent Search Exam. **Rank 16** in Secondary School Examinations, State of Karnataka. **Rank 6** in Regional Mathematical Olympiad, State of Karnataka.

1996	**Rank 13** in Regional Mathematical Olympiad, State of Karnataka. **Rank 12** XXVII Inter-State Talent Tests in Mathematics.

Talks / Presentations

Latent Variable Decomposition - Models & Applications (with B Raj).
MIT Brains and Machines Seminar, Cambridge, MA, 16 May 2007.

Probabilistic Models for Single Channel Audio Processing.
Boston University Hearing Research Seminar, Boston, MA, 19 Jan 2007.

Probabilistic Models for Acoustic Processing.
Mitsubishi Electric Research Labs, Cambridge, MA, 21 Nov 2006.

Graduate Courses

Hearing	Neural and Computational Models of Speech and Hearing Perception Neural Coding & Perception of Sound (*Harvard/MIT SHBT Program*) Psychoacoustics
CNS	Computational Neuroscience (Introductory and Advanced) Neural and Computational Models of Recognition, Memory and Attention Neural and Computational Models of Vision Neural and Computational Models of Adaptive Movement & Planning
Engineering	Information Theory and Coding Signals and Systems Technology Commercialization Estimation Theory

Professional Activities & Memberships

Student Member, Institute of Electrical and Electronics Engineers (IEEE)
External Reviewer, Workshop on Statistical and Perceptual Audition 2006
Reviewer, IEEE Transactions on Signal Processing
Reviewer, International Conference on Multimedia and Expo (ICME 2007)
Reviewer, Neural Information Processing Systems (NIPS) 2007

Miscellaneous

Languages	English, Kannada (fluent reading, writing, speaking), Hindi (functional)
Citizenship	India